

4

SAINS KOMPUTER TINGKATAN 4

SAINS KOMPUTER

TINGKATAN

4



KEMENTERIAN
PENDIDIKAN
MALAYSIA



Harga: RM25.00
ISBN 978-983-47-2013-1

9 789834 720131
FT304001





RUKUN NEGARA

Bahwasanya Negara Kita Malaysia

mendukung cita-cita hendak;

Mencapai perpaduan yang lebih erat dalam kalangan seluruh masyarakatnya;

Memelihara satu cara hidup demokrasi;

Mencipta satu masyarakat yang adil di mana kemakmuran negara akan dapat dinikmati bersama secara adil dan saksama;

Menjamin satu cara yang liberal terhadap tradisi-tradisi kebudayaannya yang kaya dan pelbagai corak;

Membina satu masyarakat progresif yang akan menggunakan sains dan teknologi moden;

MAKA KAMI, rakyat Malaysia,
berikrar akan menumpukan

seluruh tenaga dan usaha kami untuk mencapai cita-cita tersebut berdasarkan prinsip-prinsip yang berikut:

**KEPERCAYAAN KEPADA TUHAN
KESETIAAN KEPADA RAJA DAN NEGARA
KELUHURAN PERLEMBAGAAN
KEDAULATAN UNDANG-UNDANG
KESOPANAN DAN KESUSILAAN**

(Sumber: Jabatan Penerangan Kementerian Komunikasi dan Multimedia Malaysia)

KURIKULUM STANDARD SEKOLAH MENENGAH

SAINS KOMPUTER

TINGKATAN

4

PENULIS

Chua Siwe Geck
Yew Kwang Hooi
Zaliha binti Mohamad
Fatimah binti Ismail @ Mohd Nor

EDITOR

Intan Farhanna binti Uyob

PEREKA BENTUK

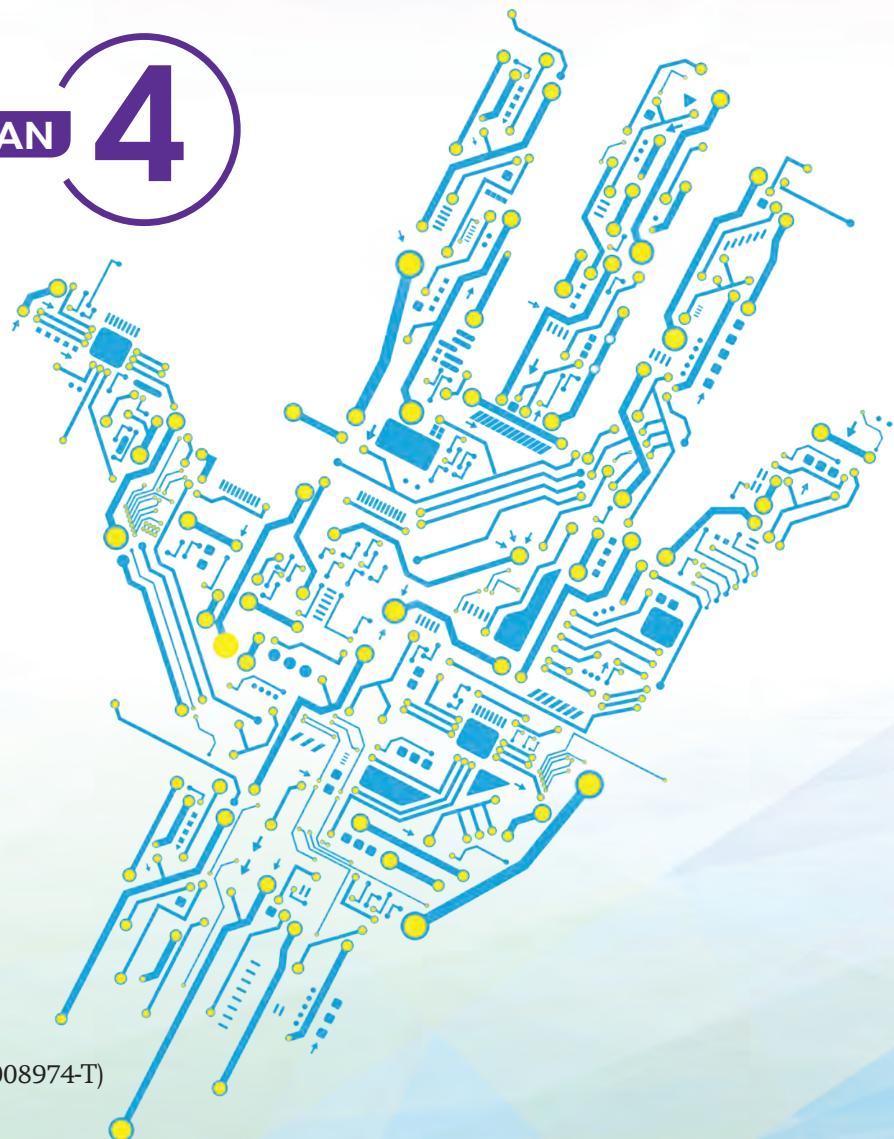
Brenda Tan

ILUSTRATOR

Nurul Atiqah binti Ibrahim



Oxford Fajar Sdn. Bhd. (008974-T)





KEMENTERIAN
PENDIDIKAN
MALAYSIA

No. Siri Buku: 0010

KPM2016 ISBN 978-983-47-2013-1

Cetakan Pertama 2016

© Kementerian Pendidikan Malaysia

Hak Cipta Terpelihara. Mana-mana bahan dalam buku ini tidak dibenarkan diterbitkan semula, disimpan dalam cara yang boleh dipergunakan lagi, ataupun dipindahkan dalam sebarang bentuk atau cara, baik dengan cara elektronik, mekanik, penggambaran semula mahupun dengan cara perakaman tanpa kebenaran terlebih dahulu daripada Ketua Pengarah Pelajaran Malaysia, Kementerian Pendidikan Malaysia. Perundingan tertakluk kepada perkiraan royalti atau honorarium.

Diterbitkan untuk Kementerian Pendidikan Malaysia oleh:

Oxford Fajar Sdn. Bhd.

4, Jalan Pemaju U1/15,Seksyen U1,

Hicom-Glenmarie Industrial Park,

40150 Shah Alam,

Selangor Darul Ehsan.

Tel: 03-56294000

Faks: 03-56294009

Laman Web:<http://www.oxfordfajar.com.my>

Reka Letak dan Atur Huruf:

Oxford Fajar Sdn. Bhd.

Muka Taip Teks: Helvetica Neue LT Std 47 Light Condensed

Saiz Muka Taip Teks: 11 pt

Dicetak oleh:

Percetakan Rina Sdn. Bhd.,

Lot 45, Persiaran Mewah,

Bandar Tun Razak, Cheras

56000 Kuala Lumpur

Penghargaan

Penerbitan buku teks ini melibatkan kerjasama banyak pihak. Sekalung penghargaan dan terima kasih ditujukan kepada semua pihak yang terlibat:

- Pegawai-pegawai Bahagian Buku Teks dan Bahagian Pembangunan Kurikulum, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penambahbaikan Pruf Muka Surat, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penyemakan Pembetulan Pruf Muka Surat, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Jawatankuasa Penyemakan Naskhah Sedia Kamera, Bahagian Buku Teks, Kementerian Pendidikan Malaysia.
- Pensyarah-pensyarah dari UNITAR International University
- Pensyarah-pensyarah dari Universiti Teknologi Petronas



Kandungan



Halaman Judul	I
Halaman Hak Cipta dan Penghargaan	II
Pendahuluan	IV

BAB

1

PENGATURCARAAN	1
1.1 Strategi Penyelesaian Masalah	2
1.2 Algoritma	20
1.3 Pemboleh Ubah, Pemalar dan Jenis Data	56
1.4 Struktur Kawalan	75
1.5 Amalan Terbaik Pengaturcaraan	111
1.6 Struktur Data dan Modular	129
1.7 Pembangunan Aplikasi	168
Imbas Semula	187
Latihan Sumatif 1	189

BAB

2

PANGKALAN DATA	193
2.1 Pangkalan Data Hubungan	194
2.2 Reka Bentuk Pangkalan Data Hubungan	218
2.3 Pembangunan Pangkalan Data Hubungan	240
2.4 Pembangunan Sistem Pangkalan Data	291
Imbas Semula	309
Latihan Sumatif 2	310

BAB

3

INTERAKSI MANUSIA DENGAN KOMPUTER	314
3.1 Reka Bentuk Interaksi	315
3.2 Paparan dan Reka Bentuk Skrin	326
Imbas Semula	342
Latihan Sumatif 3	343

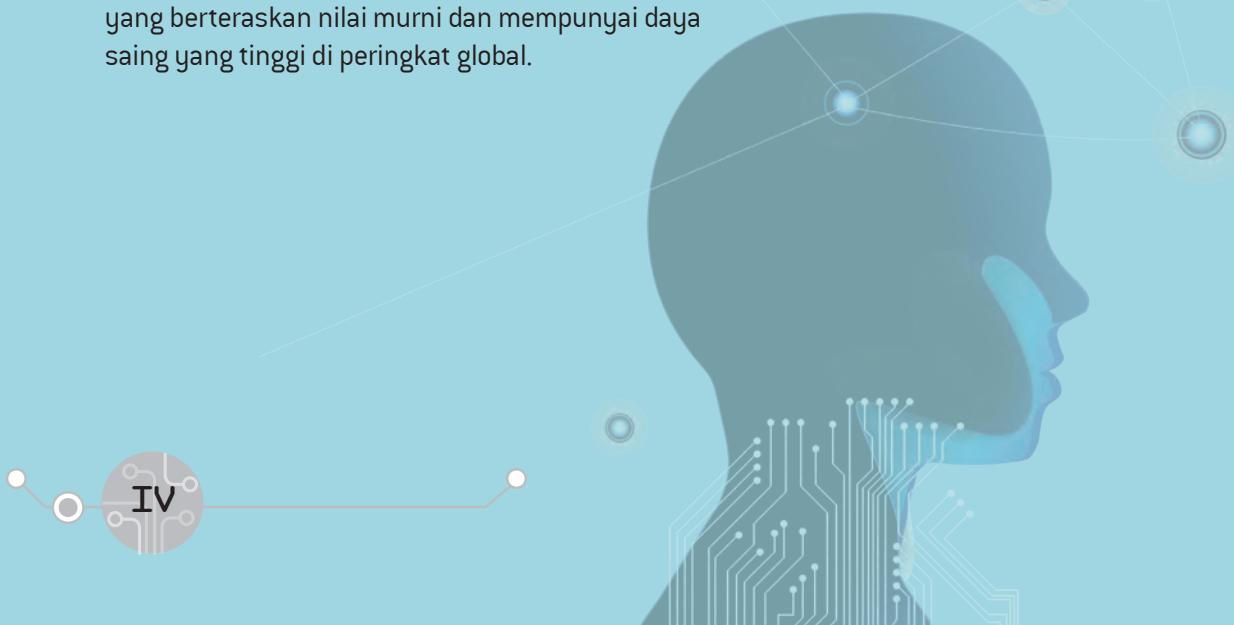
Glosari	345
Senarai Rujukan	346



Pendahuluan

Buku Teks Sains Komputer Tingkatan 4 ditulis berdasarkan **Dokumen Standard Kurikulum dan Pentaksiran (DSKP)** terbitan Kementerian Pendidikan Malaysia. Buku teks ini memperkuuh dan menyediakan murid dengan pengetahuan, kemahiran dan nilai Sains Komputer serta pemikiran komputasional. Pemikiran komputasional membangunkan kemahiran murid melalui keupayaan mereka bentuk, menghasil dan mencipta aplikasi baharu. Murid yang berfikiran komputasional mampu menyelesaikan masalah dan dapat menyumbangkan idea dalam perkembangan teknologi dunia baharu yang dapat menyelesaikan masalah secara kolaboratif.

Buku ini berorientasikan beberapa strategi pembelajaran yang berpusatkan murid. Strategi ini merangkumi pembelajaran kendiri, pembelajaran kolaboratif, pembelajaran berasaskan projek, pembelajaran berasaskan inkuiri, pembelajaran penyelesaian masalah, pembelajaran masteri, pembelajaran konstrutivisme dan pembelajaran penerokaan. Aspek Kemahiran Proses Sains Komputer dan Kemahiran Berfikir Aras Tinggi (KBAT) juga dititikberatkan dalam buku ini untuk membina pemikiran murid ke arah kreativiti dan inovasi. KBAT dapat merangsang pemikiran berstruktur dan berfokus dalam kalangan murid melalui empat tahap pemikiran, iaitu mengaplikasi, menganalisis, menilai dan mencipta. Sementelah murid berupaya memupuk beberapa kemahiran seperti kemahiran berfikiran kritis, kreatif dan menaakul serta strategi berfikir. Penggunaan peta i-Think juga dititikberatkan di dalam buku ini. Penggunaan peta i-Think mendedahkan murid dengan teknik pembelajaran yang ringkas dan mudah difahami. Murid akan lebih mudah untuk mengingati sesuatu fakta melalui penggunaan peta i-Think yang menarik serta mempunyai warna yang dapat merangsang daya pemikiran murid serta menarik perhatian murid untuk menggunakanannya. Elemen Merentas Kurikulum (EMK) merupakan unsur nilai tambah yang diterapkan dalam proses pengajaran dan pembelajaran untuk mengukuhkan kemahiran murid di samping melahirkan murid yang berwibawa dan mampu menerima cabaran pada masa hadapan. EMK merangkumi lapan aspek yang utama, iaitu bahasa, kelestarian alam sekitar, nilai murni, sains dan teknologi, patriotisme, kreativiti dan inovasi, keusahawanan serta teknologi maklumat dan komunikasi. Kemahiran Abad Ke-21 juga diterapkan dalam buku ini untuk melahirkan murid yang mempunyai kemahiran hidup yang berteraskan nilai murni dan mempunyai daya saing yang tinggi di peringkat global.



Ikon-ikon dalam Buku Teks dan Fungsinya



Aplikasi Harian

Mengaplikasikan pengetahuan sains komputer dalam kehidupan harian



Uji Minda

Menyediakan persoalan menarik untuk menguji tahap kefahaman murid tentang bidang sains komputer



MalaysiaKu!

Menonjolkan kejayaan Malaysia dalam bidang sains komputer



Kerjaya Sains Komputer

Memperkenalkan jenis kerjaya yang berhubung dengan bidang sains komputer



Inovasi Sains Komputer

Mengetengahkan perkembangan dan inovasi teknologi sains komputer pada masa kini



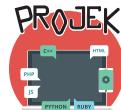
KBAT

Mendedahkan murid kepada soalan Kemahiran Berfikir Aras Tinggi



Tahukah Anda?

Memberikan info menarik tentang topik yang sedang dipelajari



Pelbagai aktiviti yang bercorak penerokaan atau berdasarkan bahan rangsangan untuk menambahkan kefahaman murid tentang standard pembelajaran yang telah dipelajari.



Pelbagai sumber digital yang menarik dan berinformasi boleh diakses daripada portal Sains Komputer Tingkatan 4, iaitu www.skomtg4.com/#2

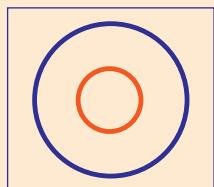


Cara Penggunaan Peta Pemikiran i-Think



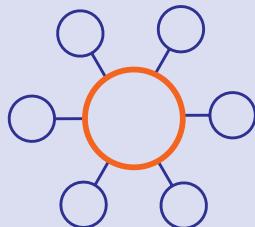
PETA BULATAN

Untuk mengadakan sumbang saran bagi menjanakan idea.



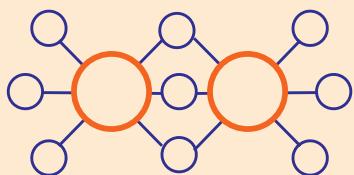
PETA BUIH

Untuk menunjukkan ciri sesuatu item atau konsep



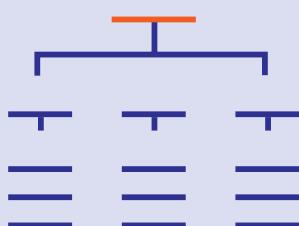
PETA BUIH BERGANDA

Untuk membandingkan dan membezakan item



PETA POKOK

Untuk mengelaskan item berdasarkan sifat sepunya



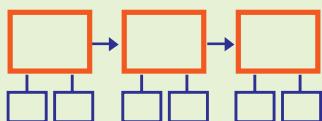
PETA DAKAP

Untuk menganalisis bahagian atau struktur objek fizikal



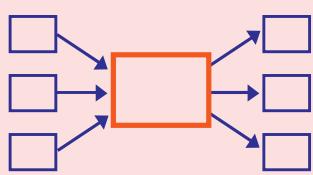
PETA ALIR

Untuk membuat urutan bagi sesuatu proses secara logik



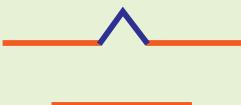
PETA PELBAGAI ALIR

Untuk menunjukkan hubungan punca dan kesan



PETA TITI

Untuk membuat analogi menggunakan faktor penghubung



1

BAB

PENGATURCARAAN

Penggunaan komputer, telefon pintar dan *tablet* telah menjadi satu keperluan dalam kehidupan masa kini. Interaksi antara manusia dengan peralatan komunikasi telah mewujudkan satu bentuk bahasa yang dikenal sebagai bahasa pengaturcaraan. Pelbagai aplikasi telah direka menggunakan bahasa pengaturcaraan untuk membantu manusia dalam kehidupan harian. Adakah penggunaan pelbagai aplikasi dapat membantu anda menyelesaikan masalah?

Dalam bab ini, anda akan mempelajari:

- » Strategi penyelesaian masalah
- » Algoritma
- » Boleh ubah, pemalar dan jenis data
- » Struktur kawalan
- » Amalan terbaik pengaturcaraan
- » Modular dan struktur data
- » Pembangunan aplikasi (*apps development*)

Fikirkan...

- Dapatkah anda bayangkan kehidupan anda tanpa penggunaan komputer dan telefon pintar?
- Apakah aplikasi-aplikasi sedia ada yang dapat menyelesaikan masalah, sama ada mudah ataupun kompleks?
- Apakah aplikasi yang boleh dicipta untuk menyelesaikan masalah dalam tempoh tiga hingga tujuh tahun dari sekarang?





STANDARD PEMBELAJARAN

- 1.1.1 Menerangkan keperluan penyelesaian masalah berstrategi
- 1.1.2 Menjelaskan ciri-ciri penyelesaian masalah berkesan
- 1.1.3 Menggunakan proses penyelesaian masalah: kumpulkan dan analisis data, tentukan masalah, jana idea – hapus halangan, jana penyelesaian, tentukan tindakan, laksanakan penyelesaian, buat penilaian, dan buat penambahbaikan

Strategi Penyelesaian Masalah

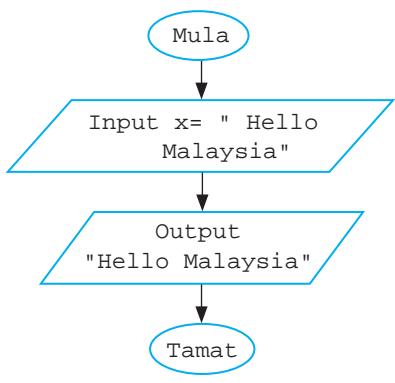
Masalah ialah keraguan, situasi yang tidak diingini, cabaran dan peluang yang dihadapi dalam kehidupan seseorang. Kemahiran membuat keputusan amat diperlukan untuk melahirkan generasi yang berkebolehan merancang strategi penyelesaian masalah. Contohnya, kita perlu membuat keputusan tentang jenis makanan yang hendak dibeli di kantin dengan jumlah wang yang ada.

Penyelesaian masalah ialah tunjang utama dalam sains komputer. Pengatur cara perlu memahami cara penyelesaian sesuatu masalah dan menterjemahkan cara tersebut kepada algoritma yang dapat difahami oleh komputer. Komponen yang bertanggungjawab untuk melaksanakan arahan ialah unit pemprosesan utama (*central processing unit*, CPU). CPU akan melaksanakan arahan yang diterima dan bermulanya proses penyelesaian masalah. Oleh itu, pengatur cara perlu menulis ‘sintaks’ yang spesifik, iaitu peraturan yang diperlukan oleh komputer untuk melaksanakan arahan dengan tepat. Rajah 1.1 dan 1.2 menunjukkan contoh penulisan algoritma dan atur cara bagi memaparkan mesej Hello Malaysia. Rajah 1.1 menunjukkan dua format perwakilan algoritma iaitu pseudokod dan carta alir.

Pseudokod

- 1 Mula
- 2 Baca input "Hello Malaysia"
- 3 Papar output "Hello Malaysia"
- 4 Tamat

Carta alir



Rajah 1.1 Contoh algoritma Hello Malaysia

```
/*Menulis program yang akan memaparkan mesej
 * "Hello Malaysia"
 */

public class HelloMalaysia {
    public static void main (String[] args){
        String x;
        x = "Hello Malaysia";
        System.out.println(x);
    }
}
```

Rajah 1.2 Contoh atur cara untuk Hello Malaysia

1.1.1 Keperluan Penyelesaian Masalah Berstrategi

Pelbagai masalah dalam kehidupan harian merupakan cabaran yang perlu diatasi. Masalah menghalang kita daripada mencapai apa yang diingini, di samping mengganggu kelancaran dan keselesaan hidup. Sesuatu masalah yang rumit memerlukan strategi penyelesaian masalah yang berkesan. Apakah faedah penggunaan strategi dalam penyelesaian masalah?



Anugerah APICTA anjuran *Multimedia Development Corporation* (MDeC) telah menampilkan inovasi dan kecemerlangan ICT kebangsaan sejak tahun 1999. Anugerah ini diadakan untuk mengetengahkan bakat dan juga mengiktiraf kreativiti rakyat tempatan, termasuk kejayaan dan sumbangan bersejarah dalam sebuah industri.



Rajah 1.3 Keperluan strategi dalam penyelesaian masalah

Pembangunan aplikasi komputer berkait rapat dengan strategi penyelesaian masalah dalam pengaturcaraan komputer. Seseorang perlu menguasai strategi penyelesaian masalah supaya dapat memahami masalah dengan mudah bagi mendapatkan strategi penyelesaian yang terbaik.



Khan academy: Apakah itu pengaturcaraan?
<https://goo.gl/2JlmX8>

Komputer boleh digunakan untuk menyelesaikan masalah. Walau bagaimanapun, cara penyelesaian terhadap masalah yang dihadapi haruslah jelas sebelum mengatasinya. Pemikiran komputasional (*computational*) membolehkan kita memahami masalah yang dihadapi dan membangunkan penyelesaian yang munasabah apabila kita menghadapi masalah yang kompleks.



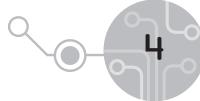
Aplikasi Harian

Penggunaan strategi penyelesaian masalah dalam kehidupan seseorang murid.

'Jumlah kalori yang diambil melebihi kalori yang diperlukan.'

Jumlah kalori harian yang diperlukan oleh remaja yang berumur antara 16 hingga 18 tahun ialah 2600–2800 kcals.

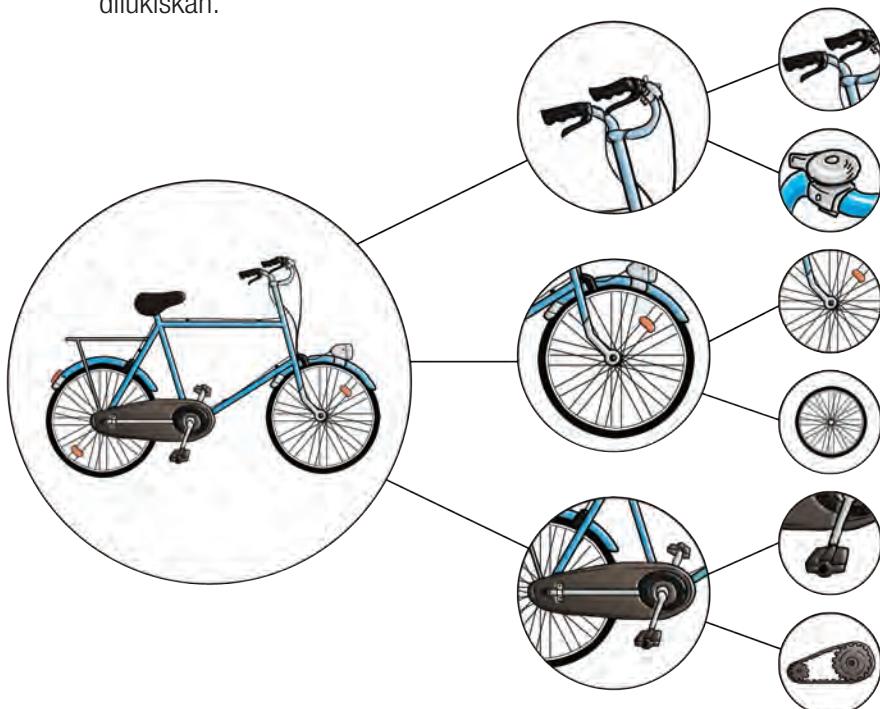
Contohnya, kalori sekeping roti canai ialah 300 kcals. Apakah jenis makanan yang sesuai dimakan supaya kalori yang diperlukan berada dalam lingkungan yang disebutkan?



Empat teknik pemikiran komputasional dalam penyelesaian masalah ialah:

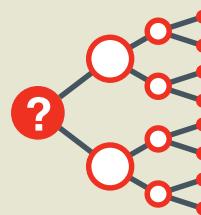
1. Teknik Leraian (Decomposition) – Memecahkan masalah kepada bahagian yang lebih kecil dan terkawal. Sebelum komputer menyelesaikan masalah, masalah dan cara untuk menyelesaikannya mestilah difahami. Penguraian masalah membantu memecahkan masalah kompleks kepada bahagian yang mudah diuruskan. Begitu juga, kita cuba memahami bagaimana sistem yang kompleks berfungsi menggunakan teknik leraian.

Contoh: Memahami cara sebuah basikal berfungsi akan jadi lebih mudah jika seluruh basikal itu dipecahkan kepada bahagian yang kecil. Rujuk rajah di bawah dan labelkan bahagian basikal yang dilukiskan.



Rajah 1.4 Teknik leraian memecahkan basikal kepada bahagian-bahagian basikal

Teknik leraian



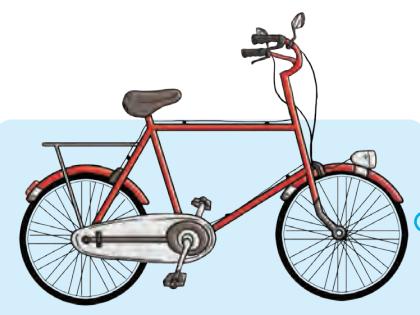
Leraikan masalah dalam aktiviti menggosok gigi. Isikan dengan rajah yang sesuai seperti contoh di atas.

- 2. Pengecaman corak** (*Pattern recognition*) – Mencari persamaan (*similarities*) antara masalah dan dalam masalah. Apabila kita telah meleraikan masalah yang kompleks kepada bahagian yang mudah urus dengan teknik leraian, ini membantu dalam mengenal pasti persamaan atau corak yang ada pada setiap bahagian masalah yang kecil. Corak-corak ini dapat membantu menyelesaikan masalah kompleks dengan lebih efisien.

Bayangkan kita hendak mengetahui cara sesbuah basikal berfungsi. Ciri-ciri basikal ialah mempunyai dua buah roda yang berputar, hendal untuk menguasai hala tuju, pedal untuk menggerakkan basikal dan rantai yang menjadi penyambung supaya basikal dapat berfungsi. Jadi, kita boleh melukis gambar basikal berdasarkan ciri-ciri di atas. Dalam pemikiran komputasional, ciri-ciri ini dikenal sebagai corak. Cuma yang berbeza secara spesifik ialah warna basikal, bunyi loceng atau saiz roda.



Ada basikal mempunyai dua roda kecil, hendal berbakul, pedal dan rangka berwarna merah jambu.



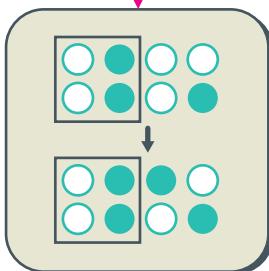
Ada pula yang mempunyai roda yang besar, hendal bercermin, pedal dan rangka berwarna merah.



Basikal juga boleh mempunyai roda yang berlainan saiz, hendal melengkung, pedal dan rangka berwarna hijau.

Dengan mengikut corak dan mengabaikan ciri-ciri spesifik, kita dapat menyelesaikan masalah tentang basikal.

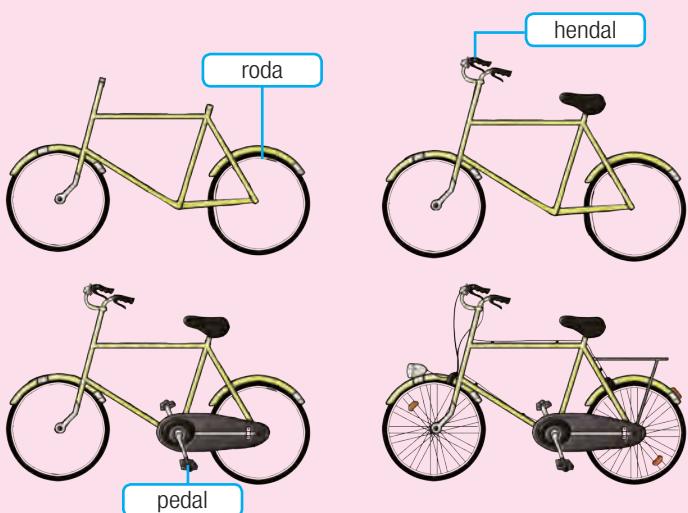
Pengecaman corak



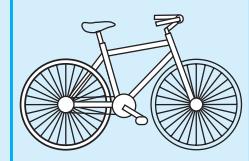
Nyatakan dua ciri yang akan menjadi corak dan dua ciri spesifik yang terdapat dalam aktiviti menggosok gigi.

Dalam pemikiran komputasional, setelah mengetahui ciri-ciri umum untuk masalah menggosok gigi, satu corak telah terbentuk. Apabila kita dapat mengetahui cara menggosok gigi, individu yang lain juga dapat berbuat demikian dengan mengikuti corak-corak tersebut.

3. **Peniskalaan (Abstraction)** – Peniskalaan adalah penting supaya dapat menjana penyelesaian yang tepat kepada masalah yang dihadapi. Contohnya, untuk menghasilkan sebuah basikal, jika tidak meniskalakan ciri-ciri umum dan spesifik, kita berkemungkinan akan menghasilkan basikal yang beroda kecil dan berbakul di hendarinya, iaitu merupakan satu penyelesaian yang tidak tepat. Satu model basikal ialah basikal yang mempunyai dua roda, hendal dan pedal. Ciri spesifik seperti saiz roda, berbakul atau berloceng dan warna rangka hendaklah diabaikan.



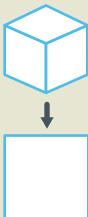
Model ialah perwakilan masalah atau idea umum masalah yang hendak diselesaikan. Sebagai contoh, model basikal dan fungsi yang terbentuk selepas proses peniskalaan.



Setelah mendapat model kepada masalah, kita boleh menggunakan corak di atas untuk melukis basikal dan juga membina algoritma untuk menyelesaikan masalah yang dihadapi.



Peniskalaan



Setelah meniskalakan corak umum dan spesifik, lukis model penyelesaian masalah untuk aktiviti menggosok gigi.

4. **Algoritma (Algorithms)** – Membangunkan penyelesaian langkah demi langkah terhadap masalah yang dihadapi atau mematuhi peraturan untuk menyelesaikan masalah. Algoritma merupakan suatu pelan, iaitu satu set langkah arahan untuk menyelesaikan masalah. Jika anda boleh mengikat tali kasut, membuat secawan kopi, atau menyediakan makanan, anda telah tahu mengikut sesuatu algoritma.

Dalam algoritma, setiap arahan dikenal pasti dan susunan pelaksanaannya dijalankan seperti yang dirancang. Algoritma juga digunakan sebagai titik mula untuk rekaan program komputer dalam bentuk carta alir dan pseudokod.



Uji Minda

Sebagai seorang murid Sains Komputer, pada pendapat anda, mengapakah kita perlukan pemikiran komputasional? Apakah contoh pemikiran komputasional dalam kehidupan sehari-hari anda?

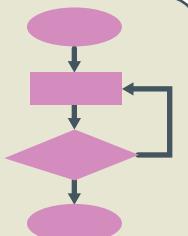
```
Mula
OUTPUT "Mempunyai 2 roda yang berputar?"
INPUT "Ya"
OUTPUT "Hendal untuk menguasai hala tuju?"
INPUT "Ya"
OUTPUT "Pedal dan rantai untuk menggerakkan basikal?"
INPUT "Ya"
OUTPUT "Ini ialah sebuah basikal."
Tamat
```

Rajah 1.5 Contoh pseudokod bagi masalah fungsi sebuah basikal



Pengasas pemikiran komputasional ialah Jeannette M Wing. Beliau telah mentafsirkan pemikiran komputasional sebagai membina kuasa dan proses penyelesaian secara pengkomputeran, sama ada dilaksanakan oleh manusia atau mesin.
Sumber: www.acm.org

Algoritma



Tuliskan pseudokod untuk membina model penyelesaian aktiviti menggosok gigi.

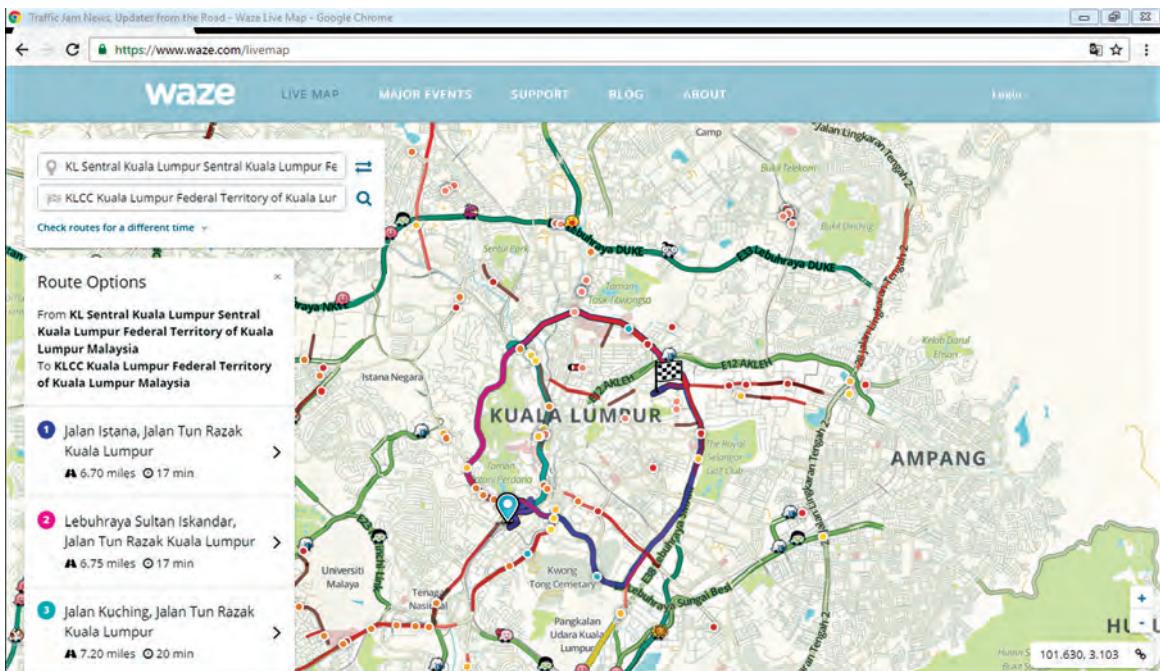


Justifikasikan ciri-ciri pemikiran komputasional dalam penyelesaian masalah

Kelas anda telah ditugaskan mengadakan Kempen Mengurangkan Penggunaan Elektrik di sekolah anda memandangkan bil elektrik telah meningkat dan adalah wajar masalah ini dikaji dan diselesaikan dengan segera. Justifikasikan ciri-ciri pemikiran komputasional dalam menyelesaikan masalah berdasarkan situasi yang diberikan.

- Leraikan masalah peningkatan bil elektrik.
- Kenal pasti ciri-ciri yang terdapat dan bina corak-corak penggunaan atau pembaziran elektrik di semua tempat yang terlibat di sekolah anda.
- Meniskalakan ciri-ciri yang hendak dijadikan model dan ciri-ciri spesifik yang hendak diketepikan.
- Jelaskan dengan ringkas langkah demi langkah penyelesaian yang diambil untuk menyelesaikan masalah di atas.

Pemikiran komputasional membolehkan kita merancang perkara yang hendak dilaksanakan oleh komputer dengan tepat. Sebagai contoh, jika anda hendak pergi ke *Kuala Lumpur Convention Centre* (KLCC) dari KL Sentral, anda perlu merancang perjalanan yang harus diambil, iaitu mengikut jalan yang terdekat dan tidak mengalami kesesakan trafik. Katakan anda menggunakan aplikasi *Waze* untuk sampai ke KLCC. Mula-mula masukkan lokasi yang dikehendaki iaitu KL Sentral ke KLCC di ruang yang disediakan. Anda akan mendapat paparan seperti di bawah.



Rajah 1.6 Paparan Waze dari KL Sentral ke KLCC

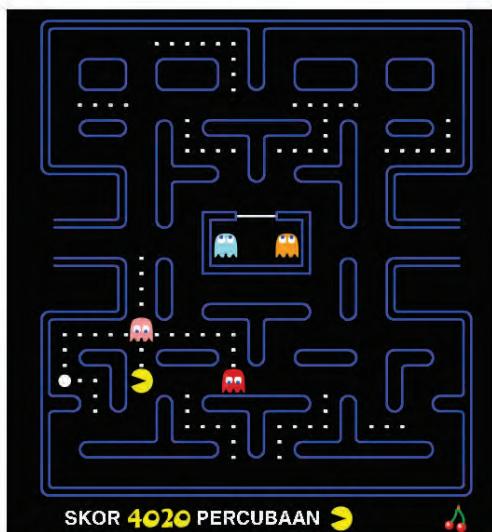
Anda boleh memilih tiga pilihan perjalanan yang dicadangkan oleh *Waze* dengan mengambil kira bilangan rakan yang bersama-sama anda dan mod pengangkutan anda. Dalam kes ini, merancang perjalanan ialah pemikiran komputasional manakala mengikut arahan langkah demi langkah ialah pengaturcaraan.



Get to know Waze
goo.gl/CtP3Cs

Penyelesaian masalah dengan pemikiran komputasional

Untuk menang dalam permainan *Pacman*, kita harus mengetahui jenis item yang perlu dimakan, laluan yang selamat, apa-apa yang perlu dielakkan, dan cara untuk memusnahkan musuh.



- Berdasarkan kepada empat teknik pemikiran komputasional dan pemahaman anda tentang permainan ini, bincangkan dengan rakan anda tentang strategi untuk memenangi permainan ini.
- Namakan dua jenis permainan lain yang memerlukan strategi penyelesaian untuk menang.
- Apakah cara-cara untuk menyelesaikan masalah itu supaya menang dalam permainan yang dinyatakan di (b)?



Meneroka pemikiran pengkomputeran: contoh penggunaan pemikiran pengkomputeran
<https://goo.gl/OrzWXS>



Play Pacman online
goo.gl/ov5vvh



Kerjaya Sains Komputer

Kemahiran dalam pengaturcaraan telah mewujudkan profesi yang berikut:

- Pembangun laman web
- Jurutera perisian
- Pembangun aplikasi mudah alih
- Jurutera rangkaian
- Pengurus IT





Laporan Pasaran Kerja ICT di Malaysia 2015 oleh Persatuan Industri Komputer dan Multimedia Malaysia (PIKOM)
<http://goo.gl/1gUAJX>

Tahukah Anda?

Kos langsung ialah kos untuk aktiviti projek seperti gaji pekerja projek. Kos tak langsung pula merupakan kos untuk aktiviti yang tidak melibatkan secara langsung dengan projek. Contohnya kos keselamatan dan bil telefon.

Tahukah Anda?

Carta PERT (*Program Evaluation Review Technique*) ialah alat pengurusan projek yang diguna untuk menguruskan, menjadualkan dan menyelaraskan aktiviti projek. Carta PERT dapat menunjukkan urutan tugas yang akan dilaksanakan. Tujuannya adalah untuk menentukan masa minimum dan maksimum untuk pelaksanaan aktiviti-aktiviti dalam projek. Perkara ini adalah penting untuk mengelakkan kelewatan dalam menyiapkan projek. Kelewatan menyiapkan projek hanya akan meningkatkan kos.

1.1.2 Ciri-ciri Penyelesaian Masalah Berkesan

Setelah memikirkan strategi untuk menyelesaikan masalah dan melaksanakan strategi tersebut, keputusan yang diperoleh akan menggambarkan keberkesanan strategi penyelesaian masalah yang telah diambil. Misi mencari penyelesaian melibatkan tiga ciri penting iaitu kos, masa dan sumber.

Kos

Kos ditakrifkan sebagai harga yang perlu dibayar untuk memperoleh, mengeluarkan dan menyenggara, biasanya berupa wang, masa, tenaga, dan perbelanjaan. (*Kamus Online Dewan Bahasa*)

Persoalan yang selalu dibincangkan oleh jawatankuasa yang melaksanakan projek ialah,

- berapakah kos projek?
- adakah kos tersebut akan menambah produktiviti?
- bolehkah kos tersebut menyelesaikan masalah yang dihadapi?

Projek yang dilaksanakan mengikut tempoh yang normal ialah projek yang dilaksanakan mengikut masa dan kos yang diperuntukkan. Manakala kos siap masa tercepat (*crashing cost*), ialah usaha yang maksimum untuk menyelesaikan projek dalam masa terpendek. Kos siap masa tercepat (*crashing cost*) meningkat apabila tempoh masa projek menurun.

Masa

Penyelesaian masalah secara berkesan yang melibatkan masa merujuk sesuatu projek yang disiapkan mengikut masa yang telah ditetapkan. Faktor masa dalam penyelesaian masalah ini memainkan peranan yang penting supaya semua perancangan awal dan pelaksanaan penyelesaian masalah dapat dilakukan mengikut tempoh yang ditetapkan. Aktiviti pelaksanaan yang tertunda atau lambat disiapkan akan meningkatkan kos. Selain itu, penyelesaian masalah yang memerlukan masa tersingkat seperti keperluan menyiapkan projek dalam jangka masa terhad, juga akan melibatkan peningkatan kos. Contoh, penambahan bilangan pekerja supaya sesuatu projek dapat disiapkan dalam masa yang dikehendaki.

Carta Gantt merupakan jadual perancangan yang dapat melihat perkembangan semua aktiviti projek yang dipaparkan mengikut tempoh masa. Sebelah kiri carta ialah senarai aktiviti manakala bahagian atas ialah tempoh masa. Setiap aktiviti diwakili oleh satu bar, panjang bar mewakili tarikh mula dan tarikh akhir aktiviti tersebut. Penggunaan carta Gantt penting dalam penyelesaian masalah kerana maklumat dapat dilihat sejelas mungkin.

- senarai aktiviti projek
- bila setiap aktiviti bermula dan berakhir

- (c) tempoh masa setiap aktiviti
- (d) aktiviti-aktiviti yang bertindih
- (e) tarikh mula dan tarikh projek dijangka tamat

Lihat contoh carta Gantt di bawah, bolehkah anda berikan maklumat yang anda dapat lihat sepantas lalu?

Senarai Aktiviti	Jan-17	Feb-17	Mar-17	Apr-17	May-17
1. Menyediakan soal selidik					
2. Mengedarkan soal selidik dan mendapatkan data					
3. Menemu ramah pengguna					
4. Menganalisis data					
5. Membuat rumusan					

Rajah 1.7 Contoh carta Gantt yang mudah



What is a Gantt chart?
<http://goo.gl/fICzss>



Inovasi Sains
Komputer



Teaching Computers
to Understand Human
Language
<https://goo.gl/NRMbzcz>



3

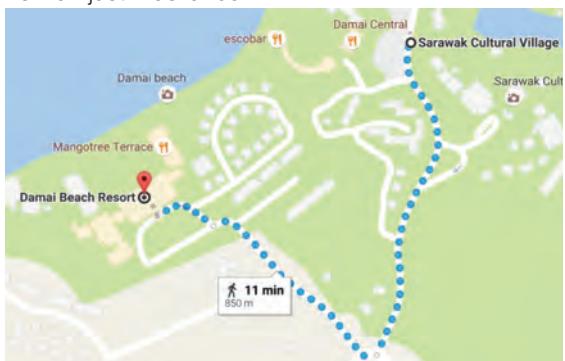
Ciri-ciri penyelesaian masalah berkesan berdasarkan kos dan masa

Kerja Berkumpulan

Sepuluh orang ahli Persatuan Pencinta Alam Sekitar sekolah anda ingin mengadakan lawatan sambil belajar dari Resort Pantai Damai ke Kampung Budaya Sarawak. Jarak di antara kedua-dua tempat ialah 850 m. Jika mereka bertolak dari Resort Pantai Damai ke Kampung Budaya Sarawak secara berjalan kaki, mereka akan memakan masa 11 minit, manakala jika mereka menaiki kereta hanya 3 minit.

Bincangkan kesan pilihan cara pengangkutan mereka dari segi kos dan masa dari Resort Pantai Damai ke Kampung Budaya Sarawak.

Pada pendapat anda, apakah pilihan yang sesuai untuk penyelesaian masalah di atas?
Berikan justifikasi anda.

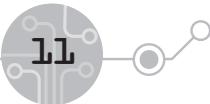


Laluan pejalan kaki



Laluan kereta

Sumber: Google Maps



Sumber

Sumber didefinisikan sebagai stok atau wang, bahan-bahan mentah, staf dan aset lain yang boleh digunakan dalam organisasi supaya dapat berfungsi dengan efektif (*Kamus Online Dewan Bahasa*). Dalam perancangan penyelesaian masalah, semua penyelesaian memerlukan sumber. Sumber diperlukan untuk menjana hasil atau perkhidmatan. Contoh sumber adalah seperti sumber kewangan, sumber manusia dan sumber teknologi. Perancangan sumber adalah tanggungjawab pihak pengurusan dan perancangan sumber yang baik adalah untuk mendapatkan keputusan yang optimum. Pembaziran sumber dapat dielakkan dengan perancangan yang rapi. Masalah yang dihadapi apabila kekurangan sumber ialah akan melambatkan masa untuk menyiapkan projek, seterusnya mengakibatkan peningkatan kos projek.

Penjadualan sumber projek perlu diadakan untuk mengelakkan berlaku kekurangan supaya tidak menjelaskan penyelesaian masalah yang dihadapi. Langkah-langkah yang berikut perlu diberikan perhatian untuk tujuan ini.

- 1 Memastikan ada pegawai atau unit yang memantau penjadualan sumber.
- 2 Menggunakan sistem baris gilir mengikut keutamaan projek (*first come first served*).
- 3 Memastikan keperluan projek diuruskan dengan teliti dengan mengambil kira sumber sedia ada dan sumber kewangan yang diperlukan.
- 4 Mengadakan proses penjadualan aktiviti supaya seimbang dan boleh dilaksanakan sepanjang projek dijalankan.



4

Ciri-ciri penting untuk menyelesaikan masalah

Kerja Berkumpulan

Situasi 1:

Seorang pengatur cara memerlukan empat bulan untuk menyiapkan tugas yang diamanahkan kepadanya tetapi akan cuti bersalin dalam tempoh masa tersebut. Apakah langkah yang harus diambil oleh pengurus projek itu untuk menyelesaikan masalah di atas? Justifikasikan jawapan anda dari segi kos yang terlibat, tempoh masa untuk menyiapkan projek dan juga sumber manusia yang terlibat.

Gunakan tip di bawah untuk membantu anda menyelesaikan masalah tersebut.

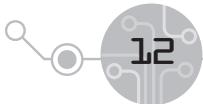
- (a) Tempoh cuti bersalin ialah 2-4 bulan
- (b) Tindakan pengatur cara sebelum beliau mula bercuti
- (c) Tindakan yang boleh diambil oleh pengurus projek

Situasi 2:

Syarikat X sedang membina aplikasi untuk kehadiran murid yang perlu disiapkan dalam tempoh masa tiga bulan. Pada masa yang sama, syarikat ini juga telah mendapat projek baharu iaitu membina aplikasi mencetak tiket bas yang perlu disiapkan dalam tempoh masa yang sama. Cadangkan langkah penyelesaian yang boleh diambil dalam pengurusan kos yang terlibat dan sumber tenaga mahir memandangkan kedua-dua projek ini perlu disiapkan dalam tempoh yang sama.

Gunakan tip di bawah untuk membantu anda menyelesaikan masalah tersebut.

- (a) Senaraikan keperluan kedua-dua projek
- (b) Senaraikan aktiviti dan tempoh dalam carta Gantt
- (c) Tindakan yang boleh diambil oleh pengurus projek

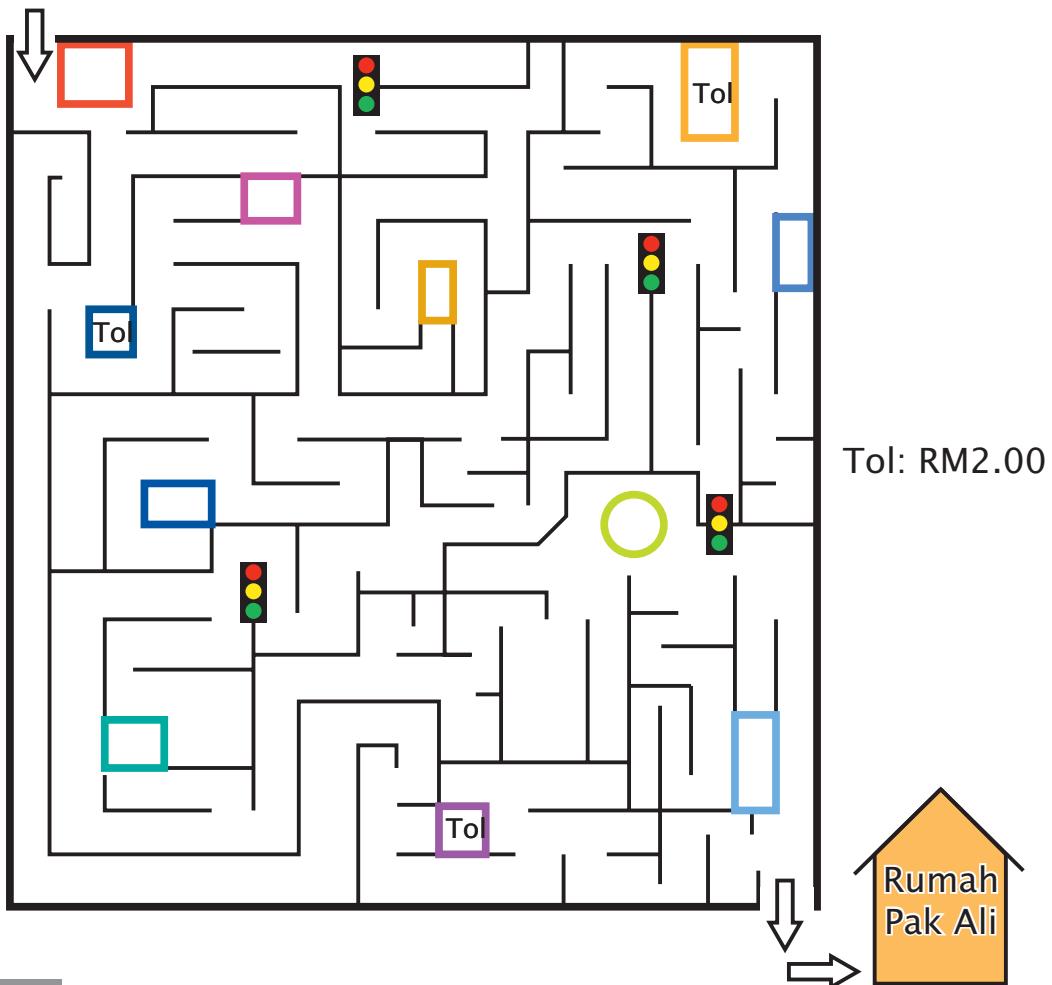




Pak Ali hendak pulang ke rumahnya dari bandar. Pak Ali mempunyai tiga pilihan jalan untuk pulang ke rumahnya. Bantu Pak Ali mencari jalan pulang dengan melorek jalan yang dicadangkan.

- Carikan ketiga-tiga jalan yang boleh membawa Pak Ali pulang ke rumahnya.
- Bincangkan dari segi kos (Tol), masa (lampau isyarat) dan sumber (jarak) ketiga-tiga jalan yang telah dicari.
- Justifikasikan pilihan jalan yang akan digunakan oleh Pak Ali dari aspek kos, masa dan sumber.

Pak Ali



Mari bermain permainan pengaturcaraan dengan menggunakan kod.
<https://goo.gl/LrEUhH>

Tahukah Anda ?

Generik bermaksud umum atau berkaitan dengan ciri-ciri khusus yang dimiliki oleh sesuatu kelompok.
Sumber: Kamus Online Dewan Bahasa

1.1.3 Proses Penyelesaian Masalah

Proses penyelesaian masalah melibatkan langkah-langkah yang sesuai dan bergantung kepada jenis masalah yang ingin diselesaikan. Setiap masalah yang ingin diselesaikan perlu melalui beberapa proses, iaitu mengikut struktur model generik sebagai panduan. Pengaturcaraan dapat membantu murid berfikir secara kreatif dan inovatif serta membantu mereka untuk menyelesaikan masalah dan melaksanakan semua proses dengan betul.

Model generik penyelesaian masalah dalam Rajah 1.8 terdiri daripada lapan proses. Setiap proses akan dijelaskan dengan persoalan yang harus difikirkan bersama-sama.



Rajah 1.8 Proses penyelesaian masalah

Proses penyelesaian masalah dalam pengaturcaraan adalah seperti yang berikut:

1 Mengumpulkan dan menganalisis data

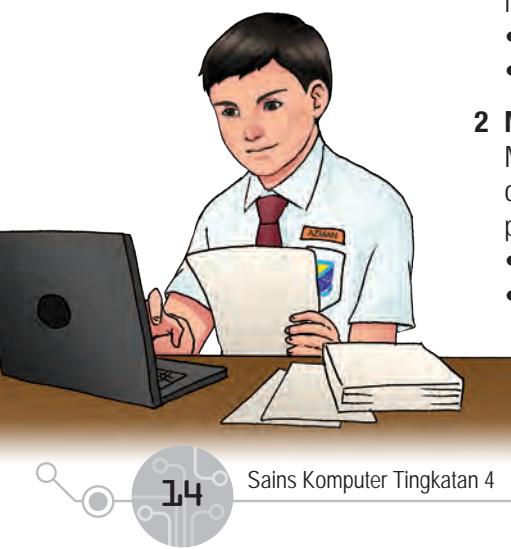
Murid perlu mengumpul data tentang punca dan skop masalah serta mengenal pasti hal yang berikut dalam sesuatu situasi.

- Sejauh manakah masalah tersebut memberikan kesan?
- Bilakah masalah itu perlu diselesaikan?

2 Menentukan masalah

Murid perlu mengenal pasti masalah yang paling utama yang perlu diselesaikan. Kemudian, masalah yang kedua juga perlu dikenal pasti sebelum mengenal pasti tahap pemahaman sedia ada murid.

- Adakah masalah ini merupakan masalah yang utama?
- Apakah masalah lain yang dapat dikesan?



Tahukah Anda?

Terdapat pelbagai model penyelesaian masalah secara kreatif, antaranya ialah Model Penyelesaian Masalah secara Kreatif oleh Osborn dan Parnes. Layari laman sesawang yang berikut untuk mengetahui lebih lanjut tentang model tersebut.



Model Penyelesaian Masalah
<http://goo.gl/pCKdx>



Proses Penyelesaian Masalah
<https://goo.gl/UD7NRt>

3 Menjana idea – menghapuskan halangan

Murid perlu menyenaraikan beberapa idea yang dapat digunakan untuk menyelesaikan setiap masalah yang telah dikenal pasti.

- Sejauh manakah idea yang disenaraikan dapat menyelesaikan masalah yang dihadapi?

4 Menjana penyelesaian

Murid hendaklah menyenaraikan idea atau langkah semasa merancang penyelesaian. Guru perlu mengenal pasti murid telah menjana suatu rancangan penyelesaian yang sesuai.

- Adakah idea penyelesaian ini telah dibincangkan dengan jelas?

5 Menentukan tindakan

Murid hendaklah membuat pilihan yang terbaik daripada senarai idea yang telah dibuat.

- Adakah pilihan yang lain harus diberikan perhatian?
- Dapatkah idea ini menyelesaikan masalah yang dihadapi?

6 Melaksanakan penyelesaian

Murid perlu menggunakan pelbagai alat dan teknik yang telah dipilih untuk melaksanakan penyelesaian. Fokus hendaklah ditumpukan kepada tindakan yang boleh menyumbang kepada penyelesaian masalah.

- Bolehkah tindakan ini berjaya dilaksanakan?

7 Membuat penilaian

Penilaian akan dilaksanakan terhadap langkah-langkah penyelesaian. Jika tidak menepati tahap yang memuaskan atau tidak menyelesaikan masalah, murid perlu mencari tindakan yang lain.

- Adakah hasil yang hendak dicapai memuaskan?

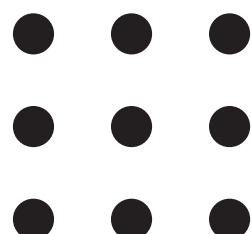
8 Membuat penambahbaikan

Setiap penyelesaian yang telah dilaksanakan perlu ditambah baik jika masih terdapat kekurangan dan mengikut keperluan.

- Apakah penambahbaikan yang boleh anda cadangkan?



Untuk merangsang pemikiran anda tentang proses penyelesaian masalah, fikirkan cara penyelesaian terhadap masalah yang berikut:
Sambungkan kesemua titik dengan hanya menggunakan empat garisan sahaja tanpa mengangkatkan pena anda.



CONTOH (1)

Baca surat di bawah dengan teliti. Anda sebagai pegawai di Majlis Perbandaran Sandakan (MPS) telah ditugaskan menyelesaikan isu tersebut. Berdasarkan langkah-langkah penyelesaian masalah langkah 1-8, selesaikan masalah yang dihadapi oleh mereka. Jelaskan setiap langkah dan sediakan contoh yang sesuai, cari langkah penyelesaian dan sediakan jawapan anda dalam bentuk surat kepada Encik Ahmad dan rakan-rakannya.

AHMAD BIN HARON
23, Jalan Maju,
90700 Sandakan.

Kepada,
Pegawai
Majlis Perbandaran Sandakan

Tuan,

Bonggol-bonggol Jalan (Speed humps) di Jalan Maju, Sandakan

Perkara di atas dirujuk.

2. Saya Ahmad bin Haron mewakili 10 orang rakan saya yang tinggal di sepanjang Jalan Maju yang sepanjang 1.8 km, ingin mencadangkan kepada Majlis Perbandaran Sandakan, agar bonggol-bonggol jalan yang dibina di Jalan Maju diratakan atau dikeluarkan atas alasan-alasan seperti yang berikut:

- Bonggol jalan melambatkan dan mewujudkan kesesakan trafik
- Meningkatkan kemalangan di kawasan bonggol jalan
- Semasa hujan, air bertakung di bonggol dan membahayakan pengguna motosikal dan basikal
- Merosakkan ‘shock absorbers’ kereta
- Mengakibatkan pencemaran udara dan bunyi di kawasan kediaman kami
- Melambatkan ambulans

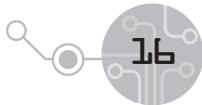
3. Untuk makluman tuan, kami terpaksa melalui sembilan bonggol setiap kali untuk keluar dari rumah dan sembilan kali lagi apabila kembali ke rumah. Setakat tahun ini sahaja telah berlaku lima kes kemalangan jalan raya kerana kereta tidak sempat berhenti di bonggol jalan.

4. Saya dan rakan-rakan berharap MPS akan menyelesaikan masalah ini dengan segera.

Sekian, terima kasih.

Yang benar,

Ahmad bin Haron & rakan-rakan



Penyelesaian:

Berdasarkan langkah-langkah penyelesaian masalah 1-8:

1 Kumpulkan dan analisis data

Merujuk surat daripada Encik Ahmad:

- Masalah bonggol jalan terlalu banyak dan lima kes kemalangan jalan raya telah berlaku.
- Perlu diselesaikan dengan segera.

2 Tentukan masalah

Merujuk surat daripada Encik Ahmad, masalah lain yang dihadapi ialah seperti yang berikut:

- Melambatkan perjalanan, mewujudkan kesesakan trafik, meningkatkan kemalangan jalan raya, air bertakung, merbahayakan pengguna mototsikal dan basikal, merosakkan kereta, melambatkan ambulans, meningkatkan pencemaran udara dan bunyi.

3 Jana idea (hapus halangan)

Idea daripada Pegawai MPS:

- Melambatkan perjalanan (tidak boleh diterima oleh sebab keutamaan diberi dari aspek keselamatan jalan raya).
- Mewujudkan kesesakan trafik (tidak boleh diterima kerana kesesakan jalan raya berlaku di mana-mana jalan utama dan pada masa tertentu sahaja).
- Meningkatkan kemalangan jalan raya (boleh diterima, papan peringatan atau lampu berkelip akan di pasang di kawasan bonggol).
- Air bertakung merbahayakan pengguna mototsikal dan basikal (akan diselesaikan dengan membaiki saliran air di kawasan bonggol dan rujuk peringatan).
- Merosakkan kereta (tidak boleh diterima kerana jika pemandu memandu di had laju yang ditentukan).
- Melambatkan ambulans (boleh diterima dengan mengurangkan bonggol).
- Pencemaran udara dan bunyi (tidak dapat dielakkan sebab jalan utama, namun boleh menanam pokok di sepanjang jalan tersebut.)

4 Jana penyelesaian

Tindakan yang akan diambil oleh MPS:

- Menanam pokok-pokok di sepanjang jalan tersebut.
- Membaiki saliran air di kawasan bonggol.
- Pemasangan papan peringatan dan tiang lampu berkelip.
- Bilangan bonggol akan dikurangkan.

5 Tentukan tindakan

Perbincangan boleh diadakan antara pihak MPS bersama Encik Ahmad dan rakan-rakannya untuk menentukan tindakan yang perlu diambil dan tindakan susulan yang bergantung kepada kos, masa dan sumber. Boleh bina carta Gantt dan senaraikan penyelesaian berdasarkan poin nombor 4. Contoh pemasangan papan peringatan dan tiang lampu berkelip boleh dijalankan dahulu.

6 Laksanakan tindakan

Tindakan memasang papan peringatan dan tiang lampu melibatkan kos dan sumber. Justifikasi kos dan juga kesan kepada masalah yang dihadapi perlu dibuat.

7 Buat penilaian

Penilaian perlu dibuat sama ada pemasangan papan peringatan dan tiang lampu memberi impak kepada pengguna jalan raya dan adakah kes kemalangan telah berkurangan.

8 Buat penambahbaikan

Berdasarkan tindakan yang telah diambil, adakah tindakan tersebut telah menyelesaikan masalah yang dihadapi oleh penduduk Jalan Maju? Jika ya, penambahbaikan boleh dibuat kepada penyelesaian yang sedia ada. Jika tidak, perbincangan dan idea baharu boleh dikemukakan.



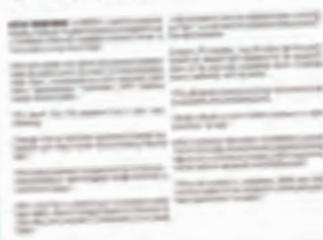
Artikel ini dipetik daripada akhbar tempatan. Dapatkan maklumat yang lebih lanjut melalui laman sesawang yang berikut:

<http://www.thestar.com.my/metro/community/2016/07/05/dengue-danger-in-carpark-shah-alam-facility-infested-with-mosquitoes-and-littered-with-rubbish/>

Berdasarkan maklumat yang diperoleh, laksanakan proses penyelesaian masalah yang meliputi lapan proses. Cadangkan penyelesaian masalah yang munasabah kepada pihak Majlis Perbandaran tempatan untuk menangani masalah tersebut. Persembahkan hasil kumpulan anda dengan menggunakan perisian *PowerPoint*, *Google Slides* atau peta *i-Think*.

5 Julai 2016

Nyamuk mengerumuni sampah pada air bertakung



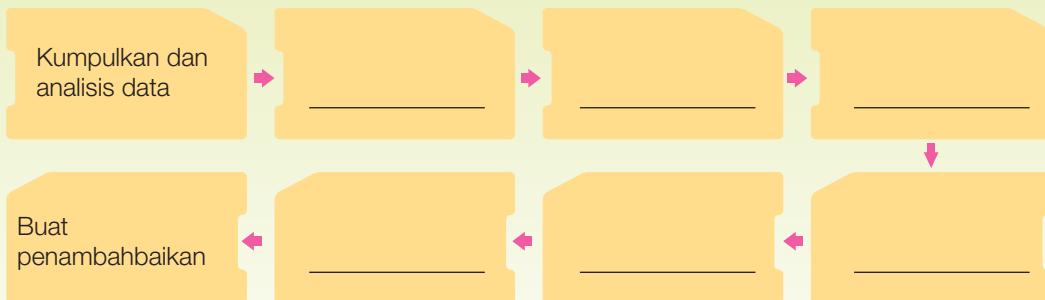
Latihan Formatif 1.1

1 Senaraikan empat teknik pemikiran komputasional dengan menggunakan peta *i-Think*.

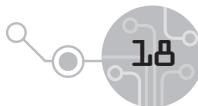
2 Jelaskan istilah yang berikut dengan contoh yang bersesuaian.

Istilah	Perincian	Contoh
Kos langsung		
Kos tak langsung		
Kos masa tercepat		

3 Lengkapkan langkah-langkah untuk setiap proses penyelesaian masalah dalam carta alir yang berikut. Jelaskan setiap langkah dengan memberikan contoh aktiviti yang dijalankan.



4 Tonton video di <http://edtechreview.in/trends-insights/insights/2376-coding-computer-programming-for-kids>. Video ini memaparkan pengalaman orang-orang terkenal dari dunia teknologi. Apakah idea dalam video ini yang dapat memotivasi anda dalam penggunaan pengaturcaraan sebagai penyelesaian masalah?



- 5** Guru tingkatan, Puan Sabariah, kelas 4 Mesra ingin merancang satu lawatan pada cuti sekolah yang akan datang. Persoalan yang harus diberikan pertimbangan semasa merancang lawatan itu ialah:
- (a) Tempat/lokasi lawatan
 - (b) Kos yang diperlukan
 - (c) Cara untuk mendapatkan penajaan sebagai satu sumbangan kecil untuk lawatan
 - (d) Ahli-ahli jawatankuasa
 - (e) Tugas ahli-ahli jawatankuasa dan murid
 - (f) Jenis aktiviti yang hendak dilakukan ketika lawatan
 - (g) Tempoh untuk merancang lawatan tersebut dan masa untuk aktiviti ketika lawatan

Berdasarkan persoalan yang telah disenaraikan, anda sebagai Puan Sabariah dikehendaki membuat keputusan dan mendapat persetujuan bersama-sama murid 4 Mesra. Gunakan teknik pemikiran komputasional untuk menyelesaikan masalah tersebut dan jelaskan penyelesaian itu secara terperinci. Anda boleh menggunakan komputer untuk mengumpulkan dan menganalisis data.

- 6** Fikirkan beberapa rutin yang anda jalankan sebagai seorang murid. Catatkan langkah-langkah anda untuk menjalankan rutin tersebut. Dalam setiap langkah yang telah dicatatkan, nyatakan langkah penyelesaian yang dipilih dan justifikasikan sebab anda memilih cadangan tersebut. Rujuk contoh yang berikut sebagai panduan.

Bersarapan sebelum pergi ke sekolah.

- (a) Bersarapan di rumah atau di kantin sekolah
- (b) Pilihan:
 - (i) Bersarapan di rumah: masak air, buat minuman dan makan biskut.
 - (ii) Bersarapan di kantin sekolah: tidak perlu membuat persediaan.
- (c) Justifikasi:
 - (i) Bersarapan di rumah: dapat menjimatkan duit.
 - (ii) Bersarapan di kantin sekolah: sediakan wang untuk belanja.

- 7** Sebagai seorang pengatur cara perintis (*naïve programmer*), apakah ciri-ciri yang dapat menggambarkan anda sebagai seorang pengatur cara yang dapat menyelesaikan masalah secara berkesan? Apakah kemahiran yang diperlukan?

-  **8** Kaji data mengenai gejala ponteng sekolah murid tingkatan 4 di sekolah anda. Tentukan masalah utama yang menyumbang kepada gejala tersebut dan cadangkan tiga cara penyelesaian yang boleh diambil. Berikan justifikasi untuk setiap cadangan.

-  **9** Kita sering membaca akhbar tempatan tentang berita kemalangan jalan raya. Layari Internet untuk memperoleh berita tentang satu kes kemalangan jalan raya. Ceritakan kes tersebut secara ringkas.

- (a) Pada pendapat anda, apakah sebab yang berkemungkinan menyebabkan kemalangan tersebut? Berikan justifikasi setiap pendapat yang dikemukakan.
- (b) Cadangkan langkah-langkah penyelesaian yang boleh diambil oleh pihak kerajaan dan pemandu untuk menyelesaikan masalah yang anda nyatakan di (a).
- (c) Kerajaan telah menjalankan pelbagai program seperti Ops Pacak, Ops Statik, Ops Sikap dan Ops Selamat untuk mengurangkan kemalangan di jalan raya, khasnya pada musim perayaan. Nyatakan langkah-langkah yang telah diambil dalam program tersebut.
- (d) Cadangkan langkah-langkah penambahbaikan terhadap program di (c) yang dapat membantu dalam menyelesaikan masalah kemalangan jalan raya dengan lebih berkesan.



STANDARD PEMBELAJARAN

1.2.1 Menggunakan algoritma untuk menyatakan penyelesaian kepada masalah

1.2.2 Membina algoritma penyelesaian masalah dengan menggunakan struktur kawalan dalam pengaturcaraan (urutan, pilihan dan pengulangan)

1.2.3 Menguji dan membaiki ralat dalam algoritma

1.2.4 Mengesan nilai pemboleh ubah pada setiap tahap dalam algoritma

1.2.5 Mengenal pasti output yang betul dari algoritma berdasarkan input yang diberikan

1.2.6 Menterjemah algoritma ke dalam bahasa pengaturcaraan



What is an algorithm?
<http://goo.gl/pEVOo3>

Algoritma

Algoritma ialah satu set arahan untuk menyelesaikan masalah. Arahan-arahan terperinci yang dapat diikuti oleh pembaca sehingga tugas terlaksana dengan jayanya ialah suatu bentuk algoritma dalam kehidupan harian. Dalam kehidupan harian, terdapat banyak contoh yang boleh dikaitkan dengan algoritma. Contohnya, seorang ibu boleh merujuk arahan-arahan untuk persediaan memasak dalam buku resepi.

Algoritma dapat dihalusi dengan menambahkan butiran. Contoh-contoh algoritma yang lain dalam kehidupan harian seperti berikut:

Contoh 1 Menjerang air



- 1 Mula
- 2 Isi air ke dalam cerek
- 3 Letakkan cerek di atas dapur gas
- 4 Hidupkan api
- 5 Tunggu sehingga air mendidih
- 6 Matikan api
- 7 Tamat

Contoh 2 Menaiki lif



- 1 Mula
- 2 Tekan butang
- 3 Tunggu lif
- 4 Masuk ke dalam lif dan tekan nombor aras
- 5 Tunggu sehingga pintu lif terbuka di aras yang dikehendaki
- 6 Keluar lif
- 7 Tamat

Contoh 3

Membeli air minuman tin daripada mesin layan diri

- 1 Mula
- 2 Masukkan wang yang secukupnya
- 3 Tekan butang air minuman tin yang dipilih
- 4 Ambil air minuman tin
- 5 Tamat



Perhatikan Contoh 3. Adakah butiran algoritma cukup jelas? Jika tidak, anda perlu memperhalus algoritma tersebut dengan idea untuk menyelesaikan masalah tersebut. Misalnya, arahan (2) dan (4) memerlukan butiran tambahan.

Penghalusan arahan (2):

- (i) Masukkan wang syiling atau wang kertas di slot wang yang disediakan
- (ii) Perhatikan baki wang yang diperlukan
- (iii) Jika baki masih belum mencapai sifar, masukkan baki wang yang dikehendaki

Penghalusan arahan (4):

- (i) Tunggu air minuman tin keluar di ruangan yang disediakan
- (ii) Ambil air minuman tin tersebut
- (iii) Tunggu baki wang (jika ada) di ruangan yang disediakan

Berdasarkan contoh-contoh algoritma dalam kehidupan harian yang telah diberikan, arahan dapat dikenal pasti mengikut urutan tertentu, mempunyai butiran yang jelas dan boleh dilaksanakan dalam waktu yang terbatas, yakni mempunyai permulaan dan pengakhiran.



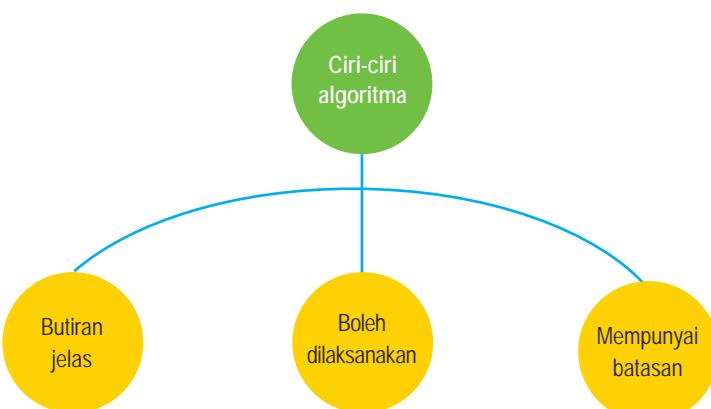
Uji Minda

Mesin layan diri boleh mengesan wang syiling yang palsu. Pada pendapat anda, bagaimanakah mesin layan diri berfungsi untuk mengesan wang palsu tersebut dengan menggunakan algoritma?



MalaysiaKu

Pelan Transformasi Ekonomi melibatkan bidang-bidang pendidikan dalam IT ataupun Sains Komputer. Melalui *Malaysia Digital Economy Corporation (MDEC)*, kerajaan telah berinisiatif untuk mewujudkan koridor teknologi tinggi di Malaysia.



Rajah 1.9 Ciri-ciri wajib algoritma



Aplikasi Harian

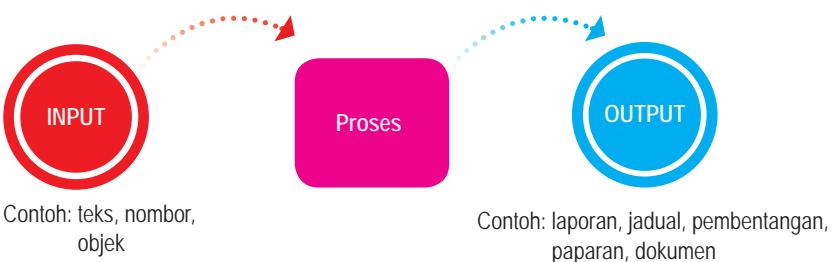
Tahukah anda bahawa algoritma telah digunakan dalam pelbagai aspek kehidupan? Contohnya, panduan untuk menuju ke suatu destinasi, resepi memasak, manual memasang perabot dan manual operasi gelombang mikro.

Arahan-arahan untuk menghasilkan algoritma berdasarkan ciri-ciri pada Rajah 1.9 adalah seperti yang berikut:

- 1 Analisis masalah yang perlu diselesaikan. Kenal pasti input, matlamat proses dan output.
- 2 Tulis arahan-arahan umum untuk memproses input kepada output.
- 3 Analisis setiap arahan umum untuk menilai kejelasan arahan itu sehingga boleh dilaksanakan.
 - a Jika kurang jelas, halusi arahan umum dengan menulis arahan-arahan yang terperinci.
 - b Untuk setiap arahan terperinci, buat penilaian sama ada arahan terperinci masih perlu dihalusi atau tidak.
- 4 Buat penghalusan bagi setiap arahan umum yang masih kurang jelas.
- 5 Arahan terakhir mestilah menghasilkan output yang dikehendaki oleh proses dan perkara ini ialah batasan algoritma yang direka cipta.

1.2.1 Penggunaan Algoritma untuk Menyelesaikan Masalah

Komputer ialah alat elektronik yang berupaya menerima dan memproses data input untuk menghasilkan maklumat sebagai output serta menyimpan kedua-dua data dan maklumat. Salah satu ciri komputer ialah kebolehan menyimpan dan menggunakan pelbagai atur cara untuk melaksanakan proses komputer. Semua atur cara komputer ialah perisian yang “menghidupkan” perkakasan komputer dengan memproses semua input kepada output.



Rajah 1.10 Konsep Input-Proses-Output (IPO) untuk perisian komputer

Apakah perkaitan atur cara komputer dengan algoritma? Atur cara komputer ialah algoritma komputer yang ditulis dalam bahasa pengaturcaraan yang boleh dilaksanakan oleh komputer.

Rajah 1.10 menunjukkan ilustrasi perisian komputer sebagai model Input-Proses-Output (IPO). Sistem komputer menerima Input daripada pengguna. Input ialah nombor, aksara, perkataan, suara atau gambar yang



How to design a good flowchart?
<https://goo.gl/E2jfoU>



dimasukkan oleh pengguna melalui papan kekunci. Proses mengubah input menggunakan formula matematik atau logik untuk menghasilkan output. Output ialah nilai yang dipaparkan pada skrin.



Kerjaya Sains Komputer

CONTOH (1)

Adam mengambil gambar keluarganya dan menghantar gambar tersebut kepada datuk dan nenek di kampung dengan menggunakan telefon pintar. Apakah input, proses dan output untuk telefon pintar tersebut?

Penyelesaian:

Input	Gambar keluarga yang diambil oleh Adam.
Proses	Proses menghantar gambar menggunakan perkakasan (<i>hardware</i>) dan perisian.
Output	Gambar yang diterima oleh datuk dan nenek.

Analisis IPO

Sebelum menulis algoritma untuk suatu masalah, analisa IPO perlu dibuat. Analisis IPO adalah untuk mengenal pasti data input, proses untuk mengubah nilai data kepada maklumat dan paparan output maklumat setelah proses.

Untuk menganalisis masalah, carta Input-Proses-Output (IPO) boleh digunakan. Lihat Rajah 1.11.

Input	
Proses	
Output	

Rajah 1.11 Carta IPO

Langkah-langkah membuat analisis IPO adalah:

- 1 Input: Harus mengenal pasti data yang perlu dibaca daripada pengguna atau persekitaran.
- 2 Proses: Langkah-langkah ataupun rumusan untuk memproses data input kepada output.
- 3 Output: Harus mengenal pasti output yang dikehendaki, yakni apa yang perlu dipaparkan pada skrin di akhir arus cara.



MalaysiaKu

Pada April 2016, pasukan Malaysia dari *Malaysian Integrated Medical Professional Association* (Mimpa) yang diketuai oleh Dr Dhesi Baha Raja telah berjaya memenangi anugerah di peringkat antarabangsa iaitu *Pistoia Alliance Life Science Award* kerana telah membangunkan sebuah aplikasi algoritma untuk mengesan wabak denggi. Anugerah ini dianjurkan oleh Pistoia Alliance of King's College di London. Sumber:
www.thestar.com.my





Uji Minda

Tahukah anda bagaimana untuk menambah nilai prabayar untuk telefon bimbit? Bina analisis IPO berdasarkan situasi yang diberikan.

CONTOH (2)

Sebuah perisian ringkas meminta pengguna memasukkan tahun kelahiran. Perisian kemudiannya menentukan dan memaparkan umur di skrin.

Penyelesaian:

Input	Tahun_kelahiran
Proses	<ol style="list-style-type: none"> 1. Baca input, tahun_kelahiran 2. Dapatkan tahun semasa daripada sistem komputer, tahun_semasa 3 Umur = tahun_semasa - tahun_kelahiran
Output	Umur

Langkah-langkah terperinci dalam proses komputer diberikan oleh algoritma. Algoritma ialah idea atau cara penyelesaian masalah dalam bentuk abstrak berdasarkan arahan-arahan asas. Arahan-arahan asas ini merujuk ciri-ciri dan struktur kawalan yang sedia ada dalam bahasa pengaturcaraan. Jika suatu proses itu tidak dapat diuraikan dengan arahan-arahan asas komputer, proses tersebut berkemungkinan tidak sesuai untuk dilaksanakan oleh komputer.

Perwakilan Algoritma

Terdapat dua format perwakilan algoritma, iaitu pseudokod dan carta alir.

- **Pseudokod** ialah senarai struktur kawalan komputer yang ditulis dalam bahasa pertuturan manusia dan mempunyai nombor turutan.
- **Carta alir** adalah alternatif kepada pseudokod menggunakan simbol grafik untuk mewakili arahan-arahan penyelesaian.

Pseudokod

Pseudokod bukanlah bahasa pengaturcaraan komputer. Arahan ditulis dalam bahasa pertuturan harian. Setiap arahan ialah ungkapan matematik, ungkapan logik, penggunaan struktur kawalan ataupun penggunaan fungsi komputer (contohnya membaca input). Setiap arahan diletakkan dalam baris baru yang diberikan nombor siri.

Langkah-langkah menulis pseudokod:

- 1 Tulis kenyataan MULA.
- 2 Baca INPUT.
- 3 Proses data menggunakan ungkapan logik atau matematik.
- 4 Papar OUTPUT.
- 5 Tulis kenyataan TAMAT.

CONTOH (3)

Tuliskan pseudokod untuk program ringkas yang meminta nama pengguna dan memaparkan "Hello" diikuti nama pengguna.

Penyelesaian:

- ```

1. MULA
2. INPUT NAMA
3. OUTPUT "HELLO" DAN NAMA
4. TAMAT

```

## CONTOH (4)

Tuliskan pseudokod untuk program ringkas yang meminta tahun kelahiran pengguna, menghitung umur dan memaparkan umur pengguna.

**Penyelesaian:**

1. MULA
2. INPUT TAHUN\_LAHIR
3. UMUR = TAHUN\_SEMASA - TAHUN\_LAHIR
4. OUTPUT UMUR
5. TAMAT

## Carta Alir

Carta alir terdiri daripada nod dan garis penghubung. Simbol piawai mewakili setiap nod, lihat Jadual 1.1. Setiap nod dihubungkan oleh garis penghubung beranak panah. Anak panah menunjukkan aliran aktiviti.

Jadual 1.1 Simbol-simbol piawai carta alir

| Simbol | Nama nod        | Fungsi                                                                                                                                                                                                                                                                                     |
|--------|-----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|        | Terminal mula   | Permulaan algoritma dalam carta alir.                                                                                                                                                                                                                                                      |
|        | Terminal tamat  | Penamat algoritma dalam carta alir.                                                                                                                                                                                                                                                        |
|        | Input/Output    | Membaca input atau memaparkan output ke skrin.                                                                                                                                                                                                                                             |
|        | Proses          | Arahan untuk memproses input dalam bentuk ungkapan, memproses fail dan sebagainya.                                                                                                                                                                                                         |
|        | Penghubung      | Titik sambungan untuk menyambungkan carta alir yang terpisah.                                                                                                                                                                                                                              |
|        | Syarat          | Menguji syarat yang terkandung dalam nod rombus. Terdapat satu anak panah masuk dan dua anak panah keluar. Satu anak panah keluar mengalir kepada proses yang dilakukan sekiranya memenuhi syarat, manakala satu lagi anak panah keluar mengalir ke proses yang dilakukan sekiranya gagal. |
|        | Aliran aktiviti | Menghubungkan nod-nod untuk menunjukkan aliran proses.                                                                                                                                                                                                                                     |

Pembinaan carta alir tidak jauh berbeza daripada pseudokod. Melalui carta alir, IPO, simbol nod untuk semua formula, hubungan antara data dan fakta berkaitan dengan masalah dapat dikenal pasti.

## Tahukah Anda ?

Carta alir telah diperkenalkan buat kali pertamanya oleh Frank Gilbreth pada tahun 1921. Sumber: creatively.com

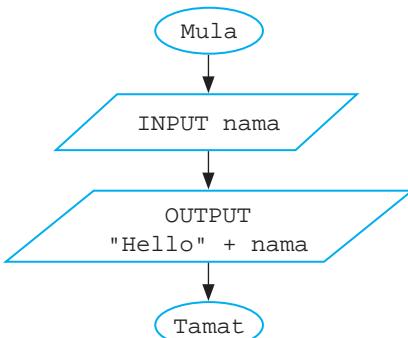
Langkah-langkah membina carta alir.

- 1 Lukis nod terminal Mula.
- 2 Lukis garis penghubung.
- 3 Lukis nod input. Masukkan butiran seperti umpanan data.
- 4 Lukis garis penghubung.
- 5 Lukis nod proses. Masukkan butiran seperti ungkapan matematik.
- 6 Lukis garis penghubung.
- 7 Sekiranya perlu, lukis nod proses atau nod input lain-lain yang diperlukan.
- 8 Sekiranya tiada, lukis nod terminal Tamat.

### CONTOH (5)

Lukis carta alir untuk program ringkas yang meminta nama pengguna dan memaparkan "Hello" diikuti nama pengguna.

**Penyelesaian:**



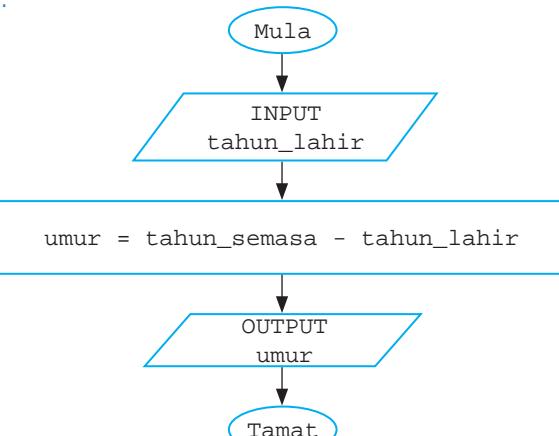
### CONTOH (6)

Lukis carta alir untuk program ringkas yang meminta tahun kelahiran pengguna, menghitung umur dan memaparkan umur pengguna.

**Penyelesaian:**

## Tahukah Anda ?

Anda harus berhati-hati ketika melukis carta alir. Apabila melukis carta alir, pastikan anak panah bersentuhan dengan nod yang diwakili oleh simbolnya. Jika tidak, carta alir dikira sebagai tidak lengkap.





Pseudocode  
[goo.gl/qzP7Gn](http://goo.gl/qzP7Gn)

Rajah 1.12 Algoritma lebih mudah difahami dalam bentuk pseudokod berbanding kod komputer



5

### Gunakan Algoritma untuk Menyelesaikan Masalah

Aktiviti Individu

Pada hujung tahun 2014, beberapa negeri di Malaysia telah mengalami bencana banjir yang besar yang merosakkan banyak tempat tinggal, tanaman, ternakan dan sebagainya. Antara negeri-negeri yang paling teruk dilanda banjir ialah negeri Kelantan, Terengganu, Pahang, Perak, Sabah dan Sarawak. Kejadian itu telah membuka mata banyak pihak untuk membuat persediaan yang rapi dan sistematik sekiranya dilanda dengan bencana yang sama sekali lagi. Sebagai rakyat Malaysia yang prihatin, cuba anda bina satu analisis IPO untuk tindakan yang perlu diambil ketika banjir besar berlaku. Kemudian, nyatakan langkah-langkah tindakan tersebut dengan menggunakan dua perwakilan algoritma, iaitu pseudokod dan carta alir. Selamat Mencuba!

## 1.2.2 Struktur Kawalan dalam Pengaturcaraan

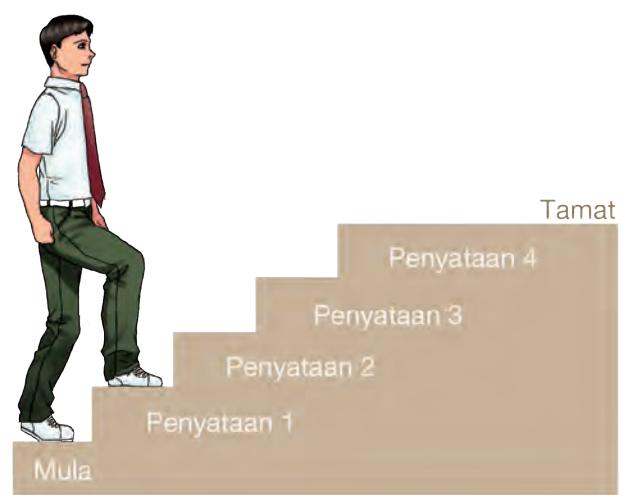
Pelbagai proses komputer yang lebih kompleks dapat dicipta dengan menggabungkan penyataan komputer. Tiga struktur kawalan komputer adalah seperti yang berikut:

- 1 Struktur kawalan urutan
- 2 Struktur kawalan pilihan
- 3 Struktur kawalan pengulangan

### Struktur Kawalan Urutan

Struktur kawalan urutan melaksanakan arahan-arahan komputer satu per satu. Urutan arahan yang betul adalah penting kerana urutan yang berlainan boleh memberikan output yang berlainan. Setiap arahan adalah satu penyataan algoritma seperti yang berikut:

- a INPUT data.
- b Memproses data.
- c OUTPUT hasil.

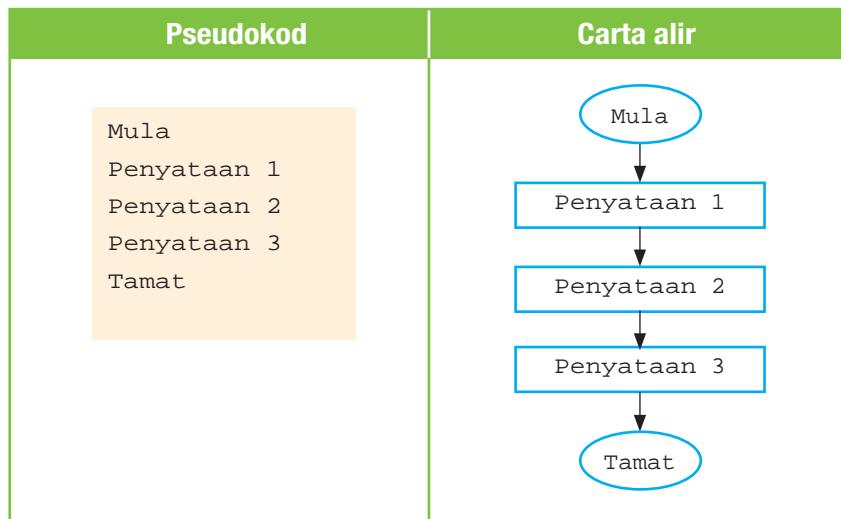




### Aplikasi Harian

Penggunaan algoritma diaplikasikan setiap hari dalam kehidupan. Setiap orang sebenarnya telah membangunkan algoritma mereka ketika melakukan sebarang pekerjaan. Misalnya, bersiap untuk ke sekolah, membeli makanan di kantin dan sebagainya.

Urutan linear algoritma dipersembahkan sebagai susunan pernyataan-pernyataan secara linear. Sintaks bagi urutan linear adalah seperti yang berikut:



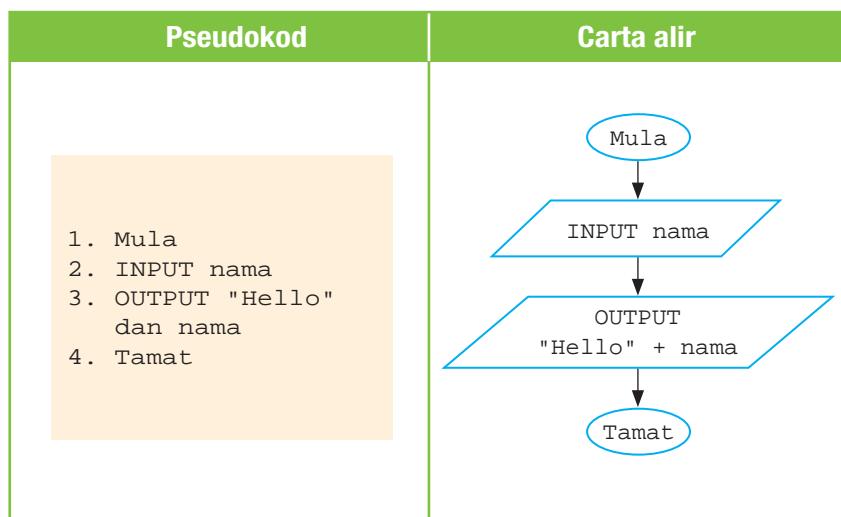
Rajah 1.13 Contoh urutan linear algoritma dalam pseudokod dan carta alir

Contoh-contoh urutan linear menggunakan pseudokod dan carta alir adalah seperti yang berikut:



### Uji Minda

Cuba anda bina urutan linear bagi menyediakan satu resepi makanan kegemaran anda. Persembakhannya dalam bentuk sintaks dan carta alir.



Rajah 1.14 Urutan linear algoritma dalam pseudokod dan carta alir untuk "Hello" dan nama

| Pseudokod                                                                                              | Carta alir                                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> 1. Mula 2. INPUT TAHUN_LAHIR 3. UMUR = TAHUN_SEMASA - TAHUN_LAHIR 4. OUTPUT UMUR 5. Tamat </pre> | <pre> graph TD     M((Mula)) --&gt; I[/INPUT<br/>tahun_lahir/]     I --&gt; P[umur = tahun_semasa -<br/>tahun_lahir]     P --&gt; O[/OUTPUT<br/>umur/]     O --&gt; T((Tamat)) </pre> |

Rajah 1.15 Pseudokod dan carta alir untuk umur

### Input Interaktif

Perhatikan pernyataan-pernyataan INPUT adalah untuk menerima data daripada pengguna. Pengguna memasukkan data melalui papan kekunci dan akan disimpan dalam pemboleh ubah. Algoritma sedemikian dikatakan interaktif kerana melibatkan input pengguna.

Pemboleh ubah ialah ruang memori yang dikhatusukan untuk menerima dan menyimpan data. Dalam subjek Matematik terutamanya dalam bab Algebra, pemboleh ubah biasanya diberikan nama abjad seperti  $x$  ataupun  $y$ .



Assignment (Umpukan)  
statement in Java  
[goo.gl/XCs16B](http://goo.gl/XCs16B)

### Umpukan

Selain input interaktif daripada pengguna, kaedah umpukan boleh digunakan untuk memasukkan data ke dalam pemboleh ubah. Akan tetapi, nilai pemboleh ubah yang diumpuk adalah sama setiap kali atur cara daripada algoritma sedemikian dijalankan.

Umpukan menggunakan simbol matematik, iaitu  $(=)$ . Merujuk Contoh 1 di bawah, di sebelah kiri operator ialah pemboleh ubah seperti  $tahun_lahir$ . Di sebelah kanan operator ialah nilai. Umpukan bermakna nilai di sebelah kanan disimpan di dalam pemboleh ubah di sebelah kiri simbol umpukan  $(=)$ .

### Contoh 1:

$tahun_lahir = 1963$  (Umpukan nilai 1963 kepada pemboleh ubah tarikh\_lahir)



Uji Minda

Berdasarkan penerangan mengenai umpukan, bina kod komputer yang ringkas untuk bulan kelahiran bagi semua murid di dalam kelas anda. Lakukan bersama-sama rakan dan bandingkan jawapan anda.





*Java Arithmetic*  
goo.gl/RVLbZb

## Contoh 2:

Umur = 21 (Umpukan nilai 21 kepada pemboleh ubah umur)

Di samping itu, operator-operator aritmetik juga digunakan dalam ungkapan yang melibatkan pengiraan nombor. Kebanyakan operator ini sama dengan operator matematik (lihat Jadual 1.2). Pengecualian ialah penggunaan simbol asterik (\*) untuk pendaraban dan simbol garis sendeng untuk pembahagian nombor. Hal ini demikian kerana, tiada simbol darab dan pembahagian untuk papan kunci piawai.

Jadual 1.2 Operator-operator aritmetik

| Operator | Makna       |
|----------|-------------|
| +        | Penambahan  |
| -        | Penolakan   |
| *        | Pendaraban  |
| /        | Pembahagian |
| ()       | Kurungan    |

## Contoh (7)

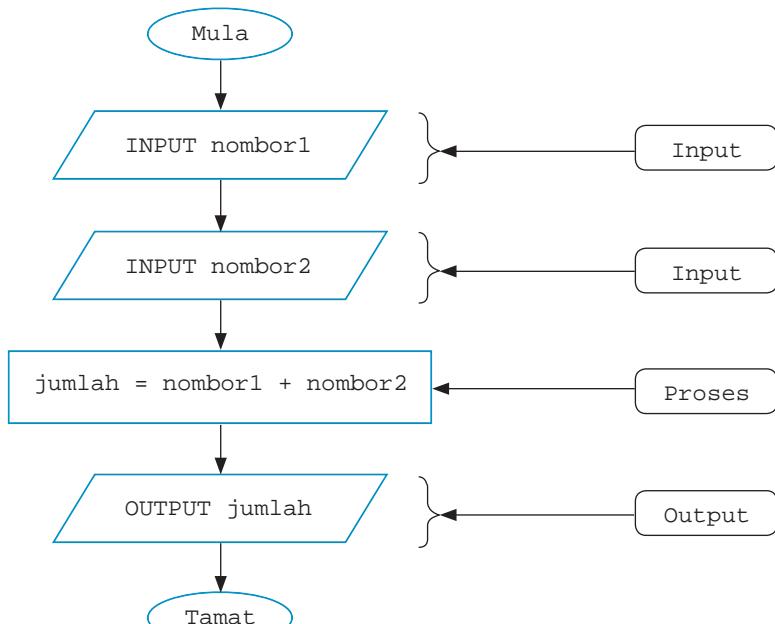
Lukiskan carta alir untuk atur cara yang menerima dua nombor dan memaparkan jumlah sebagai output.

**Penyelesaian:**



Uji Minda

Contoh 7 menunjukkan carta alir untuk atur cara yang menerima dua nombor dan memaparkan jumlah sebagai output. Bagaimanakah pula untuk melukis carta alir bagi atur cara yang menerima empat nombor dan memaparkan jumlah sebagai output?



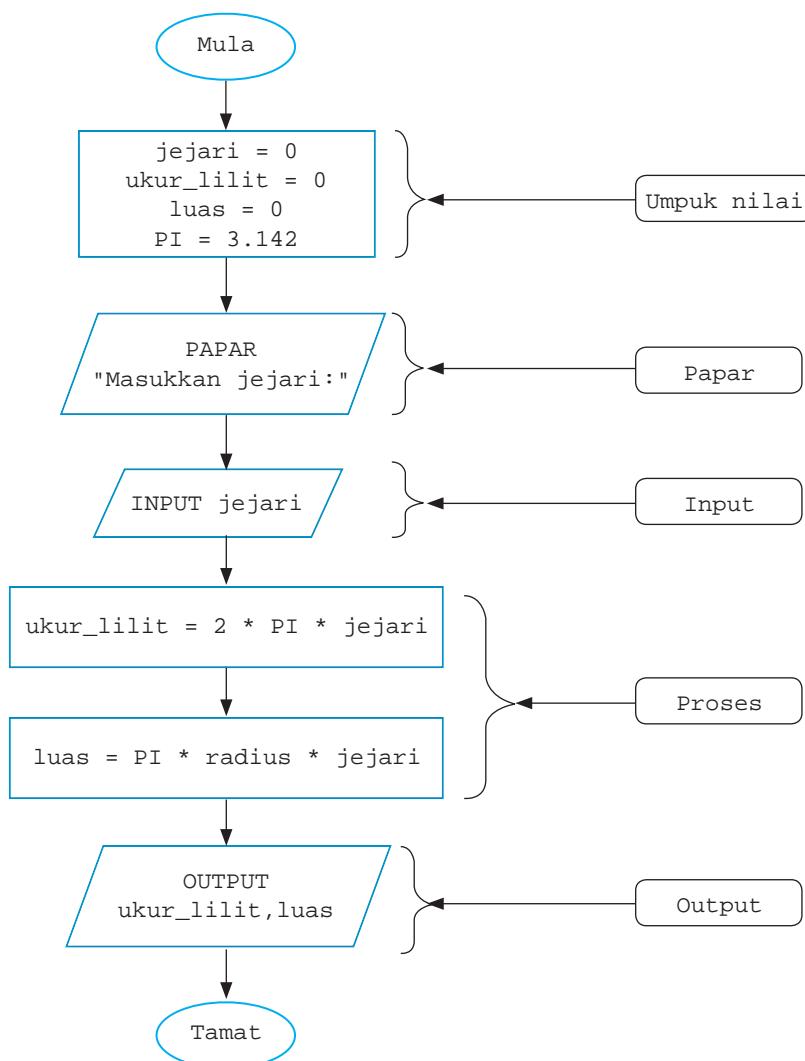
Perhatikan! Dalam nod input, perkataan-perkataan yang biasa digunakan ialah **baca**, **input** atau **dapatkan**. Dalam nod output pula, perkataan-perkataan yang biasa digunakan ialah **papar**, **cetak**, **tulis**, **output** atau **laporkan**. Perhatikan dalam nod proses, operasi aritmetik seperti tambah, tolak, darab dan bahagi digunakan dalam ungkapan. Hasil aritmetik diumpukkan kepada pemboleh ubah melalui simbol '='. Analisa penggunaan perkataan-perkataan untuk nod input, output dan proses di atas ditunjukkan dalam contoh yang berikut.



Summary of operators  
[goo.gl/5Gf2Jr](http://goo.gl/5Gf2Jr)

### CONTOH (8)

Lukis carta alir untuk atur cara yang menerima saiz jejari bulatan dan menentukan ukur lilit keluasan bulatan tersebut.



Lukis carta alir untuk atur cara yang menerima saiz jejari untuk sebuah silinder dan menentukan isi padu bagi silinder tersebut.



Berdasarkan Contoh 8, persembahkan atur cara yang diberikan dalam bentuk pseudokod.

Pernahkah anda menerima upah bekerja? Dalam aktiviti ini, anda dikehendaki mencipta sebuah perisian komputer yang dapat menghitung pendapatan kasar dan pendapatan bersih pekerja di sebuah syarikat tempatan. Perisian tersebut memerlukan input masa mula kerja, masa tamat kerja, upah sejam bekerja dan potongan PERKESO .

Formula menghitung adalah seperti yang berikut:

Pendapatan kasar = jumlah jam kerja x upah sejam

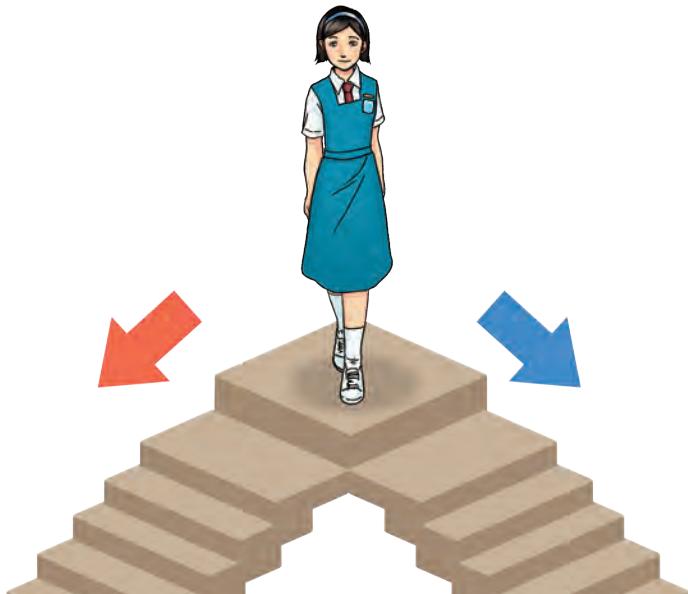
Pendapatan bersih = pendapatan kasar - potongan PERKESO

**1** Buat analisis IPO.

**2** Cadangkan penyelesaian kepada masalah di atas menggunakan pseudokod dan carta alir berdasarkan pengetahuan mengenai struktur kawalan urutan.



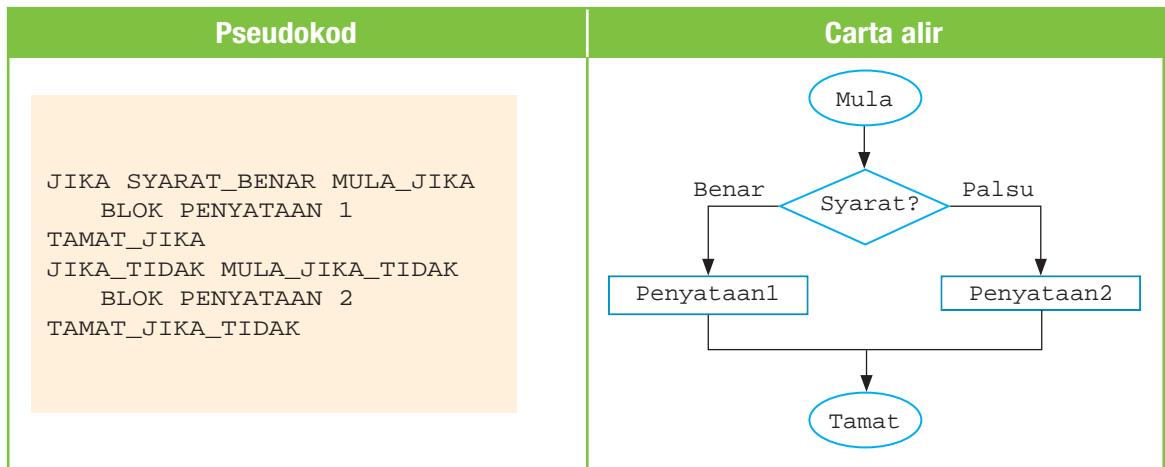
PERKESO ialah Pertubuhan Keselamatan Sosial di bawah Kementerian Sumber Manusia untuk menjaga kebajikan sosial pekerja-pekerja di Malaysia. Caruman PERKESO memerlukan pemotongan bulanan daripada gaji pekerja.



## Struktur Kawalan Pilihan

Sesetengah masalah memerlukan keputusan. Keputusan perlu dibuat setelah menilai syarat. Keputusan yang berbeza memerlukan tindakan susulan yang berbeza. Sebagai contoh, jika hari sudah gelap, ambil baju dari ampaian. Jika hari masih cerah, sidai baju di ampaian.

**Struktur kawalan pilihan** memberikan perisian komputer keupayaan untuk membuat keputusan berasaskan syarat yang telah ditentukan pengatur cara. Struktur ini membolehkan arahan-arahan lain komputer dilaksanakan dalam situasi masalah yang berbeza. Ciri umum adalah satu syarat yang menentukan satu daripada dua cabang dipilih. Setiap cabang mengandungi urutan arahan komputer yang berbeza. Sekiranya satu urutan sudah terpilih, urutan satu lagi akan diabaikan.



Rajah 1.16 Contoh pseudokod dan carta alir untuk struktur kawalan pilihan

### Menghasilkan syarat

Syarat yang digunakan dalam struktur kawalan pilihan ialah ungkapan logik. Ungkapan logik akan menilai data untuk menghasilkan keputusan "Benar" (*True*) atau "Palsu" (*False*).

#### Contoh (9)

Pengguna memasukkan input bagi harga barang.

- Cadangkan nama pemboleh ubah.
- Tuliskan ungkapan logik untuk menguji sama ada input melebihi RM100.

#### Penyelesaian:

- harga
- harga > 100



Anda membeli sepinggan nasi lemak, sekeping roti canai dan semangkuk mi kari ketika waktu rehat di kantin. Rakan anda pula membeli dua pinggan nasi lemak, dua keping roti canai dan semangkuk mi kari. Berdasarkan Contoh 9, tuliskan ungkapan logik untuk menguji sama ada input anda lebih kecil daripada rakan anda atau sebaliknya.

Untuk menghasilkan ungkapan logik, operator hubungan boleh digunakan. Dalam contoh sebelumnya, operator ' $>$ ' digunakan untuk menguji jika nilai dalam pemboleh ubah harga lebih tinggi daripada nilai yang diuji, iaitu 100. Operator hubungan membandingkan nilai dalam pemboleh ubah dengan nilai diuji. Hasil ujian ialah nilai "Benar" atau "Palsu". Jadual 1.3 menunjukkan operator hubungan yang kerap digunakan.

Jadual 1.3 Operator-operator aritmetik

| Operator | Makna                          |
|----------|--------------------------------|
| $<$      | Lebih kecil daripada           |
| $\leq$   | Lebih kecil daripada atau sama |
| $>$      | Lebih besar daripada           |
| $\geq$   | Lebih besar daripada atau sama |
| $\neq$   | Tidak sama dengan              |
| $\equiv$ | Sama dengan                    |



*Equality, Relational and conditional operators*  
[goo.gl/r2r5Re](http://goo.gl/r2r5Re)





Uji Minda

Berdasarkan Contoh 10, tuliskan ungkapan logik untuk menguji nilai dalam boleh ubah bagi situasi yang berikut:

Pengguna memasukkan ketinggian dalam aplikasi komputer. Uji sama ada ketinggian minimum ialah 150 cm.

### CONTOH 10

Pengguna memasukkan umur ke dalam aplikasi komputer. Uji sama ada umur sudah mencapai usia minimum 17 tahun.

- Kenal pasti boleh ubah.
- Kenal pasti nilai yang diuji.
- Tuliskan ungkapan logik untuk menguji nilai dalam boleh ubah.

#### Penyelesaian:

- umur
- 17
- umur  $\geq 17$  atau umur  $> 16$

### Pseudokod untuk Struktur Kawalan Pilihan

Perhatikan bahawa terdapat dua blok dalam pseudokod struktur kawalan pilihan. Blok pertama mengandungi blok pernyataan yang akan dilaksanakan sekiranya syarat memberikan nilai "Benar". Perkataan JIKA diikuti oleh syarat diuji. Blok pernyataan mesti berada di antara **MULA\_JIKA** dan **TAMAT\_JIKA**.

```
JIKA SYARAT_BENAR MULA_JIKA
BLOK PENYATAAN 1
TAMAT_JIKA
```

Blok selainnya mengandungi blok pernyataan yang akan dilaksanakan sekiranya syarat dari blok pertama gagal memberikan nilai "Benar", yakni "Palsu". Blok pernyataan "Palsu" mesti berada di antara **MULA\_JIKA\_TIDAK** dan **TAMAT\_JIKA\_TIDAK**.

```
JIKA_TIDAK MULA_JIKA_TIDAK
BLOK PENYATAAN 2
TAMAT_JIKA_TIDAK
```

### CONTOH 11

Tulis pseudokod untuk menilai hari sama ada hari sudah gelap atau sebaliknya. Pengguna dikehendaki menjawab sama ada "Benar" atau "Palsu" apabila ditanya jika hari sudah gelap. Jika "Benar" untuk hari sudah gelap, papar "Ambil baju dari ampaian." dan "Simpan dalam bakul.". Jika sebaliknya, papar "Sidai baju di ampaian".

#### Penyelesaian:

- MULA
- Papar "Hari sudah gelap: Benar atau Palsu?"
- INPUT jawapan dengan nilai "Benar" atau "Palsu"



Kerjaya Sains Komputer

Arkitek rangkaian komputer membina pelan induk rangkaian data komunikasi dan mereka bentuk *patch* untuk infrastruktur yang sedia ada sebagai tindak balas kepada ancaman keselamatan.



```

4. JIKA Hari_sudah_gelap? MULA_JIKA
 OUTPUT "Ambil semua baju dari ampaian."
 OUTPUT "Simpan semua baju dalam bakul."
 TAMAT_JIKA
JIKA_TIDAK MULA_JIKA_TIDAK
 OUTPUT "Sidai baju di ampaian."
 TAMAT_JIKA_TIDAK
5. TAMAT

```

---

### Contoh (12)

Tulis pseudokod bagi blok struktur kawalan pilihan untuk menguji sama ada umur pengguna sudah mencapai usia minimum 17 tahun. Jika umur pengguna adalah 17 tahun atau ke atas, paparkan bahawa pemohon boleh memohon menduduki ujian lesen memandu . Paparkan bahawa pemohon belum cukup umur sekiranya ujian syarat gagal.

#### Penyelesaian:

```

1. MULA
2. Papar "Masukkan umur pemohon:"
3. INPUT umur
4. JIKA umur > 16 MULA_JIKA
 OUTPUT "Anda boleh pohon lesen."
 TAMAT_JIKA
JIKA_TIDAK MULA_JIKA_TIDAK
 OUTPUT "Anda belum cukup umur."
 TAMAT_JIKA_TIDAK
5. TAMAT

```

---

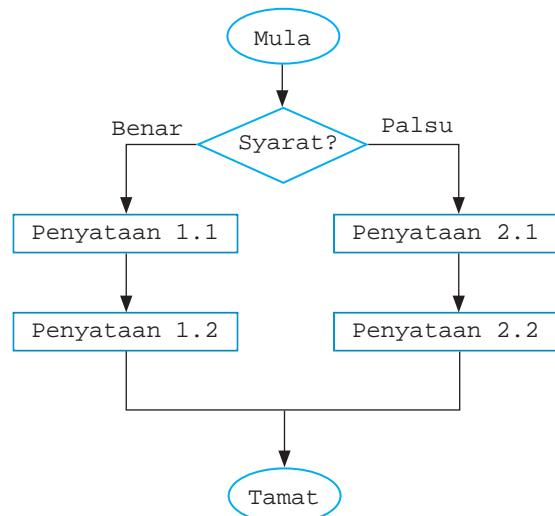


Persatuan bola sepak di SMK Gemilang sedang mencari pemain bola sepak yang baharu. Syarat yang diletakkan bagi pemain bola sepak yang baharu mestilah tidak memakai kaca mata. Tulis pseudokod untuk menguji sama ada pemain memakai kaca mata atau tidak.

### Carta alir untuk Struktur Kawalan Pilihan

Dalam carta alir, nod syarat, diwakili oleh bentuk geometri rombus. Terdapat satu anak panah masuk dan dua anak panah keluar. Anak panah masuk membawa data untuk dinilai oleh syarat dalam nod syarat. Hasil penilaian adalah sama ada "Benar" (True) atau "Palsu" (False).

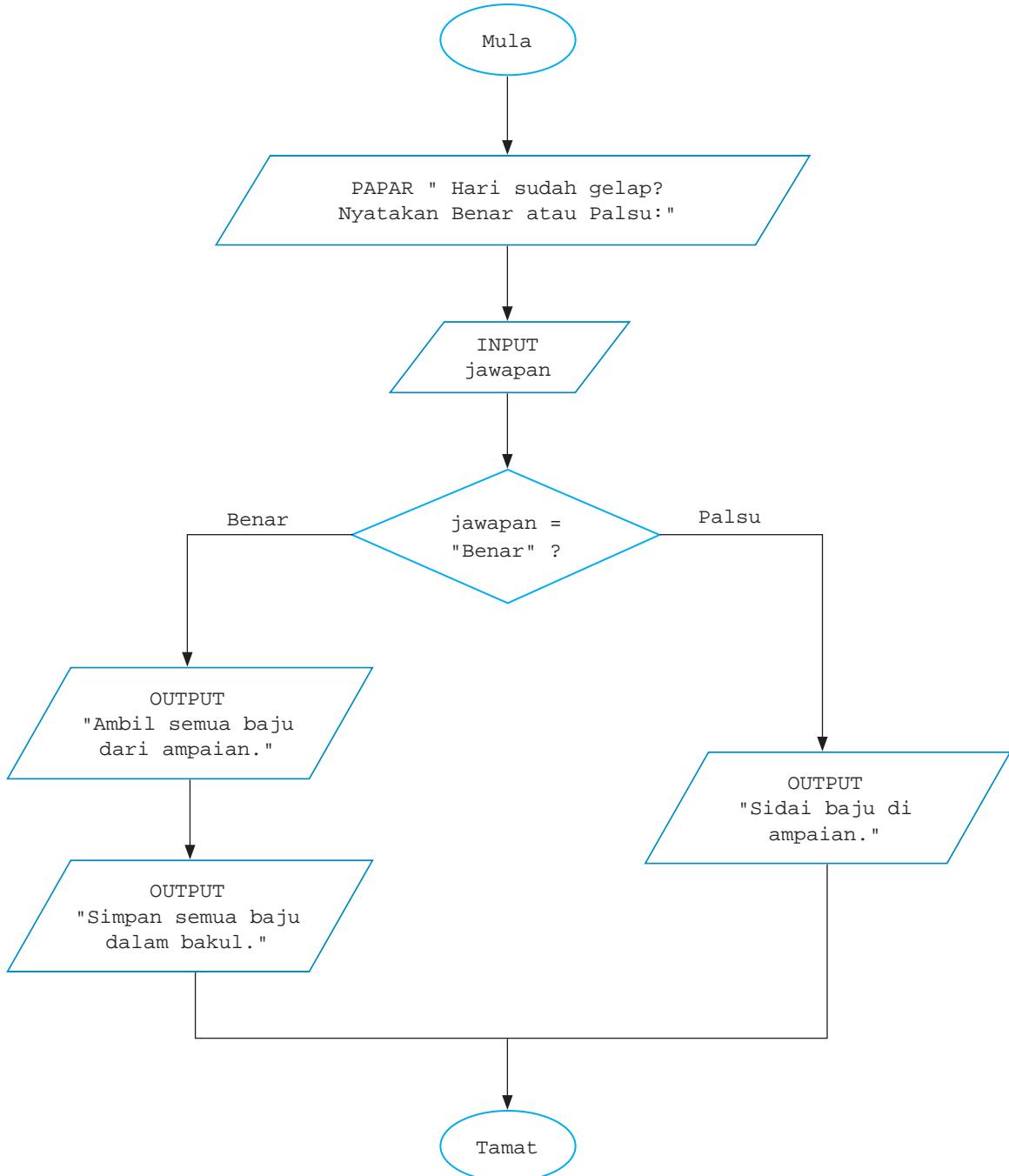
Dua anak panah keluar ialah keputusan untuk "Benar" dan "Palsu". Anak panah "Benar" dipilih apabila syarat dalam nod menghasilkan keputusan "Benar" dan "Palsu". Setiap anak panah menuju kepada blok pernyataan-pernyataan yang berbeza. Hal ini bererti tindakan susulan adalah berbeza dan bergantung kepada hasil ujian syarat.



### CONTOH 13

Lukis carta alir untuk menilai hari sama ada sudah gelap atau sebaliknya. Pengguna dikehendaki menjawab sama ada "Benar" atau "Palsu" apabila ditanya jika hari sudah gelap. Jika "Benar" untuk hari sudah gelap, papar "Ambil baju dari ampaian." dan "Simpan dalam bakul.". Jika sebaliknya, papar "Sidai baju di ampaian".

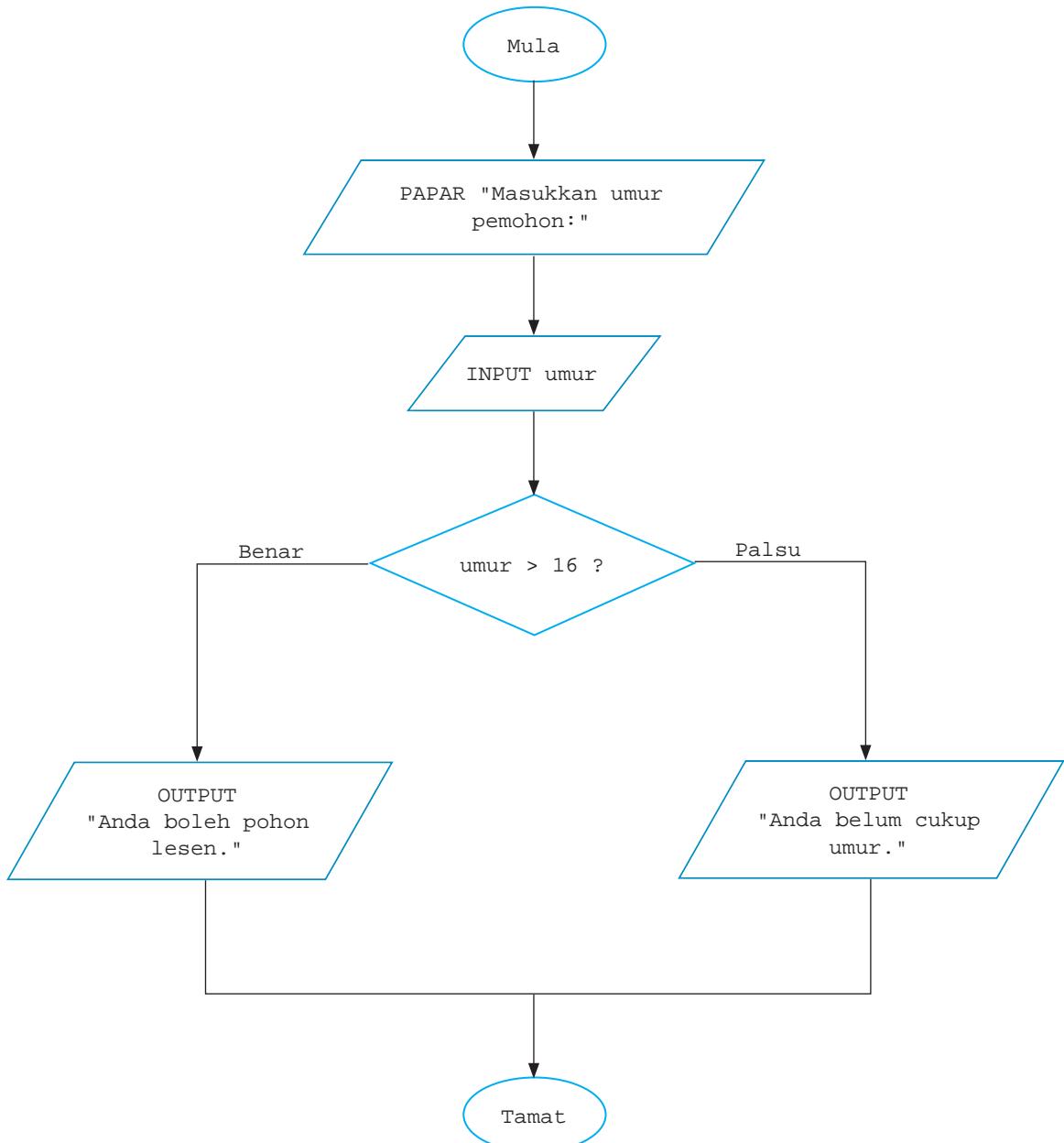
**Penyelesaian:**



### CONTOH 14

Lukis carta alir untuk atur cara yang menentukan umur pemohon dan tentukan sama ada pemohon boleh memohon lesen memandu ataupun tidak. Umur minimum ialah 17 tahun.

**Penyelesaian:**



Aktiviti ini boleh dilakukan di dalam kelas.

**Persediaan:**

- 1 Guru menyediakan 10 biji bola ping pong.
- 2 Bola-bola ping pong diberikan abjad setiap satu dari 1 hingga 10.
- 3 Guru memasukkan kesemua 10 biji bola ping pong ke dalam satu karung kain.



**Aktiviti kelas:**

- 1 Setiap murid diminta mengeluarkan sekeping kertas kosong bersaiz lebih kurang A4.
- 2 Guru akan menulis peraturan permainan berikut di atas papan tulis:

```
JIKA NOMBOR_BOLA % 2 == 0 MULA_JIKA
 OUTPUT "E"
TAMAT_JIKA
JIKA_TIDAK MULA_JIKA_TIDAK
 OUTPUT "A"
TAMAT_JIKA_TIDAK
```

**Nota:** Ujian nombor genap ialah menggunakan modulo %. Modulo % mencari baki pembahagian nombor dengan 2. Nombor genap memberikan baki sifar. Nombor ganjil memberikan baki satu.

- 3 Mulakan aktiviti. Guru akan mengambil bola daripada karung dan tunjukkan kepada kelas. Guru akan menulis jawapan dalam kertas A4 guru. Bola yang dikeluarkan tidak akan dimasukkan semula ke dalam karung.
- 4 Guru akan meminta murid menulis jawapan dalam kertas A4.
- 5 Guru akan mengulangi langkah 3 dan 4 sebanyak 6 kali.
- 6 Pada akhir aktiviti, guru menunjukkan jawapan beliau kepada murid dan meminta murid mengira bilangan jawapan yang benar.

**Contoh jawapan:**

Setiap jawapan mempunyai enam abjad kombinasi A dan E seperti berikut.

AEEEEAA

## Struktur Kawalan Pengulangan

Struktur kawalan pengulangan mengulang arahan-arahan komputer dalam blok. Ulangan boleh berlangsung sehingga menerima syarat berhenti ataupun mencapai bilangan yang sudah ditetapkan.

Cuba bayangkan seorang murid berlari mengelilingi bulatan dan terus-menerus membuat pusingan.

Sintaks struktur kawalan ulangan adalah seperti yang berikut:



| Pseudokod                                                        | Carta alir                                                                                                                                        |
|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>SELAGI SYARAT MULA_SELAGI BLOK PENYATAAN TAMAT_SELAGI</pre> | <pre>graph TD     M([Mula]) --&gt; SP{Syarat?}     SP -- Benar --&gt; BP[Blok Penyataan]     BP --&gt; SP     SP -- Palsu --&gt; T([Tamat])</pre> |

Dalam pseudokod, perhatikan bahawa syarat juga digunakan seperti dalam kawalan pilihan untuk menentukan tindakan seterusnya. Syarat diletakkan selepas kata kunci **SELAGI**. Tidak seperti kawalan pilihan, kawalan pengulangan menggunakan keputusan syarat "Benar" untuk mengulang arahan-arahan dalam blok penyataan. Jika keputusan syarat adalah "Palsu", maka blok penyataan tidak akan diendahkan. Blok penyataan ialah penyataan-penyataan di antara kata kunci **MULA\_SELAGI** dan **TAMAT\_SELAGI**.

Demikian halnya dalam carta alir, lihat rajah 1.15. Nod syarat dalam struktur kawalan pengulangan menentukan sama ada blok penyataan perlu diulang. Jika keputusan adalah "Benar", anak panah menuju



Repetition statements  
[goo.gl/9zXfhs](http://goo.gl/9zXfhs)



### Kerjaya Sains Komputer

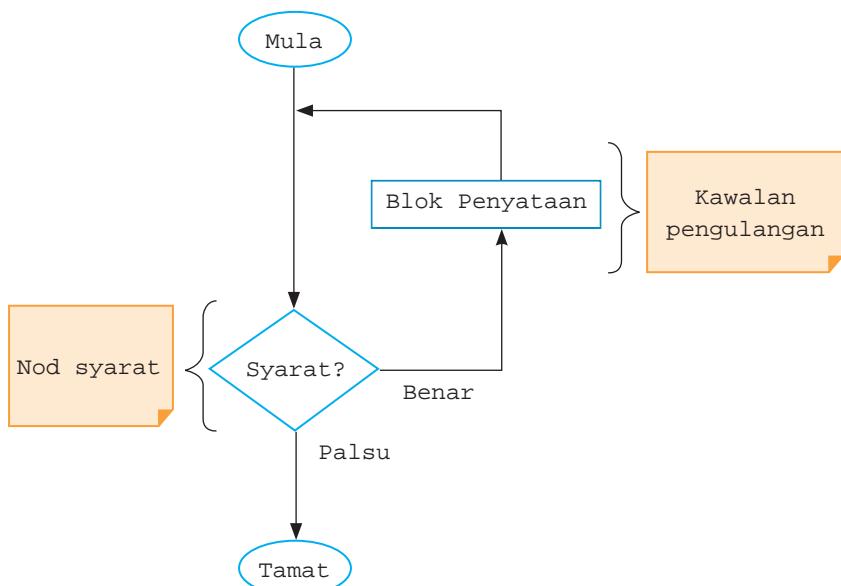
Pakar pereka data bekerja dengan set data yang sangat besar. Kerja ini juga berkaitan dengan pengiraan statistik. Pakar pereka data akan mereka bentuk algoritma dan perisian analisis data dalam persekitaran yang tertentu.



### Aplikasi Harian

Jika kita lihat kepada kehidupan seharian, didapati struktur kawalan pengulangan juga berlaku dalam kehidupan kita. Misalnya, murid yang setiap pagi akan bangun pada jam 6:00 pagi dan bersiap-siap untuk ke sekolah. Perkara ini berulang dari hari Isnin hingga ke hari Jumaat. Perkara ini hanya tidak berulang jika terdapat gangguan. Contohnya seperti murid tersebut demam pada hari Khamis. Maka, tidak akan berlaku pengulangan dalam kawalan hidupnya pada hari Jumaat minggu tersebut.

kepada blok pernyataan dan seterusnya balik kepada nod syarat untuk diuji sekali lagi. Sekiranya keputusan adalah "Palsu", anak panah keluar akan meninggalkan kesemua blok pernyataan dan nod syarat.



Rajah 1.17 Nod syarat dalam struktur kawalan pengulangan.

Dalam rajah di atas, perhatikan perbezaan reka bentuk carta alir di antara struktur kawalan ulangan dengan struktur kawalan pilihan. Perbezaan ketara adalah hala tuju anak panah bagi keputusan "Benar" yang berpatah balik supaya pengulangan boleh dilaksanakan jika perlu. Oleh sebab itulah struktur ini disebut sebagai struktur kawalan ulangan.

### CONTOH (15)

Tulis pseudokod untuk aplikasi yang akan mencetak teks "Saya gembira hadir ke sekolah hari ini." sebanyak 5 kali.

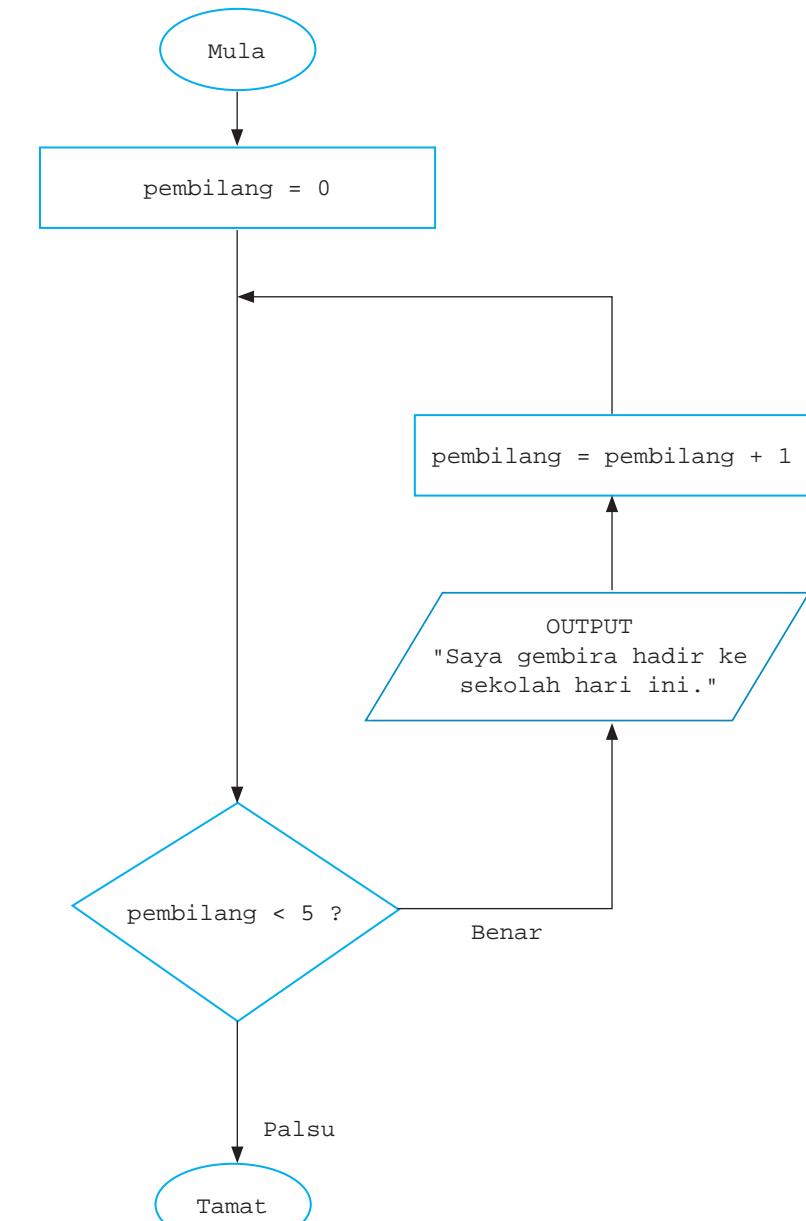
#### Penyelesaian:

1. Mula
2. pembilang = 0
3. SELAGI pembilang < 5 MULA\_SELAGI
  - OUTPUT "Saya gembira hadir ke sekolah hari ini."
  - pembilang = pembilang + 1
  - TAMAT\_SELAGI
4. Tamat

### CONTOH (16)

Lukis carta alir untuk aplikasi yang akan mencetak teks "Saya gembira hadir ke sekolah hari ini," sebanyak 5 kali.

**Penyelesaian:**



Diana, Ai Lee dan Nathan sedang berlatih di padang sekolah untuk menyertai acara larian 100 meter. Mereka telah diarahkan oleh Cikgu Hashim untuk berlari di dalam bulatan sebanyak 20 pusingan tanpa henti bagi tujuan menguatkan stamina. Berdasarkan situasi ini, bina carta alir yang sesuai.

Aktiviti ini boleh dilakukan di dalam kelas.

**Persediaan:**

- 1 Guru menyediakan 12 biji bola ping pong.
- 2 Bola-bola ping pong diberikan nombor setiap satu dari 0 hingga 9.
- 3 Dua lagi bola ping pong diberikan nombor negatif, iaitu -1 dan -2.
- 4 Guru memasukkan kesemua 12 biji bola ping pong ke dalam satu karung kain.

**Aktiviti kelas:**

- 1 Setiap murid diminta mengeluarkan sekeping kertas kosong bersaiz lebih kurang A4.
- 2 Guru akan menulis peraturan permainan berikut di atas papan tulis:

```
1. MULA
2. INPUT NOMBOR_BOLA
3. SELAGI NOMBOR_BOLA > -1 MULA_SELAGI
 3.1 OUTPUT NOMBOR_BOLA
 3.2 AMBIL BOLA DARI KARUNG
 3.3 INPUT NOMBOR_BOLA
TAMAT_SELAGI
4. TAMAT
```

- 3 Mulakan aktiviti. Guru akan mengambil bola daripada karung dan tunjukkan kepada kelas. Guru meminta murid melihat nombor atas ping pong sebagai "INPUT nombor\_bola". Guru akan menulis jawapan dalam kertas A4 guru. Bola yang dikeluarkan tidak dimasukkan semula ke dalam karung.
- 4 Guru akan meminta murid menulis jawapan dalam kertas A4 sebagai "OUTPUT nombor\_bola".
- 5 Teruskan aktiviti sehingga bola yang dikeluarkan daripada karung ialah nombor negatif.
- 6 Pada akhir aktiviti, guru menunjukkan jawapan beliau kepada murid dan meminta murid mengira bilangan jawapan yang benar.

**Contoh jawapan:**

31745968

## 1.2.3 Menguji dan Membaiki Ralat dalam Algoritma

Algoritma diuji setelah ditulis untuk tujuan pemberian. Pengujian algoritma dibuat sebelum algoritma ditulis sebagai kod komputer. Lihat Rajah 1.18. Hal ini demikian kerana, pembetulan di peringkat kod mengambil masa yang lebih lama dan lebih mahal daripada pembetulan di peringkat algoritma.



Rajah 1.18 Carta alir penulisan algoritma untuk menghasilkan arur cara.

Matlamat pengujian algoritma adalah untuk memastikan logik algoritma adalah betul dan memikirkan pemberian algoritma supaya lebih efisien. Algoritma yang telah diuji mempunyai ciri-ciri yang berikut:

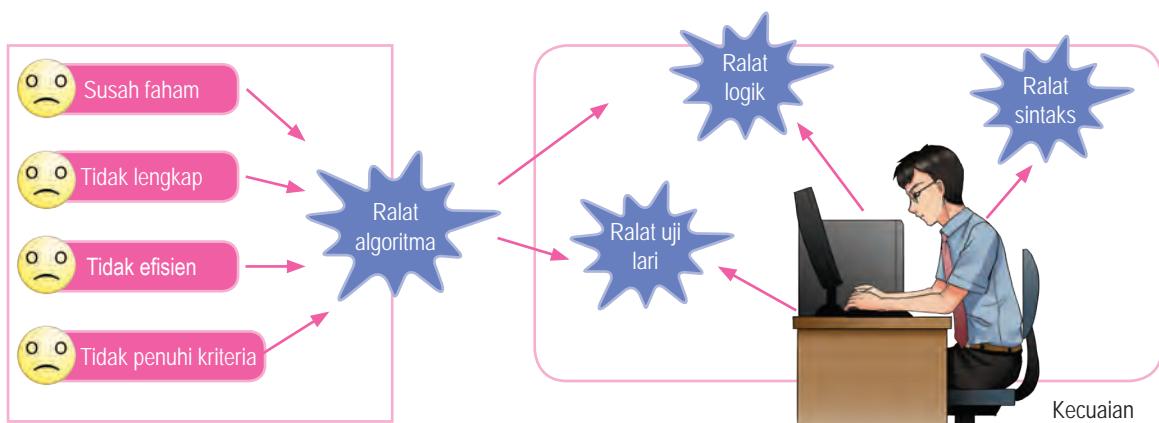
- **Mudah difahami**, maka mudah ditulis sebagai arur cara komputer.
- **Lengkap**, yakni memenuhi semua keperluan penyelesaian masalah.
- **Efisien**, yakni algoritma berfungsi dengan pantas dan tidak memakan banyak ruang memori.
- **Memenuhi kriteria reka bentuk** perisian yang ingin dibangunkan.



Penganalisis operasi penyelidikan menyelesaikan masalah dan mewujudkan strategi yang didorong oleh data dalam bidang seperti logistik dan kewangan. Penganalisis bahagian ini sentiasa berusaha untuk mencari penambahbaikan operasi data supaya tidak mengganggu program data dan sentiasa memantau proses organisasi data.

### Ralat dalam Algoritma

Algoritma yang tidak memenuhi kriteria akan sukar difahami, tidak lengkap, kurang efisien dan tidak memenuhi kriteria reka bentuk perisian. Perkara ini akan mengakibatkan perisian yang dihasilkan mengandungi ralat. Hal ini kerana pengatur cara bergantung kepada algoritma untuk menghasilkan arur cara. Algoritma yang mengandungi ralat akan menyebabkan pengatur cara menghasilkan arur cara yang juga mengandungi ralat. Rajah di bawah menunjukkan hubungan ralat algoritma dan ralat arur cara.



Rajah 1.19 Hubungan ralat algoritma dan ralat arur cara

Ujian algoritma boleh mencegah ralat algoritma. Ralat algoritma boleh menyebabkan dua jenis ralat dalam atur cara yang bakal dihasilkan daripada algoritma tersebut, lihat Jadual 1.4.

- (a) **Ralat sintaks** – tidak wujud dalam algoritma. Ralat sintaks berlaku kerana menggunakan bahasa pengaturcaraan seperti *Java* dengan cuai. Ralat sintaks biasanya ditemui secara automatik oleh perisian kompiler bahasa pengaturcaraan. Ralat algoritma tidak menyebabkan ralat sintaks.
- (b) **Ralat logik** – ralat logik berlaku kerana perisian yang dihasilkan tidak menjalankan fungsi-fungsi yang sepatutnya, tidak lengkap ataupun fungsi menghasilkan output yang tidak tepat.

Jadual 1.4 Contoh-contoh ralat logik

| Jenis ralat logik         | Contoh                                                                                                                                                                                                                               |
|---------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Fungsi yang salah         | Algoritma menggunakan formula lilitan untuk keluasan bulatan. Ralat ini berpunca daripada ungkapan yang salah.                                                                                                                       |
| Fungsi yang tidak lengkap | Algoritma sepatutnya memberikan dua output iaitu ukur lilit dan luas bulatan tetapi setelah diuji, cuma satu output yang telah dipamerkan.<br>Ralat ini berpunca daripada kecuaian sehingga mengabaikan kriteria-kriteria atur cara. |
| Output yang tidak tepat   | Algoritma tidak mempermohonkan titik perpuluhan bagi keluasan bulatan menyebabkan kehilangan kejituuan.<br>Ralat ini berpunca daripada penggunaan jenis data yang tidak sesuai ataupun umpanan yang tidak betul.                     |

Untuk mencegah ralat ini daripada berlaku dalam atur cara, algoritma mestilah dipastikan mudah difahami pengatur cara, lengkap dan memenuhi kriteria reka bentuk.

- (c) **Ralat masa larian** – ralat yang timbul apabila atur cara dijalankan. Ralat masa larian termasuklah atur cara yang tidak dapat dimulakan, menjadi sangat perlahan ataupun tidak responsif. Ralat ini boleh dikenal pasti daripada kegagalan output dan paparan amaran dalam atur cara.

Contoh:

```
Exception in thread "main" java.lang.
NullPointerException
```

Ralat masa larian boleh dikesan melalui reka bentuk algoritma yang tidak efisien atau salah. Misalnya:

- Struktur kawalan urutan yang tidak betul.
- Pemboleh ubah tidak mengandungi nilai atau objek.
- Pembahagian dengan sifar.
- Struktur kawalan pengulangan yang tidak boleh berhenti kerana logik syarat yang salah.

## Tahukah Anda?

Algoritma boleh diuji secara manual, iaitu dengan menggunakan kertas dan pen. Teknik ini dipanggil *desk-checking*. Matlamat *desk-checking* ialah pemeriksaan sama ada algoritma berfungsi dengan logik yang betul atau sebaliknya. Pemeriksaan *desk-checking* dilakukan sebelum algoritma ditulis sebagai kod komputer. Hal ini kerana, pembetulan di peringkat kod mengambil masa yang lebih lama daripada pembetulan di peringkat algoritma.



## Langkah-langkah Pengujian Algoritma

Pengujian algoritma ialah pemeriksaan secara manual dan teliti. Pengujian algoritma juga dikenal sebagai *desk-checking* kerana teknik ini menggunakan kertas dan pensel. Pengujian algoritma harus dilakukan secara sistematis. Umumnya, pengujian melibatkan langkah-langkah yang berikut:

- 1 Kenal pasti "Output Dijangka"
- 2 Kenal pasti "Output Diperoleh"
- 3 Bandingkan "Output Diperoleh" dengan "Output Dijangka".
- 4 Analisis dan baiki algoritma.



Doing a desk-check  
[goo.gl/7BImwg](http://goo.gl/7BImwg)

### Langkah 1: Kenal pasti "Output Dijangka"

Kenal pasti nilai-nilai beberapa pasangan input-output tanpa menggunakan algoritma. Nilai-nilai ini mudah diketahui penguji berdasarkan kriteria/fakta umum ataupun pengiraan manual.

#### Contoh berdasarkan kriteria:

Umur layak memohon lesen memandu ialah 17 tahun dan ke atas.

#### Analisis:

Jadi, sekiranya input umur ialah 18, output dijangka ialah permohonan dibenarkan. Begitu juga dengan umur 20, 30 dan ke atas. Namun, umur 15 tidak memenuhi syarat, oleh itu tidak dibenarkan. Analisis boleh disimpulkan dalam jadual yang ditunjukkan di bawah.

| Input uji | Output Dijangka       |
|-----------|-----------------------|
| 18        | Permohonan dibenarkan |
| 20        | Permohonan dibenarkan |
| 30        | Permohonan dibenarkan |
| 15        | Tidak dibenarkan      |

### Langkah 2: Tentukan "Output Diperoleh"

Matlamat semakan adalah untuk memastikan algoritma berfungsi dengan logik yang betul. Uji algoritma menggunakan input bagi output yang telah dikenal pasti dalam langkah 1. Output yang diperoleh melalui proses algoritma adalah "Output Diperoleh".

Salah satu kaedah sistematik adalah dengan menggunakan jadual pengujian tahap algoritma. Jadual ini memastikan setiap tahap dalam algoritma diuji. Format jadual adalah seperti ditunjukkan di bawah.

Jadual 1.5 Contoh-contoh ralat logik

| No | $P_1$ | $P_2$ | $P_n$ | I/O |
|----|-------|-------|-------|-----|
| 1  |       |       |       |     |
| 2  |       |       |       |     |
| 3  |       |       |       |     |

Penunjuk:  
No – nombor baris dalam algoritma  
 $P_1$  – boleh ubah pertama  
 $P_2$  – boleh ubah kedua  
 $P_n$  – boleh ubah ke-n  
I/O – Input/Output



Cuba buat analisis bagi situasi yang berikut:  
Ketinggian yang layak untuk menyertai pasukan bola keranjang ialah 170 cm dan ke atas. Pastikan analisis anda merangkumi maklumat input yang diuji dan output dijangka.



## Kerjaya Sains Komputer

Pakar sokongan komputer berkhidmat untuk orang ramai seperti menyediakan perkhidmatan pelanggan dan pekerja dalam bidang teknologi maklumat jabatan bagi membantu pekerja yang lain. Pakar sokongan komputer memberikan tumpuan kepada isu-isu dengan data dan rangkaian komunikasi.



## Aplikasi Harian

Konsep yang digunakan dalam *desk checking* boleh digunakan dalam kehidupan seharian kita. Misalnya, sasaran kita untuk setiap mata pelajaran ialah A dalam peperiksaan. Sasaran kita dikategorikan sebagai output dijangka. Selepas peperiksaan, keputusan yang kita peroleh ialah output yang diperoleh sama ada A, B, C dan seterusnya. Sekiranya kita tidak mencapai sasaran, kita akan membaiki kelemahan kita. Begitu juga dengan konsep yang didapati pada *desk checking*.

Untuk menghasilkan Jadual 1.5 secara efisien, pastikan perkara-perkara berikut dilaksanakan:

- 1** Pastikan pseudokod mempunyai nombor bagi setiap baris.
- 2** Kenal pasti pemboleh-pemboleh ubah yang digunakan dalam pseudokod. Hal ini adalah penting kerana pemboleh-pemboleh ubah ini akan menghasilkan lajur-lajur jadual.
  - (a) Bilangan lajur = bilangan pemboleh ubah + 2
  - (b) Lajur pertama adalah untuk nombor baris.
  - (c) Lajur-lajur seterusnya adalah untuk setiap pemboleh ubah.
  - (d) Lajur terakhir I/O adalah untuk input pengguna dan output diperolehi.
  - (e) Bilangan baris jadual = bilangan baris tahap algoritma. Rujuk pseudokod.
- 3** Dalam jadual yang ingin dilukis, masukkan 1 sebagai nombor baris pertama untuk MULA dan diakhiri dengan nombor baris terakhir pseudokod.

Contoh proses-proses dalam setiap tahap adalah seperti menerima input melalui lajur I/O, mengumpuk nilai pada lajur pemboleh ubah, mengubah nilai data dalam lajur pemboleh ubah dan mempamerkan hasil dalam lajur I/O. Output yang diperolehi melalui proses algoritma dipanggil "Output Diperoleh".

### Langkah 3: Bandingkan "Output Diperoleh" dengan "Output Dijangka"

Sekiranya "Output Diperoleh" sama dengan "Output Dijangka", algoritma mungkin bebas daripada ralat. Ulangi langkah-langkah di atas dengan beberapa "Output Dijangka" lain untuk pengesahan.

Contoh keputusan algoritma TANPA ralat:

| # | Input uji | Output Dijangka       | Output Diperoleh      |
|---|-----------|-----------------------|-----------------------|
| 1 | 18        | Permohonan dibenarkan | Permohonan dibenarkan |
| 2 | 20        | Permohonan dibenarkan | Permohonan dibenarkan |
| 3 | 30        | Permohonan dibenarkan | Permohonan dibenarkan |
| 4 | 15        | Tidak dibenarkan      | Tidak dibenarkan      |

Contoh keputusan algoritma DENGAN ralat:

| # | Input uji | Output Dijangka       | Output Diperoleh |
|---|-----------|-----------------------|------------------|
| 1 | 18        | Permohonan dibenarkan | Tidak dibenarkan |
| 2 | 20        | Permohonan dibenarkan | Tidak dibenarkan |
| 3 | 30        | Permohonan dibenarkan | Tidak dibenarkan |
| 4 | 15        | Tidak dibenarkan      | Tidak dibenarkan |



## Langkah 4: Membaiki Ralat Dalam Algoritma



- 1 Semak nama boleh ubah
- 2 Semak umpanan
- 3 Semak ungkapan
- 4 Semak urutan betul
- 5 Semak syarat struktur kawalan
- 6 Semak cabang struktur kawalan

**1 Susah faham:** Adakah baris-baris algoritma mudah diikuti semasa pengujian?

Jika tidak, semak penggunaan bahasa. Gunakan bahasa mudah, ringkas dan konsisten.

**2 Tidak lengkap:** Adakah algoritma melaksanakan semua fungsi yang diperlukan?

Sekiranya tidak, tambah baris-baris pseudokod untuk fungsi yang tidak dimasukkan.

**3 Tidak efisien:** Adakah urutan dan struktur algoritma kompleks?

- (a) Penyataan-penyataan yang tidak digunakan langsung harus dibuang.
- (b) Semak syarat dalam struktur kawalan pilihan dan struktur kawalan pengulangan supaya boleh menghasilkan kedua-dua keputusan "Benar" atau "Palsu".
- (c) Jika syarat struktur kawalan pilihan sentiasa menghasilkan "Benar" sahaja atau "Palsu" sahaja sepanjang masa, buang salah satu cabang pilihan dan ubah kepada struktur kawalan urutan.
- (d) Jika syarat dalam struktur pengulangan sentiasa "Palsu", semak sekiranya blok pengulangan benar-benar diperlukan.
- (e) Jika syarat dalam struktur pengulangan sentiasa "Benar", semak ungkapan syarat jika ada kesilapan untuk mengemaskinikan nilai boleh ubah.

**4 Tidak penuhi kriteria:** Adakah "Output Diperoleh" berlainan daripada "Output Dijangka"?

- (a) Semak ungkapan yang digunakan untuk sebarang kesilapan.
- (b) Semak urutan baris algoritma supaya logik.
- (c) Pastikan nilai-nilai diumpuk kepada boleh ubah yang sesuai.



*An algorithm for error  
correcting cyclic redundancy  
checks*  
[goo.gl/HWyu2](http://goo.gl/HWyu2)



Uji Minda

Berdasarkan Contoh 17, semak sekiranya algoritma berikut adalah bebas ralat.

```
Mula
INPUT NAMA
PAPAR "SELAMAT
TINGGAL"
Tamat
```

Algoritma ini meminta input nama daripada pengguna dan mempamerkan "Selamat tinggal" diikuti nama input sebagai output.

### Contoh 17

Semak sekiranya algoritma berikut adalah bebas ralat. Algoritma ini meminta input nama daripada pengguna dan mempamerkan "Hello" diikuti nama input sebagai output.

#### Penyelesaian:

```
Mula
INPUT NAMA
PAPAR "HELLO "
Tamat
```

Ikuti langkah-langkah berikut:

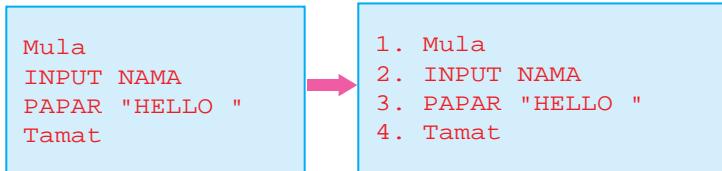
#### 1 Kenal pasti "Output Dijangka":

| Input uji | Output Dijangka |
|-----------|-----------------|
| Adam      | Hello Adam      |

#### 2 Tentukan "Output Diperoleh"

- (a) Pastikan setiap baris arahan dalam pseudokod diberikan nombor.

Contoh:



- (b) Senaraikan semua pemboleh ubah dalam algoritma:

Pemboleh ubah:nama

- (c) Lukiskan jadual pengujian tahap algoritma:

| No | Nama | I/O |
|----|------|-----|
| 1  |      |     |
| 2  |      |     |
| 3  |      |     |
| 4  |      |     |



(d) Lukiskan jadual pengujian tahap algoritma:

| No | Nama | I/O |
|----|------|-----|
| 1  | null |     |
| 2  |      |     |
| 3  |      |     |
| 4  |      |     |

Nota: "null" ialah nilai pemboleh ubah jenis data teks yang belum diumpuk dengan sebarang nilai.

(e) Laksanakan arahan dalam setiap baris. Arahan tersebut mungkin mengubah data dalam pemboleh ubah. Kemas kini nilai data setiap pemboleh ubah di setiap baris.

| No | Nama   | I/O         |
|----|--------|-------------|
| 1  | null   |             |
| 2  | "Adam" | nama?"Adam" |
| 3  | "Adam" | "Hello"     |
| 4  | —      |             |

**Output Diperoleh:**  
Hello

(f) Bandingkan "Output Diperoleh" dengan "Output Dijangka". Dalam contoh berikut, kelihatan "Output Diperoleh" berlainan daripada "Output DiJangka". Untuk mengesahkan keputusan, ulang dengan input uji yang lain.

| # | Input uji | Output Dijangka | Output DiPeroleh |
|---|-----------|-----------------|------------------|
| 1 | Adam      | Hello Adam      | Hello            |
| 2 | Zhi-Yi    | Hello Zhi-Yi    | Hello            |
| 3 | Darshini  | Hello Darshini  | Hello            |

Kesimpulannya, algoritma mengandungi ralat. Ralat ialah kegagalan input dipamerkan dalam output.



Penganalisis keselamatan maklumat memainkan peranan yang penting dalam menjaga kerahsiaan sesuatu organisasi. Mereka memeriksa sistem keselamatan teknologi maklumat dan mencadangkan langkah-langkah keselamatan, termasuk membaiki kelemahan sesuatu sistem teknologi maklumat sesebuah organisasi.



*Logic errors in Java*  
goo.gl/a1n26U

### 3 Membaiki ralat dalam algoritma.

Nombor baris yang menghasilkan output yang tidak tepat ialah baris ketiga. Baris pseudokod yang berkait adalah:

3. Papar "Hello "

Setelah diteliti, kelihatan jelas bahawa memboleh ubah nama yang mengandungi input pengguna tidak ada dalam baris tersebut. Baik baris tersebut: **3. Papar "Hello " + nama**

Jadi, pseudokod yang dibaiki kelihatan seperti yang berikut:

- ```

1. Mula
2. Input nama
3. Papar "Hello " + nama
4. Tamat

```



Uji Minda

Apakah perbezaan antara ralat algoritma dan ralat atur cara? Antara ralat yang algoritma dan ralat atur cara, yang manakah lebih mudah dikesan dan dibaiki?

Ulangi langkah-langkah pengujian menggunakan pseudokod yang telah dibaiki. Pada kali ini, hasil "Output Diperoleh" menepati "Output Dijangka". Ujian dibuat beberapa kali untuk pengesahan.

#	Input uji	Output Dijangka	Output Diperoleh
1	Adam	Hello Adam	Hello Adam
2	Zhi-Yi	Hello Zhi-Yi	Hello Zhi-Yi
3	Darshini	Hello Darshini	Hello Darshini



9

Desk checking

Aktiviti Berkumpulan

Anda dan rakan-rakan ingin melakukan perbandingan harga bagi makanan yang dijual di kantin sekolah anda dengan harga jangkaan anda. Lakukan aktiviti ini berdasarkan arahan yang berikut.

1. Bina satu jadual harga makanan yang anda dan rakan-rakan anda jangkakan. Misalnya, harga sepinggan nasi lemak, mi goreng, air sirap, dan kuih-muih. Contoh jadual adalah seperti berikut:

Nama makanan	Harga yang dijangka
Nasi lemak	RM1.50
Mi goreng	RM 1.50
Air sirap	RM 0.50
Kuih-muih (sebijil)	RM 0.40

2. Kemudian, pergi ke kantin sekolah anda dan bina jadual harga makanan yang sama yang didapati di kantin sekolah anda. Bina jadual seperti di arahan 1.
3. Setelah itu, bandingkan harga makanan yang anda jangkakan dengan harga sebenar makanan di kantin sekolah anda. Aplikasikan pembelajaran *desk checking* yang telah anda pelajari.



1.2.4 Pengesahan Nilai Pemboleh Ubah pada Setiap Tahap dalam Algoritma

Algoritma mengumpuk dan mengubah nilai sesuatu pemboleh ubah. Oleh sebab itu, nilai pemboleh ubah adalah tidak tetap. Setiap baris algoritma mungkin membuat perubahan pada pemboleh ubah tertentu. Jadual pemboleh ubah digunakan untuk mengesahkan nilai pemboleh ubah pada setiap tahap algoritma. Contohnya adalah seperti yang berikut:

Contoh (18)

Algoritma yang berikut mengumpuk pemboleh ubah z dengan nilai 100, kemudian dibahagi dengan 2, didarab dengan 4 dan ditambah sebanyak 25%. Apakah nilai z di setiap baris algoritma yang berikut?

1. Mula
2. Input $z = 100$
3. $z = z/2$
4. $z = z * 4$
5. $z = z * 0.25 + z$
6. PAPAR z
7. Tamat

Penyelesaian:

- a. Lukis jadual bernombor untuk setiap baris kod di atas.
- b. Kenal pasti dan sediakan lajur input dan output program. Lajur ini diperlukan untuk mengenal pasti input pengguna dan output yang dihasilkan program.
- c. Kenal pasti pemboleh ubah dan sediakan lajur khas pemboleh ubah z .



Uji Minda

Berdasarkan Contoh 17, berikan nilai z sekiranya pemboleh ubah z diumpuk dengan nilai 75, kemudian dibahagi dengan 5, didarab dengan 9 dan ditambah sebanyak 30%.

No	z	I/O
1	–	–
2	100	100
3	50	–
4	200	–
5	250	–
6	250	250
7	–	–



Analysis of Algorithm
goo.gl/ukdHpQ

Contoh 18 menunjukkan bahawa nilai pemboleh ubah z diubah beberapa kali. Dalam baris algoritma 2, z diumpuk dengan nilai 100, tetapi dalam baris 3 diubah menjadi 50. Dalam baris 4, nilai z didarab dengan 4 menjadi 200. Algoritma baris 5 menambah sebanyak 25% nilai z iaitu 50 kepada nilai asal z iaitu 200 menjadikan nilai baharu z ialah 250. Dalam baris 6, nilai z tidak berubah pada 250. Manakala dalam baris 7, semua nilai dibersihkan kerana algoritma sudah tamat.

Lakukan aktiviti ini mengikut arahan yang berikut.

1. Bina kumpulan dengan empat orang ahli.
2. Setiap kumpulan diberikan satu pemboleh ubah iaitu y yang diumpuk dengan nilai 60.
3. Ahli yang pertama perlu memilih suatu nombor antara nombor 1 hingga 10.
4. Ahli yang kedua pula perlu memilih suatu nombor antara nombor 11- 20.
5. Kemudian, ahli yang ketiga perlu memilih suatu operasi aritmetik sama ada “+”, “-” dan “ \times ”.
6. Ahli yang keempat juga perlu memilih suatu operasi aritmetik sama ada “+”, “-” dan “ \times ”.
7. Gabungkan semua nombor dan operasi aritmetik yang anda pilih.
8. Selepas itu, bina perwakilan algoritma berdasarkan pemboleh ubah dan semua nombor serta operasi aritmetik yang anda pilih.
9. Binakan jadual untuk mengetahui nilai y di setiap baris algoritma yang telah anda bina.



Uji Minda

Uji kemahiran anda dengan melayari laman sesawang di bawah.



goo.gl/mTFyW3

1.2.5

Mengenal Pasti Output yang Betul dari Algoritma Berdasarkan Input yang Diberi

Output yang betul bergantung kepada nilai pemboleh ubah sewaktu algoritma papar dipanggil. Dalam Contoh 18, paparan dibuat pada baris keenam. Oleh sebab itu, nilai z pada baris keenam sahaja digunakan, iaitu 250.

Adakah output yang ditentukan menggunakan algoritma betul? Membandingkan output dijangka merupakan satu-satunya cara untuk menentukan kesahihan output algoritma. Output dijangka ditentukan secara manual, iaitu hitungan manual.

Jika output algoritma adalah betul, output algoritma sepatutnya bersamaan dengan output dijangka.

CONTOH 19

Algoritma berikut menerima dua nombor dan memaparkan jumlah seperti yang berikut:

1. Mula
2. INPUT x
3. INPUT y
4. jumlah = $x + y$
5. PAPAR jumlah
6. Tamat

Tentukan sama ada algoritma di atas adalah betul.

Penyelesaian:

- 1 Tentukan pemboleh ubah: x , y , jumlah
- 2 Gunakan data ujian yang sesuai:
 - (a) Pengguna masukkan integer 12 apabila program meminta input untuk pemboleh ubah x .
 - (b) Pengguna masukkan integer 88 apabila program meminta input untuk pemboleh ubah y .
- 3 Keputusan dijangka: Hitung $x + y = 12 + 88$. Maka, output dijangka ialah 100.
- 4 Lukis jadual.
 - (a) Kesan nilai pemboleh ubah pada setiap baris algoritma.
 - (b) Kenal pasti output dari algoritma berdasarkan input.
- 5 Bandingkan output algoritma dengan output dijangka.



Berdasarkan Contoh 19, cuba gantikan nilai x dengan integer 5 dan y dengan integer 31.

Jawapan:

Bagi input $x = 12$ dan $y = 88$, output dijangka ialah 100 (jumlah).

Output algoritma, jumlahnya ialah 100.

Jumlah ini adalah sama dengan output dijangka.

Maka, algoritma telah mencapai ketepatan nilai dalam ujian di atas.

Bagi menyemak jawapan untuk kepastian, gunakan nilai input-input yang berbeza dan tentukan output dijangka.

No	Y	y	Jumlah	I/O
1	–	–	–	–
2	12	–	–	–
3	12	88	–	–
4	12	88	$12 + 88 = 100$	–
5	12	88	100	Jumlah = 100
6	–	–	–	–

Output algoritma, jumlahnya ialah 100.

Jumlah ini adalah sama dengan output dijangka.

Maka, algoritma telah mencapai ketepatan nilai dalam ujian di atas.

Bagi menyemak jawapan semula untuk kepastian, gunakan nilai input-input yang berbeza dan tentukan output dijangka.

Tahukah Anda?

Selain daripada *Java*, terdapat banyak lagi bahasa pengaturcaraan yang telah dibangunkan oleh pengatur cara. Antara bahasa pengaturcaraan yang popular ialah *Python*, *Forth*, *Perl*, *Lua*, *Smalltalk*, *Logo* dan *Ruby*.

1.2.6 Terjemahan Algoritma ke dalam Bahasa Pengaturcaraan

Setiap baris algoritma yang direka bentuk dapat ditukarkan kepada kod komputer. Algoritma berbentuk universal. Oleh itu, simbol dan perkataan yang digunakan tidak perlu bersandarkan kepada mana-mana bahasa pengaturcaraan. Banyak lagi bahasa pengaturcaraan yang sering digunakan termasuklah *Visual Basic (VB)*, *Java*, *C#* dan lain-lain.

CONTOH (20)

Terjemahkan setiap baris algoritma kepada kod komputer yang sempurna.

Penyelesaian:

Algoritma	Kod komputer (Java)
Mula	public static void main(String[] args){
PAPAR "Nama pengguna:"	System.out.print("Masukkan nama pengguna: ");
INPUT nama	String nama = new java.util.Scanner(System.in). nextLine();
PAPAR "Apa khabar", nama, "?"	System.out.println("Hello " + nama);
Tamat	}



11

Menterjemah algoritma ke dalam bahasa pengaturcaraan

Aktiviti Berpasangan

Anda dan rakan anda diberikan dua situasi. Terjemahkan situasi yang diberikan kepada bentuk algoritma dan kod komputer *Java*. Selamat mencuba!

Situasi 1

Ramesh mengajak ibu dan bapanya untuk bercuti di Perak ketika cuti sekolah. Dia membuat perancangan untuk melawat ke Zoo Taiping ketika percutian tersebut. Ibu bapanya telah bersetuju dengan rancangan Ramesh. Untuk mengelakkan daripada kesesakan ketika beratur bagi membeli tiket memasuki zoo tersebut, Ramesh membuat keputusan untuk membeli tiket secara atas talian. Bantu Ramesh membeli tiket secara atas talian dengan menggunakan algoritma dan terjemahkannya kepada kod komputer.

Situasi 2

Chew An dan Jamilah ingin meminjam buku rujukan Sains Komputer Tingkatan 4 di sekolahnya. Bantu Chew An dan Jamilah untuk meminjam buku tersebut dengan menggunakan algoritma dan terjemahkannya kepada kod komputer.

Latihan Formatif 1.2

- 1 Tulis algoritma yang dapat menentukan keluasan dinding sebuah bilik setelah mengambil kira dimensi tingkap.
- 2 Berdasarkan soalan 1, ubah algoritma untuk menentukan kos mengecat sebuah bilik setelah kos satu kaki per segi dimasukkan.
- 3 Tulis pseudokod untuk atur cara yang telah menerima dua nombor dan memaparkan jumlahnya sebagai output.
- 4 Tulis pseudokod untuk atur cara yang menerima jejari bulatan dan menentukan ukur lilit dan luas bulatan tersebut.
- 5 Lukis carta alir untuk atur cara yang menerima dua nombor dan kemudian mempamerkan purata.
- 6 Tentukan sama ada pernyataan berikut "Benar" atau "Palsu":
 - (a) Pengujian algoritma dibuat setelah kod Java ditulis. ()
 - (b) Ralat logik berpunca daripada ralat algoritma ()
 - (c) Ralat sintaks berpunca daripada ralat algoritma. ()
 - (d) Salah satu punca ralat algoritma ialah kegagalan memenuhi kriteria. ()
- 7 Katakan anda menetap di Kemaman dan perlu berulang-alik ke Kuantan. Anda ingin mengetahui sama ada memandu kereta ataupun menaiki bas lebih menjimatkan. Anda juga mengetahui harga tiket bas dari Kemaman ke Kuantan dan jarak perjalanan dan kos petrol bagi satu kilometer. Tuliskan algoritma untuk menyelesaikan masalah ini.



- 8 Sebuah aplikasi menerima dan menjumlahkan lima nombor integer daripada pengguna menggunakan struktur kawalan pengulangan. Seterusnya, aplikasi menghasilkan output purata lima nombor tersebut.
 - (a) Bina analisis IPO.
 - (b) Hasilkann pseudokod yang bersesuaian.
 - (c) Lukis carta alir untuk algoritma penyelesaian.
- 9 Hasilkann algoritma untuk atur cara menghitung lilitan bulatan dengan jejari dalam meter diberikan. Diketahui formula menghitung lilitan ialah $2 \times \pi \times \text{jejari}$. ($\pi = 3.142$)
 - (a) Buat analisis untuk menentukan Output Dijangka.
 - (b) Tentukan pseudokod untuk Output Diperoleh.
 - (c) Sekiranya terdapat ralat, cadangkan pembaikan bagi ralat tersebut.





STANDARD PEMBELAJARAN

1.3.1 Menjelaskan hubungan antara jenis data, saiz data dan kapasiti ingatan komputer

1.3.2 Memilih dan menggunakan jenis data yang bersesuaian: Integer (*int*), nombor nyata (*float, double*), aksara (*char*), rentetan (*string*) dan Boolean.

1.3.3 Membezakan pemboleh ubah sejagat (*global*) dan setempat (*local*)

1.3.4 Mengisytiharkan, memulakan dan menetapkan nilai pada pemboleh ubah dan pemalar

1.3.5 Menggunakan pernyataan umpukan dan pernyataan aritmetik

1.3.6 Menulis atur cara untuk memasukkan input dari papan kekunci dan memaparkan output

Pemboleh Ubah, Pemalar dan Jenis Data

Pemboleh ubah atau pemalar dalam Java, perlu diisyiharkan sebelum digunakan. Konsep dan prosedurnya adalah sama seperti ingin melancong ke luar negara, iaitu seseorang individu perlu mengisytiharkan barang yang ada di dalam bagasi di pejabat Kastam.

Dalam proses pengaturcaraan, seorang pengatur cara perlu mengisytiharkan jenis data yang diperlukan dalam sesuatu program yang hendak dilaksanakan.

Pemboleh ubah

Pemboleh ubah ialah ruang simpanan sementara untuk nombor, teks dan objek. Nilai pemboleh ubah sentiasa berubah semasa berlakunya pemprosesan dan tidak akan memegang sebarang nilai selepas program tamat.

Pengatur cara perlu memberikan nama kepada setiap pemboleh ubah yang digunakan untuk menyelesaikan sesuatu masalah dalam program yang dibangunkan. Pengatur cara menggunakan nama pemboleh ubah sebagai nama rujukan untuk nilai spesifik pemboleh ubah tersebut. Komputer pula akan menggunakan nama pemboleh ubah tersebut sebagai rujukannya untuk mencari nilai pemboleh ubah itu dalam memorinya. Pemboleh ubah merupakan elemen dalam pengaturcaraan dan nilainya boleh berubah pada bila-bila masa sewaktu proses pengaturcaraan dilaksanakan. Contohnya,

```
public class kiraLuas {  
    public static void main(String[] args) {  
        float panjang, lebar, luas;  
    }  
}
```

Contoh yang lain adalah dalam situasi kos sepasang kasut. Item input ini perlu diberikan satu nama pemboleh ubah kerana kos sepasang kasut mungkin berubah mengikut jenis kasut. Nama pemboleh ubah sepatutnya konsisten dengan nilai pemboleh ubah. Dalam kes ini, nama pemboleh ubah ialah kosKasut kerana nama ini merujuk kos sepasang kasut. Sekiranya kos sepasang kasut berubah, nilai pemboleh ubah akan berubah tetapi nama pemboleh ubah tidak akan berubah.



Pemalar

Sewaktu proses pengaturcaraan dilaksanakan, nilai pemalar adalah tetap dan tidak akan berubah. Pemalar biasanya digunakan semasa pengatur cara ingin mengisyiharkan nilai yang tidak berubah. Contohnya, nilai Pi (π), bilangan hari dalam seminggu, bilangan jam dalam sehari, kadar pertukaran nilai mata wang Malaysia dan sebagainya. Jika ingin menggunakan nilai seperti ini, pengatur cara perlu mengisyiharkannya sebagai pemalar. Contohnya,

```
//Pengisytiharan pemalar
final double pi = 3.142;

//Pengisytiharan pemalar
final int BILANGAN_JAM_DALAM_SEHARI = 24;

//Pengisytiharan boleh ubah
final int BILANGAN_HARI_DALAM_SEMINGGU = 7;

//Pengisytiharan pemalar
final double Dolar_Singapura_kepada_Ringgit_Malaysia = 3.03;
```

Jenis data yang diisyiharkan mestilah sepadan dengan nilai. Contohnya, pemalar jenis integer mestilah memberikan nilai integer.

Jenis Data

Jenis data dalam bahasa pengaturcaraan merujuk satu set data yang mempunyai nilai dan ciri-ciri yang telah ditetapkan. Jenis data dan contoh penggunaan nilai yang dibenarkan adalah seperti dalam Jadual 1.6.

Jadual 1.6 Jenis data dan contoh penggunaan nilai yang dibenarkan

Jenis data	Contoh nilai	Kapasiti Ingatan Komputer
Integer	Minimum -2147483648 Maksimum 2147483647	4 bait
float	Minimum -3.4e38 Maksimum 3.4e38	4 bait
double	Minimum -1.7e308 Maksimum 3.4e38	8 bait
char	Satu karakter sahaja	2 bait
String	Bermula dari 0 hingga tiada had	>10 bait
Boolean	Benar (true) Palsu (false)	1 bit

Data merupakan satu elemen yang memainkan peranan yang penting dalam sesebuah pengaturcaraan. Data yang dimasukkan dalam pengaturcaraan akan diproses untuk menghasilkan output bagi sesuatu permasalahan yang wujud. Jenis data boleh dikategorikan kepada dua kelas, iaitu data primitif dan data bukan primitif.



Jenis, nilai dan pemboleh ubah
<https://goo.gl/tXAp55>



Tahukah Anda?

Cuba anda lihat contoh pengisytiharan pemalar yang ditunjukkan di bawah.

```
//Pengisytiharan pemalar
final double pi = 3.142;
```

Tulisan yang berwarna hijau dikenal sebagai komen. Komen ialah penanda yang dibuat oleh pengatur cara untuk setiap atur cara yang dibina. Dalam Java, terdapat sintaks yang telah ditetapkan untuk komen. Antara tiga gaya tatatanda komen yang boleh digunakan ialah,

- (i) //Pengisytiharan boleh ubah
- (ii) /* Pengisytiharan boleh ubah */
- (iii) /** Pengisytiharan boleh ubah */

Anda akan menjumpai lebih banyak tatatanda komen selepas ini.

Untuk penerangan yang lebih lanjut mengenai komen, boleh rujuk di subtopik 1.5.

Tahukah Anda ?

Data terdiri daripada dua jenis iaitu data primitif dan data bukan primitif.

Data primitif

Data primitif ialah data yang nilainya dalam bentuk paling asas dan tidak boleh diper mudahkan lagi. Data primitif juga boleh dikenal sebagai data tidak berstruktur.

Data bukan primitif

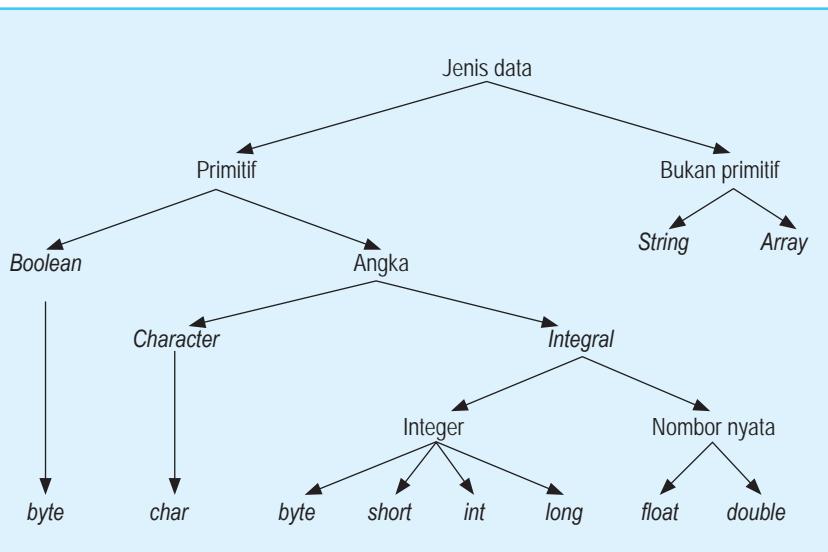
Data bukan primitif ialah data yang nilainya terdiri daripada beberapa gabungan nilai asas. Data bukan primitif juga boleh dikenal sebagai data berstruktur.



Sukarelawan ICT Malaysia atau '*Malaysia ICT Volunteer*' (MIV) ialah program sukarelawan ICT yang berhasrat untuk memberikan khidmat sukarelawan bagi membantu komuniti di kawasan yang telah dikenal pasti dari segi penggunaan peralatan ICT dan pembangunan ICT dalam masyarakat.



<http://goo.gl/qba9Qr>



Rajah 1.20 Jenis data dalam bahasa pengaturcaraan



12

Kategori jenis data

Kerja Berkumpulan

Seorang pengatur cara ingin membangunkan sistem maklumat murid.

Berdasarkan pernyataan di atas, bincang bersama-sama ahli kumpulan tentang,

- 1 (a) jenis data yang diperlukan
(b) kategori jenis data
- 2 Berikan tiga contoh situasi yang lain. Berdasarkan contoh yang telah diberikan, ulang langkah di 1(a) dan (b).
- 3 Bentangkan hasil perbincangan kumpulan anda.

1.3.1 Hubungan antara Jenis Data, Saiz Data dan Kapasiti Ingatan Komputer

Setiap jenis data dan saiz data akan disimpan dalam ingatan komputer. Nama pemboleh ubah memainkan peranan yang penting dalam menentukan saiz data dalam ingatan. Berdasarkan jenis data yang digunakan pada pemboleh ubah, sistem operasi akan menentukan apa-apa yang boleh disimpan dalam ingatan komputer. Kepelbagaiannya penggunaan jenis data pada pemboleh ubah dapat menjimatkan ruang pada ingatan komputer. Oleh itu, penggunaan jenis data yang sesuai

diperlukan untuk nilai yang panjang supaya menjimatkan ruang ingatan komputer. Jadual 1.6 pada muka surat 57 menunjukkan jumlah memori yang digunakan untuk menyimpan jenis data bagi boleh ubah dalam ingatan komputer. Simpanan memori ini disimpan dalam unit bait (*bytes*).



13

Jenis data dan kapasiti ingatan komputer

Kerja Berkumpulan

Anda dikehendaki berada dalam kumpulan kecil dan membuka akaun Sistem Analisis Peperiksaan (SAPS). Berdasarkan slip peperiksaan yang dipaparkan, senaraikan berapakah jenis data dan kapasiti ingatan komputer yang digunakan bagi setiap data yang telah disenaraikan.

Jenis Data	Kapasiti Ingatan Komputer



Jom belajar kod *Java* ringkas!
<http://goo.gl/c2AD4e>



Aplikasi Harian

Pengetahuan sumber terbuka memudahkan anda mendapatkan peluang pekerjaan dalam bidang ICT



<http://goo.gl/9JuhTe>



Inovasi Sains Komputer

Dagang Net ialah sebuah syarikat GLC yang menyediakan perkhidmatan untuk menjalankan perniagaan melalui *e-commerce* iaitu alat strategi untuk perniagaan. Dagang Net merupakan pembekal perkhidmatan *e-commerce* yang pertama dan terbesar di Malaysia.



<http://goo.gl/nDmQr9>

1.3.2 Pemilihan dan Penggunaan Jenis Data yang Bersesuaian

Pemilihan dan penggunaan data yang sesuai amat penting dalam sesebuah atur cara supaya atur cara dapat dibangunkan tanpa ralat sintaks. Jenis data bagi boleh ubah dan pemalar yang menentukan jenis maklumat akan disimpan dalam ruang ingatan yang diperlukan.

Penggunaan jenis data yang sesuai juga berfungsi sebagai penanda aras kepada sesuatu boleh ubah itu sama ada boleh ubah tersebut menyimpan data yang tetap, data yang boleh dikira, huruf, nombor perpuluhan atau data yang mempunyai pilihan benar atau palsu.



BAB 1 PENGATURCARAAN



Integer

int

Nombor nyata

*float,
double***Apakah integer?**

Integer merangkumi semua nombor yang tidak mempunyai pecahan atau perpuluhan. Integer ialah nombor asal yang lebih besar daripada bait. Integer boleh bernilai antara -2147483648 hingga 2147483648.

Apakah yang perlu dilakukan untuk mengisyiharkan pemboleh ubah?
Pengatur cara perlu menggunakan kata kunci *integer*.

CONTOH

```
int nombor_pertama, nombor_kedua, jawapan;
nombor_pertama = 10;
nombor_kedua = 20;
jawapan = nombor_pertama + nombor_kedua;
```

Apakah nombor nyata?

Nombor nyata ialah nombor yang terdiri daripada semua nombor yang mempunyai titik perpuluhan atau bahagian pecahan. Nombor nyata bernilai antara:

1.79769313486231e308 hingga 4.94065645841247e-324, jika nombor adalah negatif, dan 1.79769313486231E308 hingga 4.94065645841247E-324, jika nombor adalah positif.

Apakah yang perlu dilakukan untuk mengisyiharkan pemboleh ubah yang boleh menyimpan nombor perpuluhan yang besar?

Pengatur cara perlu menggunakan kata kunci *Double*.

CONTOH

```
float tinggil, tapak1; //Pengisyiharan pemboleh ubah
double luas; //Pengisyiharan pemboleh ubah
"" digunakan apabila ingin menerangkan kod arahan yang berada dalam satu baris.
```

Aksara/rentetan

char/
String

Boolean

Apakah aksara/rentetan?

Aksara dan rentetan merupakan data dalam bentuk ruang kosong (*space*), teks, perkataan atau nilai yang mengandungi susunan aksara atau simbol.

Apakah yang perlu dilakukan untuk mengisytiharkan pemboleh ubah rentetan (**string**)?

Pengatur cara perlu menggunakan data jenis *String*.

CONTOH

```
char ampersand = '@';  
String email_address = "meme@me.com";
```

Apakah Boolean?

Boolean ialah data dalam bentuk pilihan, iaitu memilih salah satu daripada sesuatu yang "benar" (*True*) atau "palsu" (*False*).

Apakah yang perlu dilakukan untuk mengisytiharkan pemboleh ubah?

Pengatur cara perlu menggunakan kata kunci "benar" atau "palsu".

CONTOH

```
boolean forLeft = false;  
boolean forStraight = true;  
boolean forRight = false;
```





Kerjaya Sains Komputer



Pengatur cara terkenal:
Steve Wozniak
<http://goo.gl/rdGKvh>



14

Jenis data

Kerja Berkumpulan

Bahagikan kelas kepada empat kumpulan. Setiap kumpulan perlu menyenaraikan contoh data sebanyak mungkin berdasarkan enam jenis data yang diberikan.

integer	String	float	double	char	boolean



Inovasi Sains Komputer



Supercomputer
<https://goo.gl/8zj0zW>

1.3.3 Perbezaan Pemboleh Ubah Sejagat dan Setempat

Setiap pemboleh ubah dan pemalar mungkin wujud dan akan digunakan untuk keseluruhan atur cara atau hanya bagi satu fungsi. Kewujudan pemboleh ubah atau pemalar dikenal sebagai kawasan yang kedua-duanya boleh digunakan secara sah. Pemboleh ubah dan pemalar bagi pengatur cara terdiri daripada dua skop iaitu:

- a** Pemboleh ubah menjangkau seluruh program (*Global*)
 - Pemboleh ubah menjangkau hanya berfungsi dalam atur cara sahaja.
 - Penggunaan pemboleh ubah menjangkau adalah semasa dalam fungsi atur cara sahaja. Apabila tatacara itu telah tamat, ruang memori yang diperuntukkan kepada pemboleh ubah dan pemalar juga akan tamat.

- b** Pemboleh ubah setempat (*Local*)
 - Pemboleh ubah setempat hanya berfungsi dalam subatur cara yang diisytiharkan.
 - Digunakan dalam fungsi 'di mana' pemboleh ubah diisytiharkan, bermula 'dari mana' pemboleh ubah diisytiharkan dan bila penamat akhir tatacara tersebut.



NetBeans untuk Java—
Cara untuk memasang
dan menggunakan
pengaturcaraan Java pada
Windows, Mac dan Linux
<https://goo.gl/bM8nG6>



Lihat ciri-ciri pemboleh ubah sejagat dan pemboleh ubah setempat dalam Jadual 1.7.

Jadual 1.7 Perbezaan ciri-ciri antara pemboleh ubah sejagat dengan pemboleh ubah setempat

Ciri-ciri	Pemboleh ubah sejagat (<i>Global variable</i>)	Pemboleh ubah setempat (<i>Local variable</i>)
Pengisytiharan	Pengisytiharan yang dilakukan di luar mana-mana fungsi.	Pemboleh ubah yang diisytiharkan dalam sebuah fungsi dalam atur cara.
Akses	Boleh diakses di mana-mana fungsi.	Tidak boleh diakses di luar fungsi itu.
Fungsi	Boleh digunakan hingga ke akhir program. Jika sesuatu pemboleh ubah setempat mempunyai nama yang sama dengan pemboleh ubah sejagat, rujukan hanya dibuat terhadap pemboleh ubah yang terdekat, iaitu pemboleh ubah setempat.	Hanya boleh digunakan untuk fungsi yang diisi. Jika dua fungsi menggunakan pemboleh ubah tempatan yang sama, tiada masalah akan berlaku.

Tahukah Anda?

Pemboleh ubah sejagat juga dikenal sebagai pemboleh ubah statik di dalam Java.



Apakah pemboleh ubah statik?
<https://goo.gl/gIloIb>

```
public class A{  
    int data = 50;  
    static int m = 100; //global variable  
  
    public static void main(String[] args) {  
        int n=90; //local variable  
    }  
}//end of class
```

Pemboleh ubah sejagat

Pemboleh ubah setempat

Rajah 1.21 Contoh atur cara yang menggunakan pemboleh ubah sejagat dan pemboleh ubah setempat



15

Pemboleh ubah sejagat dan pemboleh ubah setempat

Kerja Berkumpulan

Berdasarkan atur cara di bawah, labelkan pemboleh ubah sejagat dan pemboleh ubah setempat.

```
public class info {  
    static int l = 5;  
    public static void main(string[] arg) {  
        int m = 7;  
    }  
}
```

1.3.4 Pengisytiharan, Permulaan dan Penetapan Nilai pada Pemboleh Ubah dan Pemalar

Tahukah Anda?

Pencam ialah nama yang digunakan untuk mewakili nama objek, pemboleh ubah, pemalar, prosedur dan fungsi.

Sebelum membangunkan sesuatu atur cara, pemboleh ubah dan pemalar perlu melalui proses pengisytiharan, permulaan dan penetapan nilai.

Pengisytiharan Nilai Pemboleh Ubah

Pemboleh ubah dikenal sebagai tempat untuk menyimpan data. Setiap pemboleh ubah di dalam Java mempunyai jenis yang tertentu, yang menentukan saiz dan susun atur memori dan set operasi yang boleh digunakan untuk pemboleh ubah. Anda mestilah mengisytiharkan semua pemboleh ubah sebelum boleh digunakan.

Pengisytiharan pemboleh ubah perlu dilaksanakan dengan memberikan jenis data dan nama pemboleh ubah. Contoh pengisytiharan pemboleh ubah adalah seperti yang berikut:

```
public class pembolehUbah {  
    public static void main(String[] args) {  
        int x; ←  
    }  
}
```

Pengisytiharan Integer

- Int ialah jenis data
- x ialah nama pemboleh ubah

```
public class pembolehUbah {  
    public static void main(String[] args) {  
        String nama; ←  
    }  
}
```

Pengisytiharan String

- String ialah jenis data
- nama ialah nama pemboleh ubah

```
public class pembolehUbah {  
    public static void main(String[] args) {  
        double s, t, u; ←  
    }  
}
```

Pengisytiharan Double

- Double ialah jenis data
- s, t,u ialah nama pemboleh ubah

```
public class pembolehUbah {  
    public static void main(String[] args) {  
        boolean v; ←  
    }  
}
```

Pengisytiharan Boolean

- Boolean ialah jenis data
- v ialah nama pemboleh ubah



```
public class pembolehUbah {
    public static void main(String[] args) {
        char w; ←
    }
}
```

Pengisytiharan Char

- Char ialah jenis data
- w ialah nama pemboleh ubah

```
public class pembolehUbah {
    public static void main(String[] args) {
        float y; ←
    }
}
```

Pengisytiharan Float

- Float ialah jenis data
- y ialah nama pemboleh ubah

Permulaan dan Penetapan Nilai untuk Pemboleh Ubah

Selepas mengisytiharkan jenis data dan nama pemboleh ubah, nilai kepada pemboleh ubah boleh ditetapkan atau diumpukan. Sintaks untuk menetapkan nilai pemboleh ubah adalah seperti yang berikut.

Jenis Data NamaPembolehUbah = nilai;

Integer

```
public class pembolehUbah {
    public static void main(String[] args) {
        int x;
        x = 10;
    }
}
```

Penetapan nilai

Pengisytiharan &
Penetapan nilai

```
public class pembolehUbah {
    public static void main(String[] args) {
        → int x = 10;
    }
}
```

String

```
public class pembolehUbah {
    public static void main(String[] args) {
        String nama;
        nama = "hardeep";
    }
}
```

Penetapan nilai

Pengisytiharan &
Penetapan nilai

```
public class pembolehUbah {
    public static void main(String[] args) {
        → String nama = "hardeep";
    }
}
```

Double

```
public class bolehUbah {  
    public static void main(String[] args) {  
        double s, t, u;  
        s = 0.123; ←  
        t = 1.1;  
        u = s+t;  
    }  
}
```

Penetapan nilai

Pengisytiharan &
Penetapan nilai

```
public class bolehUbah {  
    public static void main(String[] args) {  
        } → double s = 0.123, t = 1.1, u = s+t;  
    }  
}
```

Boolean

```
public class bolehUbah {  
    public static void main(String[] args) {  
        boolean v;  
        v = true; ←  
    }  
}
```

Penetapan nilai

Pengisytiharan &
Penetapan nilai

```
public class bolehUbah {  
    public static void main(String[] args) {  
        } → boolean v = true;  
    }  
}
```

Char

```
public class bolehUbah {  
    public static void main(String[] args) {  
        char w;  
        w = 'y'; ←  
    }  
}
```

Penetapan nilai

Pengisytiharan &
Penetapan nilai

```
public class bolehUbah {  
    public static void main(String[] args) {  
        } → char w = 'y';  
    }  
}
```



Float

```
public class bolehUbah {  
    public static void main(String[] args) {  
        float y;  
        y = 342.234f;  
    }  
}
```

Penetapan nilai

Pengisytiharan &
Penetapan nilai

```
public class bolehUbah {  
    public static void main(String[] args) {  
        float y = 342.234f;  
    }  
}
```

Pengisytiharan dan Penetapan Nilai Pemalar

Dalam dunia sebenar, terdapat beberapa nilai yang tidak akan berubah seperti segi empat mempunyai 4 sisi, satu hari mempunyai 24 jam dan nilai Pi mempunyai tiga tempat perpuluhan iaitu 3.142. serta nilai mata wang bagi Malaysia. Nilai-nilai ini akan kekal dan tidak akan berubah. Dalam pengaturcaraan Java, nilai-nilai ini masih sama apabila dijadikan sebagai boleh ubah. Pemboleh ubah ini dikenal sebagai pemalar. Nilai pemalar tidak akan berubah semasa menjalankan arus cara. Sintaks untuk mengisytiharkan pemalar dalam Java adalah seperti yang berikut:

```
final int BILANGAN_HARI_DALAM_SEMINGGU = 7;
```

```
final double Pi = 3.14;
```

```
static final double = 4.14;
```



16

Pengisytiharan boleh ubah dan penentuan boleh ubah

Kerja Individu

Anda dikehendaki membuat satu arus cara yang mempunyai nilai seperti di bawah:

- 1 Pengunjung ke Malaysia
- 2 199
- 3 L

Berdasarkan nilai yang diberikan, sila buat pengisytiharan boleh ubah mengikut jenis data yang sesuai dan masukkan nilai kepada boleh ubah tersebut.



BAB 1 PENGATURCARAAN



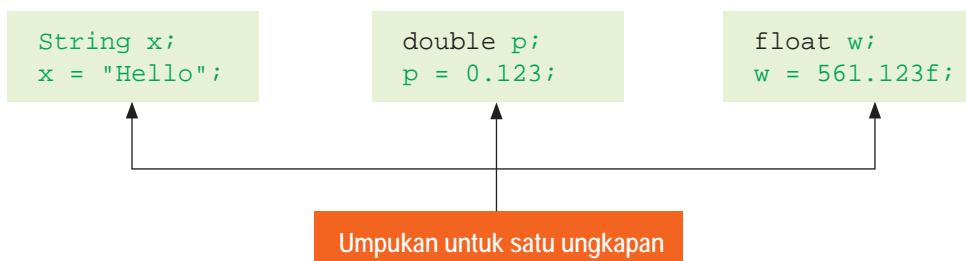


1.3.5 Penggunaan Penyataan Umpukan dan Penyataan Aritmetik

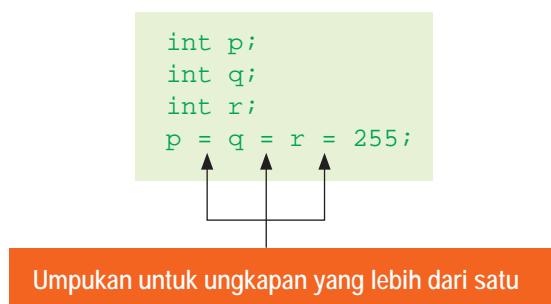
Cipta Permainan
Aritmetik
<http://goo.gl/YbsnEi>

Semasa menulis atur cara, terdapat beberapa jenis penyataan yang boleh digunakan. Antaranya ialah Penyataan Umpukan dan Penyataan Aritmetik.

Penyataan Umpukan boleh terdiri daripada satu atau lebih ungkapan yang lain. Penyataan umpukan merujuk “sama dengan” atau simbol “=”. Penyataan umpukan akan memberikan nilai kepada pemboleh ubah. Contohnya `int x = y`. `y` adalah nilai yang diumpukkan kepada pemboleh ubah `x`. Contoh penyataan umpukan ialah:



Rajah 1.22(a) Umpukan untuk satu ungkapan



Rajah 1.22(b) Umpukan untuk ungkapan yang lebih dari satu

Penyataan aritmetik boleh terdiri daripada satu atau lebih operasi aritmetik. Penyataan ini merujuk operasi aritmetik iaitu operasi penambahan, penolakan, pendaraban dan pembahagian. Unit pemproses utama boleh membaca operasi aritmetik dari kiri ke kanan sahaja. Sekiranya pengatur cara ingin menjalankan operasi darab terlebih dahulu, penggunaan simbol kurungan perlu diutamakan. Contoh:



Rajah 1.23 Contoh penyataan aritmetik

Lihat jadual 1.8 yang menunjukkan penyataan aritmetik dan contoh atur cara.

Jadual 1.8 Penyataan aritmetik dan contoh atur cara

Notasi Pengaturcaraan	Notasi Matematik	Penerangan	Contoh Atur cara	Contoh Output
+	+	Operasi tambah	<pre>public class operasiTambah { public static void main(String[] args) { int i = 10; int j = 20; i = i + j; System.out.println("Hasil Tambah ialah: " + i); } }</pre>	Hasil Tambah ialah: 30
-	-	Operasi tolak	<pre>public class operasiTolak { public static void main(String[] args) { int i = 30; int j = 20; i = i - j; System.out.println("Hasil Tolak ialah: " + i); } }</pre>	Hasil Tolak ialah: 10
*	x	Operasi darab	<pre>public class operasiDarab { public static void main(String[] args) { int i = 30; int j = 20; i = i * j; System.out.println("Hasil Darab ialah: " + i); } }</pre>	Hasil Darab ialah: 600
/	÷	Operasi bahagi	<pre>public class operasiTolak { public static void main(String[] args) { int i = 30; int j = 2; i = i / j; System.out.println("Hasil Bahagi ialah: " + i); } }</pre>	Hasil Bahagi ialah: 15

Setiap kali peperiksaan dilaksanakan di sekolah, semua murid akan dapat mengetahui :

- Jumlah markah keseluruhan
- Purata markah secara keseluruhan
- Peratus markah secara keseluruhan

Berdasarkan situasi di atas, umpukan nilai dan bina pernyataan aritmetik untuk mendapatkan perkara (a) hingga (c) mengikut mata pelajaran yang ditawarkan di sekolah anda.

1.3.6 Penulisan Atur Cara untuk Memasukkan Input daripada Papan Kekunci dan Memaparkan Output

Sesetengah atur cara telah diumpukan nilai atau input bagi operasi pengiraan atau pernyataan dalam atur cara tersebut. Ada pula atur cara yang membenarkan nilai atau input dimasukkan selepas atur cara sudah siap dibina iaitu semasa atur cara dilaksanakan dengan menggunakan papan kekunci. Atur cara perlu ditambah dengan pengekodan seperti di sebelah.

```
import java.util.Scanner;
Scanner input = new Scanner(System.in);
System.out.println("Sila masukkan nilai tapak : ");
int tapak = input.nextInt();
```

Lihat dua perbezaan atur cara di bawah.

CONTOH (2)

Atur cara yang membenarkan input dimasukkan menggunakan papan kekunci

```
import java.util.Scanner;
public class luasSegiTiga {
    public static void main(String[] args) {
        Scanner input = new Scanner(System.in);
        System.out.println("Sila masukkan nilai tapak ");
        int tapak = input.nextInt();

        System.out.println("Sila masukkan nilai tinggi ");
        int tinggi = input.nextInt();

        double luas = (1.0/2)* tapak * tinggi;
        System.out.println("Luas segi tiga ialah : " + luas);
    }
}
```

Output

Tulisan yang berwarna hijau ialah input yang dimasukkan dengan papan kunci



```
Sila masukkan nilai tapak :
6
Sila masukkan nilai tinggi :
4
Luas segi tiga ialah : 12.0
```



Atur cara yang telah diumpukan nilai atau input.

```
public class luasSegiTiga2 {  
    public static void main(String[] args) {  
        int tapak = 6;  
        int tinggi = 4;  
        double luas = (1.0/2)* tapak * tinggi;  
        System.out.println("Luas segi tiga ialah : " + luas);  
    }  
}
```

Luas segi tiga ialah : 12.0

Output

Input tidak perlu dimasukkan kerana atur cara mengira jawapan berdasarkan nilai yang telah dimasukkan dalam atur cara.

CONTOH (2)

Atur cara yang membenarkan input dimasukkan menggunakan papan kekunci.

```
import java.util.Scanner;  
  
public class infoAnda {  
  
    public static void main(String[] args) {  
        Scanner input = new Scanner(System.in);  
        System.out.println("Siapakah nama anda? : ");  
        String nama = input.nextLine();  
  
        System.out.println("Berapakah umur anda? : ");  
        int umur = input.nextInt();  
  
        System.out.println("Apakah hobi anda? : ");  
        String hobi = input.nextLine();  
  
        nama = nama;  
        umur = umur;  
        hobi = hobi;  
  
        System.out.println("Nama : " + nama);  
        System.out.println("Umur : " + umur);  
        System.out.println("Hobi : " + hobi);  
    }  
}
```

Siapakah nama anda? :
Haziq
Berapakah umur anda? :
7
Apakah hobi anda? :
Bermain
Nama : Haziq
Umur : 7
Hobi : Bermain

Output

Tulisan yang berwarna hijau ialah input yang dimasukkan dengan papan kekunci.

Atur cara yang telah diumpukan nilai atau input.

```
public class infoAnda2 {  
    public static void main(String[] args) {  
  
        String nama;  
        int umur;  
        String hobi;  
  
        nama = "Haziq";  
        umur = 7;  
        hobi = "Bermain";  
  
        System.out.println("Nama : " + nama);  
        System.out.println("Umur : " + umur);  
        System.out.println("Hobi : " + hobi);  
    }  
}
```

Nama : Haziq
Umur : 7
Hobi : Bermain

Output

Input tidak perlu dimasukkan kerana atur cara memaparkan output berdasarkan nilai atau input yang telah dimasukkan dalam atur cara



18

Memasukkan input daripada papan kekunci dan memaparkan output

Kerja Individu

Berikut merupakan cadangan masalah yang telah diberikan.

Cadangan masalah:

- 1 Mengira luas sebuah bilik.
 - 2 Mengira purata lima nombor.
 - 3 Mencari harga jualan suatu barang apabila diberikan potongan baucar sebanyak RM3.00.
 - 4 Mengira luas dinding termasuk tingkap dan pintu.
 - 5 Mengira luas dinding tidak termasuk sebuah pintu dan dua buah tingkap.
- Pilih salah satu penyelesaian masalah pengiraan dalam matematik, bina satu atur cara yang membenarkan input dimasukkan dengan menggunakan papan kekunci dan paparkan outputnya.

(Nota: Gunakan nama boleh ubah anda sendiri)

Latihan Formatif 1.3

- 1 Namakan tiga contoh pemboleh ubah yang sering digunakan untuk membina Daftar Murid.
- 2 Pengatur cara ingin membangunkan sebuah atur cara mengenai penghitungan gaji. Mengapakah pengatur cara perlu mengambil kira jenis data dan kapasiti ingatan komputer?



- 3 Tentukan jenis data yang akan digunakan untuk mewakili data bagi atur cara penghitungan markah murid.

Data	Jenis data
Nama murid	
Jantina	
Markah	
Jumlah markah	
Purata markah	

- 4 Mengapa laras pemboleh ubah perlu diisyiharkan?

- 5 Berdasarkan pengisytiharan yang berikut, bina satu atur cara yang boleh memaparkan output bagi penambahan dua nombor tanpa sebarang ralat.

```
int nombor_pertama, nombor_kedua, jawapan; //pengisytiharan pemboleh ubah
```

- 6 Berdasarkan rajah yang ditunjukkan di bawah, kenal pasti dan labelkan pemboleh ubah sejagat dan pemboleh ubah setempat.

```
public class A {  
    static int nombor = 10;  
    double data = 50;  
  
    public static void main(String[] args) {  
        int n = 90;  
        int t = 190;  
    }  
}
```

- 7 Berdasarkan proses yang berikut, bina satu atur cara penghitungan yang boleh memaparkan output luas tanpa sebarang ralat.

```
luas = 0.5 * tinggil * tapak1;
```

- 8 Sebagai seorang pengatur cara, anda dikehendaki membangunkan sebuah atur cara penghitungan aritmetik, iaitu pembahagian dua nombor. Semasa membangunkan atur cara, nyatakan jenis data yang sesuai digunakan untuk atur cara pembahagian dua nombor tersebut.



-  9 Berdasarkan atur cara yang berikut, kenal pasti kesilapan yang telah dilakukan.

```
import java.util.Scanner
public class purataBerat {

    public static void main(String[] args) {
        Scanner input = new (System.in);
        System.out.println("Sila masukkan berat 1 : ");
        int berat1 = input.nextInt();

        System.out.println("Sila masukkan berat 2 : ");
        int berat2 = input.nextInt();

        System.out.println("Sila masukkan berat 3 : ");
        int berat3 = input.nextInt();

        char purata = (berat1 + berat2 + berat3) * 3;
        System.out.println("Purata berat ialah : " + purata);
    }
}
```

-  10 Anda dikehendaki membangunkan satu atur cara yang mempunyai data jenis *String*, *int*, *double/float* dan *boolean*. Atur cara tersebut mestilah mengenai diri anda seperti nama, umur, nombor kegemaran dan kegemaran anda. Lihat cadangan yang berikut:

- (a) Siapakah nama anda?
- (b) Berapakah umur anda?
- (c) Apakah nombor kegemaran anda?
- (d) Adakah anda suka membaca atau tidak?

Bina atur cara yang bebas ralat.



STANDARD PEMBELAJARAN

1.4.1 Menerangkan aliran struktur kawalan pilihan

- (i) *If-else*
- (ii) *Switch-case*

1.4.2 Menulis atur cara menggunakan struktur kawalan pilihan dengan menggabungkan operator hubungan dan operator logik

1.4.3 Menerangkan aliran struktur kawalan ulangan

- (i) *For*
- (ii) *While*
- (iii) *Do-while*

1.4.4 Menulis atur cara menggunakan struktur kawalan ulangan yang melibatkan

- (i) *Operator increment*
(`++`) dan *decrement*
(`--`)
- (ii) *Class Math* (*method random*)
- (iii) Boolean



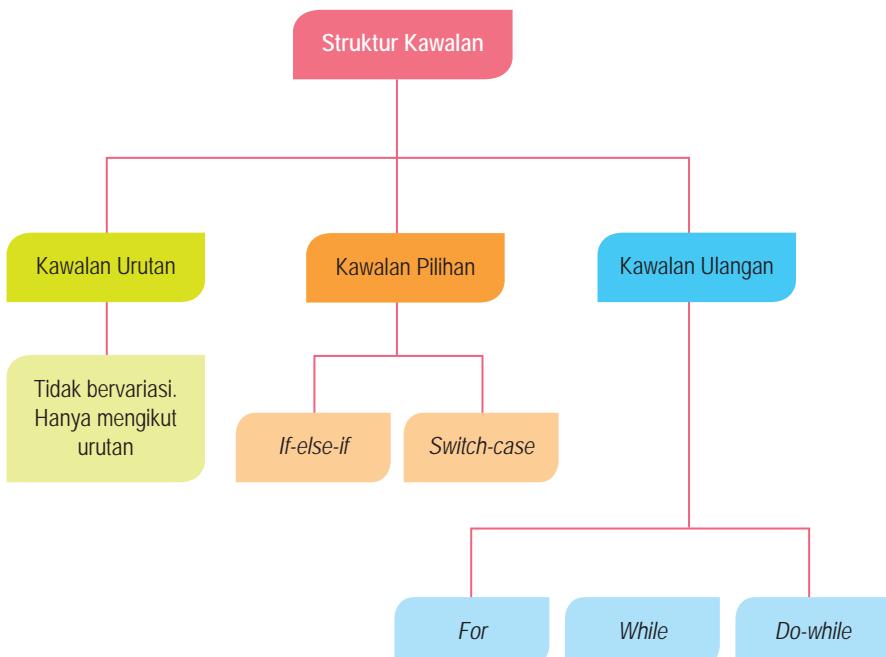
Kerjaya Sains Komputer

Penulis teknikal bertanggungjawab untuk menulis penerangan bagaimana atur cara komputer berfungsi.

Struktur Kawalan

Arahan-arahan dalam susunan linear biasanya digunakan untuk mendapatkan input pengguna, memaparkan hasil atau output atau mengisytiharkan dan mengumpulkan nilai-nilai bagi boleh ubah. Seperti yang telah dipelajari dalam subtopik algoritma, langkah-langkah yang terdapat dalam algoritma komputer bukan sahaja linear tetapi bercabang dan juga berulang. Dalam struktur yang bercabang, langkah-langkahnya bolehkan keputusan dibuat dengan serta-merta. Struktur berulang pula membenarkan langkah yang sama diulang berkali-kali.

Dalam bahasa pengaturcaraan, terdapat persamaan unsur bagi struktur kawalan dan struktur algoritma, iaitu kawalan pilihan dan kawalan ulangan. Beberapa variasi wujud dalam struktur kawalan pilihan dan kawalan ulangan tetapi tidak bagi kawalan urutan. Bagi kawalan urutan, atur cara dilaksanakan mengikut urutan. Misalnya, seorang guru mengajar muridnya mengikut bab yang telah diaturkan untuknya. Bagi kawalan pilihan dan kawalan ulangan, setiap variasi mempunyai struktur tertentu dengan kelebihan dan kekurangan masing-masing. Oleh itu, murid perlu menyesuaikan masalah dengan struktur kawalan yang sesuai.



Rajah 1.24 Jenis-jenis struktur kawalan dalam pengaturcaraan





Aplikasi Harian

Terdapat aplikasi yang jarang diketahui yang menggunakan struktur kawalan seperti aplikasi kawalan dalam kereta, aplikasi suhu peti ais, ketuhar gelombang mikro, pencetak dan penyaman udara.



BBC Bitesize Selection in programming
<http://goo.gl/91MKkt>

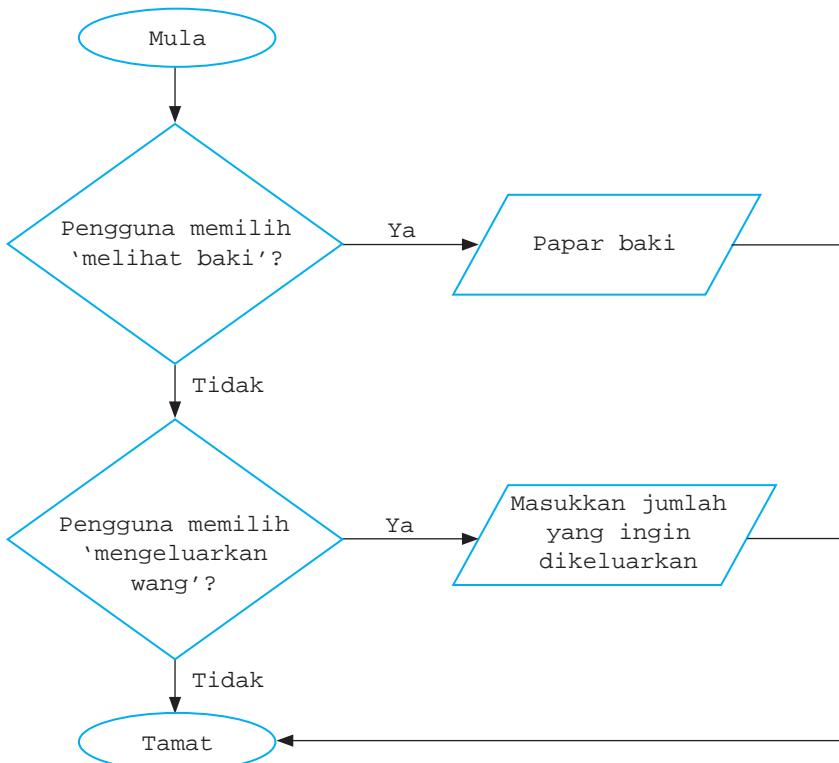


How to use Boolean expressions in Java?
<http://goo.gl/DNtv8Z>

1.4.1 Struktur Kawalan Pilihan

Berdasarkan input tertentu, arus cara komputer boleh membuat keputusan. Keputusan atau pilihan itu telah direkodkan dalam arus cara tersebut. Kawalan pilihan ialah mekanisme yang membolehkan keputusan atau pemilihan dibuat secara automatik.

Sebagai contoh, mesin pengeluaran wang automatik (ATM) kini, mempunyai banyak fungsi seperti untuk melihat baki, mengeluarkan wang, memindahkan wang dan membayar bil. Berdasarkan pilihan yang dimasukkan, perisian di dalam ATM akan membuat keputusan apakah paparan yang akan ditunjukkan kepada pengguna dan bagaimana untuk memproses input yang dimasukkan. Penyataan syarat *boolean*, seperti "adakah pengguna memilih untuk melihat baki?" digunakan untuk menguji nilai input yang dimasukkan dan ini seterusnya akan menentukan set atau blok arahan yang akan dilaksanakan (Rajah 1.25).



Rajah 1.25 Penyataan syarat *boolean*

Penyataan tentang komputer membuat keputusan disebut sebagai **penyataan bersyarat *boolean***. Syarat *boolean* tersebut membolehkan perbandingan memboleh ubah, sifat objek atau nilai yang dilakukan melalui operator hubungan atau operator logikal. Perbandingan ini memberikan keputusan dalam bentuk data jenis *boolean*.

Nilai Boolean

Selain daripada jenis-jenis data seperti integer, *double* atau *String*, terdapat satu lagi jenis data lain yang lazim digunakan terutamanya dalam struktur kawalan, iaitu jenis data *boolean*. Data jenis *boolean* hanya mempunyai dua nilai yang mungkin, iaitu Ya/Benar (*True*) atau Tidak/Palsu (*False*).

Contoh (23)

Kita sedia maklum nilai Pi (π) dalam matematik ialah 3.142. Buat penilaian bagi ungkapan-ungkapan yang berikut:

- (a) $\text{Pi} == 3$
- (b) $\text{Pi} > 0$
- (c) $\text{Pi} < 10$
- (d) $\text{Pi} != 5$
- (e) $\text{Pi} > 5$

- Jawapan:
- (a) *False*
 - (b) *True*
 - (c) *True*
 - (d) *True*
 - (e) *False*



Tahukah
Anda

Boolean telah dibangunkan oleh ahli penyelidik Matematik dan komputer iaitu George Boole (1815-1864). Sistem *boolean* dibina berdasarkan pendekatan binari yang melibatkan hanya dua objek iaitu:

- Ya/tidak
- Benar/palsu
- Buka/tutup
- Sifar/satu

Sumber: www.kerry.net

Penyataan bersyarat *boolean* ditulis menggunakan struktur kawalan pilihan. Terdapat tiga bentuk struktur kawalan pilihan seperti yang berikut:

- *if*
- *if-else*
- *if-else-if*
- *switch-case*

Kawalan Pilihan *If*

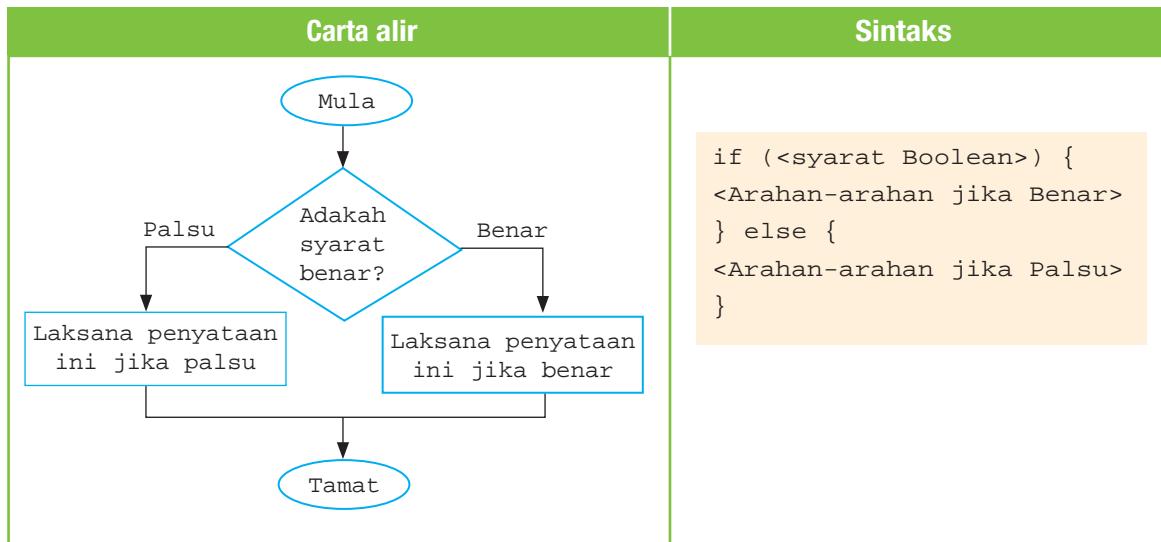
Struktur kawalan ini dipanggil *if* (Rajah 1.26). Struktur kawalan *if* hanya akan melaksanakan pernyataan-pernyataan tertentu seperti memproses data melalui ungkapan sekiranya syarat adalah benar.

Carta alir	Sintaks
<pre>graph TD; M([Mula]) --> D{Adakah syarat benar?}; D -- Benar --> L[Laksana kenyataan ini jika benar]; D -- Tidak --> T([Tamat]);</pre>	<pre>if (<syarat Boolean>) { <Arahan-arahan jika Benar> } Contoh: if (umur > 20) { System.out.println("Anda layak mengundi."); }</pre>

Rajah 1.26 Carta alir dan sintaks *if*

Kawalan Pilihan If-Else

Penyataan *if-else* digunakan untuk membuat keputusan dalam sesuatu aratur cara. Hasil perbandingan itu menunjukkan hasil *boolean*, iaitu Ya (Benar) atau Tidak (Palsu). Penyataan susulan bergantung kepada hasil Boolean tersebut.



Rajah 1.27 Carta alir dan sintaks *if-else*

Contoh syarat-syarat Boolean yang digunakan dalam kawalan pilihan adalah seperti berikut:

- Adakah input 10 lebih besar daripada 0?
- Adakah suhu bilik lebih tinggi daripada 27°C?
- Adakah suhu kotak mengeram telur dalam lingkungan 37 sehingga 39°C?
- Adakah pemohon berumur 16 tahun ke atas?
- Adakah markah input 40 atau ke atas?

CONTOH 24

Komputer boleh meminta umur seseorang pengguna untuk menentukan kelayakan mengundi. Syarat *boolean* untuk mengundi ialah 21 tahun. Jadi, aratur cara kawalan pilihan *if-else* yang digunakan adalah seperti yang berikut:

```
public class Contoh24{  
    public static void main(String[] args){  
        int umur=25;  
        if (umur > 20) {  
            System.out.println("Anda layak mengundi.");  
        } else {  
            System.out.println("Maaf, anda masih belum  
            cukup umur.");  
        }  
    }  
}
```

If-else statement in Java
<http://goo.gl/kkyqN9>



Kawalan Pilihan *If-Else-If*

Komputer juga boleh diprogramkan untuk membuat keputusan yang lebih kompleks, iaitu dengan menggunakan struktur kawalan *if-else-if*. Oleh hal yang demikian, komputer boleh mencuba syarat *boolean* yang baharu sekiranya syarat terdahulu menghasilkan keputusan palsu. Sekiranya syarat *boolean* menghasilkan keputusan benar, penyataan yang dipaut kepada syarat *boolean* tersebut sahaja akan dilaksanakan. Syarat-syarat *boolean* yang lain tidak akan diuji.

Carta alir	Sintaks
<pre> graph TD M([Mula]) --> D1{Adakah syarat 1 benar?} D1 -- Ya --> L1[Laksana penyataan 1] D1 -- Tidak --> D2{Adakah syarat 2 benar?} D2 -- Ya --> L2[Laksana penyataan 2] D2 -- Tidak --> D3{Adakah syarat 3 benar?} D3 -- Ya --> L3[Laksana penyataan 3] D3 -- Tidak --> L4[Laksana penyataan 4] L4 --> T([Tamat]) </pre>	<pre> if (syarat boolean1_Benar) { penyataan01 } else if (syarat boolean2_Benar) { penyataan02 } else if (syarat boolean3_Benar) { penyataan03 } else { penyataan04 } </pre>

Rajah 1.28 Carta alir dan sintaks *if-else-if*

CONTOH 25

Sebuah aplikasi menerima markah murid dan menentukan gred yang sesuai. Jadual yang berikut menunjukkan penentuan gred berdasarkan julat markah. Berikut ialah atur cara untuk mewakili pemilihan gred berdasarkan jadual yang menggunakan struktur kawalan *if-else-if*.

Markah	Gred
90–100	A+
80–89	A
75–79	A+
70–74	B+
65–69	B
60–64	C+
50–59	C
45–49	D
40–44	E
0–39	G

```
public class Contoh25{  
    public static void main(String[] args){  
        double markah=73;  
        String Gred;  
        if (markah >= 90) {  
            Gred = "A+";  
        }else if (markah >=80) {  
            Gred = "A";  
        }else if (markah >=75) {  
            Gred = "A-";  
        }else if (markah >=70) {  
            Gred = "B+";  
        }else if (markah >=65) {  
            Gred = "B";  
        }else if (markah >=60) {  
            Gred = "C+";  
        }else if (markah >=50) {  
            Gred = "C";  
        }else if (markah >=45) {  
            Gred = "D";  
        }else if (markah >=40) {  
            Gred = "E";  
        }else if (markah >=0) {  
            Gred = "G";  
        }  
        System.out.println("Markah=" +markah+"dan  
        Gred ialah"+Gred);  
    }  
}
```

Kawalan Pilihan *Switch-Case*

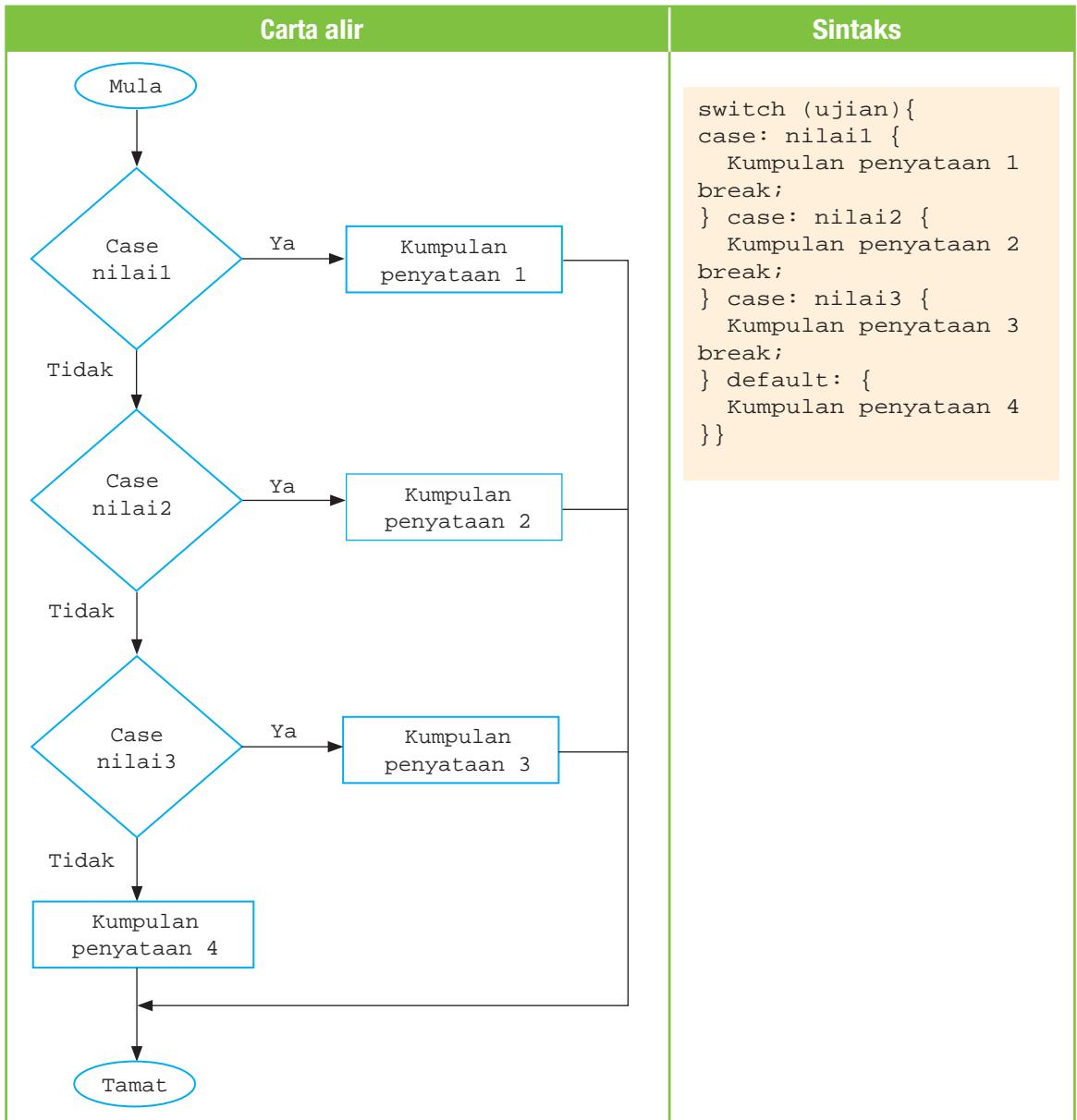
Penggunaan *if-else-if* untuk membuat sebilangan keputusan adalah rumit kerana pernyataan *if-else-if* perlu diulang banyak kali dan pengulangan tersebut boleh menimbulkan kekeliruan kepada pengguna.

Untuk mengatasi kekurangan penggunaan *if-else-if* tersebut, struktur kawalan *switch-case* boleh digunakan. Struktur ini lebih mudah difahami kerana nilai ujian dalam ‘Select Case’ dipadankan dengan beberapa calon ‘Case’.

Rajah 1.29 menunjukkan pelanggan berbaris untuk mendapatkan keperluan perkhidmatan tertentu. Keperluan dipadankan dengan kaunter yang menawarkan perkhidmatan tersebut. Input dipadankan dengan calon-calon sedia ada.



Rajah 1.29 Beberapa kaunter untuk perkhidmatan yang berlainan di pejabat pos



Rajah 1.30 Carta alir dan sintaks *switch-case*

Ujian untuk ‘*Switch*’ ialah ungkapan nombor, abjad atau rentetan. ‘*Case*’ mengandungi nilai masing-masing yang akan dipadankan dengan ujian. Jika ujian bersamaan dengan nilai selepas ‘*Case*’, maka kumpulan pernyataan selepas ‘*Case*’ akan dilaksanakan. Ungkapan ‘*Break*’ digunakan sebagai arahan untuk keluar dari blok ‘*Switch*’. Jika ungkapan ‘*Break*’ tidak disertakan, kumpulan pernyataan selepas ‘*Break*’ akan turut dilaksanakan. ‘*Default*’ kadang-kadang dimasukkan dalam blok ‘*Switch*’ sebagai langkah tambahan. Sekiranya ujian tidak bersamaan dengan mana-mana nilai ‘*Case*’, kumpulan pernyataan ‘*Default*’ akan dilaksanakan. Langkah ini menjamin bahawa setidak-tidaknya ada pernyataan yang telah dilaksanakan.

CONTOH 26

Sebuah aplikasi akan memaparkan tarikh dalam format 'dd-nama bulan-yy' (contoh: 28-April-16) Gunakan bentuk *switch-case* untuk menulis atur cara yang akan memaparkan nama bulan berdasarkan bulan yang diberikan dalam bentuk nombor.

Penyelesaian

```
import java.util.Scanner;
public class Contoh26 {
    public static void main (String [] args) {
        int hari = 23, tahun = 2016;
        Scanner input = new Scanner (System.in);
        System.out.println ("Masukkan nombor antara 1 hingga 12");
        int bulan = input.nextInt ();
        switch (bulan){
            case 1: {
                System.out.println(hari+"-Januari-"+tahun);
                break;
            } case 2: {
                System.out.println(hari+"-Februari-"+tahun);
                break;
            } case 3: {
                System.out.println(hari+"-Mac-"+tahun+"-"+tahun);
                break;
            } case 4: {
                System.out.println(hari+"-April-"+tahun);
                break;
            } case 5: {
                System.out.println(hari+"-Mei-"+tahun);
                break;
            } case 6: {
                System.out.println(hari+"-Jun-"+tahun);
                break;
            } case 7: {
                System.out.println(hari+"-Julai-"+tahun);
                break;
            } case 8: {
                System.out.println(hari+"-Ogos-"+tahun);
                break;
            } case 9: {
                System.out.println(hari+"-September-"+tahun);
                break;
            } case 10: {
                System.out.println(hari+"-Oktober-"+tahun);
                break;
            } case 11: {
                System.out.println(hari+"-November-"+tahun);
                break;
            } case 12: {
                System.out.println(hari+"-Disember-"+tahun);
                break;
            } default: {
                System.out.println("Nombor yang dimasukkan bukan diantara
1 hingga 12");
            }
        }
    }
}
```

Berikut ialah atur cara untuk mewakili pemilihan gred berdasarkan jadual yang ditunjukkan di bawah. Atur cara yang berikut menggunakan struktur kawalan *W*.

Anda ditugaskan untuk menukar atur cara di bawah kepada struktur kawalan switch case.

Markah	Gred
90–100	A+
80–89	A
75–79	A+
70–74	B+
65–69	B
60–64	C+
50–59	C
45–49	D
40–44	E
0–39	G

```
public class ContohW{
    public static void main(String[] args){
        double markah=73;
        String Gred;
        if (markah >= 90) {
            Gred = "A+";
        }else if (markah >=80) {
            Gred = "A";
        }else if (markah >=75) {
            Gred = "A-";
        }else if (markah >=70) {
            Gred = "B+";
        }else if (markah >=65) {
            Gred = "B";
        }else if (markah >=60) {
            Gred = "C+";
        }else if (markah >=50) {
            Gred = "C";
        }else if (markah >=45) {
            Gred = "D";
        }else if (markah >=40) {
            Gred = "E";
        }else if (markah >=0) {
            Gred = "G";
        }
        System.out.println("Markah=" +markah+" dan
Gred ialah"+Gred);
    }
}
```



Pada tahun 2013, Syed Muhammad yang ketika itu masih seorang murid sekolah menengah telah berjaya mencipta lima aplikasi untuk syarikat telekomunikasi terkemuka dunia iaitu *Apple*. Antara aplikasinya ialah *Surfing Fast*, *Who Created*, *Simple exCode* dan *Digital Harimau Malaya* serta permainan *Crazy Ball*.
Sumber: www.utusan.com.my



Switch statements in Java
<http://goo.gl/B7rW7h>



Tahukah Anda?

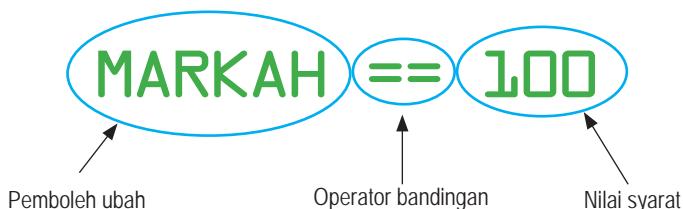
Operator hubungan juga dikenal sebagai operator bandingan. Hal ini adalah kerana operator hubungan digunakan untuk menguji hubungan antara lebih besar dan lebih kecil bagi memboleh ubah dan nilai syarat.

1.4.2 Operator Hubungan dan Operator Logikal dalam Struktur Kawalan Pilihan

Dua kategori operator yang sering digunakan dalam ungkapan syarat ialah operator hubungan dan operator logikal. Operator hubungan digunakan untuk membandingkan dua nilai bagi menghasilkan keputusan *boolean*. Operator logikal pula digunakan untuk menghasilkan beberapa ungkapan *boolean* bagi menghasilkan syarat yang lebih kompleks.

Operator Hubungan

Penyataan bersyarat *Boolean* menggunakan ungkapan yang bertujuan untuk membuat perbandingan. Input pengguna akan disimpan dalam memboleh ubah. Pemboleh ubah tersebut akan dibandingkan menerusi nilai-nilai dalam syarat *Boolean*. Operator-operator perbandingan membuatkan perbandingan seperti dalam contoh-contoh sebelumnya.



Rajah 1.31 Ungkapan yang mempunyai operator hubungan "="

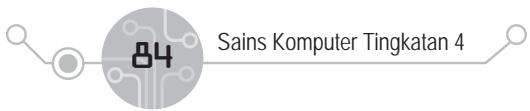
Dalam Rajah 1.31, "markah" ialah suatu pemboleh ubah. Pemboleh ubah mengandungi nilai tertentu yang telah diumpukan kepada pemboleh ubah tersebut sewaktu atau selepas pengisytiharan pemboleh ubah dibuat. Katakan pemboleh ubah markah telah diumpukan dengan nilai 90. Maka,

keputusan kenyataan syarat *boolean* "markah adalah sama dengan 100" adalah tidak benar. Bagaimana sekiranya pemboleh ubah diumpuk dengan nilai 100? Keputusan untuk kenyataan syarat *Boolean* "markah adalah sama dengan 100" adalah benar.

Jadual 1.9 menunjukkan operator-operator hubungan yang biasa digunakan dalam struktur kawalan pilihan.

Jadual 1.9 Operator-operator hubungan

Operator Hubungan	Penerangan
==	Sama dengan
!=	Tidak sama dengan
>	Lebih besar daripada
>=	Lebih besar daripada atau sama dengan
<	Kurang daripada
<=	Kurang daripada atau sama dengan



CONTOH (27)

Katakan pemboleh ubah nombor menerima input integer -5 . Apakah keputusan ungkapan bandingan yang berikut?

Ungkapan	Keputusan (benar?)
Nombor == 0	Tidak
Nombor != 0	Ya
Nombor > 0	Tidak
Nombor < 0	Ya



Operators
goo.gl/HevGUJ

CONTOH (28)

Katakan pemboleh ubah nombor menerima input integer 24 . Apakah keputusan ungkapan bandingan yang berikut?

Ungkapan	Keputusan (benar?)
Nombor == 0	Tidak
Nombor != 0	Ya
Nombor > 0	Ya
Nombor < 0	Tidak



Berdasarkan contoh 27, apakah keputusan ungkapan bandingan yang berikut?

Ungkapan	Keputusan (benar?)
Nombor == 7	
Nombor != -5	
Nombor > 8	
Nombor < -2	

CONTOH (29)

Katakan pemboleh ubah nombor menerima input integer 0 . Apakah keputusan ungkapan bandingan yang berikut?

Ungkapan	Keputusan (benar?)
Nombor == 0	Ya
Nombor != 0	Tidak
Nombor > 0	Tidak
Nombor < 0	Tidak

CONTOH (30)

Katakan pemboleh ubah nombor menerima input integer 0 . Apakah keputusan ungkapan bandingan yang berikut?

Ungkapan	Keputusan (benar?)
Nombor == 10	Tidak
Nombor != 0	Tidak
Nombor > -5	Ya
Nombor < 12	Ya

Untuk menguji syarat *boolean*, letakkan ungkapan tersebut dalam struktur kawalan pilihan yang telah dipelajari. Lihat contoh yang berikut.

CONTOH (31)

Uji satu nombor untuk mengenal pasti integer positif.

```
public class Contoh31a{
    public static void main(String[] args){
        int nombor=15;
        if (nombor > 0)
            System.out.println("Nombor ini adalah integer positif");
        else
            System.out.println("Nombor ini bukan integer positif");
    }
}
```

Contoh output:

```
Command Prompt
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh31a
Nombor ini adalah integer positif
```

```
public class Contoh31b{
    public static void main(String[] args){
        int nombor=-7;
        if (nombor > 0)
            System.out.println("Nombor ini adalah integer positif.");
        else if (nombor == 0)
            System.out.println ("Nombor ini adalah sifar.");
        else
            System.out.println ("Nombor ini adalah integer negatif.");
    }
}
```

Contoh output:

```
Command Prompt
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh31b
Nombor ini adalah integer negatif.
```



Uji Minda

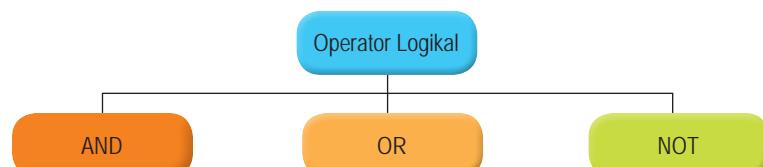
Berdasarkan contoh 31, uji satu nombor untuk mengenal pasti integer negatif.



Bab Penaakulan dalam Matematik menggunakan konsep AND, OR dan NOT yang sama!

Operator Logikal

Dalam suatu kenyataan *if-else*, dua atau lebih syarat *boolean* boleh diuji serentak dengan menggabungkan syarat *boolean* melalui operator logikal. Operator-operator logikal yang lazim digunakan ialah AND, OR dan NOT.



Rajah 1.32 Operator-operator logikal

Operator Logikal AND

Operator logik AND digunakan apabila dua atau lebih syarat *boolean* perlu digabungkan dan kesemua syarat perlu benar.

Sebagai contoh, markah peperiksaan haruslah berada dalam julat 0 hingga 100 dan boleh ditulis sebagai gabungan markah ≥ 0 dan juga markah ≤ 100 . Operator yang sesuai dalam hal ini ialah operator AND. Operator AND ditulis menggunakan simbol '&&'. Oleh itu, syarat-syarat *boolean* tadi boleh ditulis sebagai satu ungkapan seperti yang berikut:

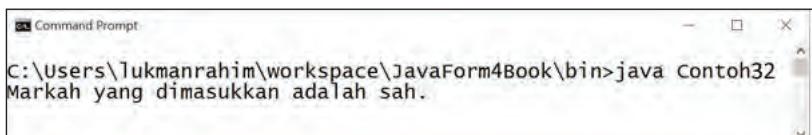
```
Markah >=0 && Markah <= 100
```

CONTOH (32)

Tentukan markah input berada dalam julat yang sah, iaitu 0 hingga 100.

```
public class Contoh32{  
    public static void main(String[] args){  
        int markah=55;  
        if (markah >=0 && markah <=100)  
            System.out.println("Markah yang  
dimasukkan adalah sah.");  
        else  
            System.out.println("Markah yang  
dimasukkan adalah tidak sah.");  
    }  
}
```

Contoh output:



```
Command Prompt  
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh32  
Markah yang dimasukkan adalah sah.
```

Berdasarkan Contoh 32, jelaslah bahawa kesemua syarat *boolean* perlu dipenuhi. Jika terdapat salah satu syarat *boolean* yang tidak benar, maka keputusan adalah tidak benar. Lihat Jadual 1.10.

Jadual 1.10 Contoh penggunaan operator AND

markah ≥ 0	markah ≤ 100	markah $\geq 0 \text{ && } \text{markah} \leq 100$
False	False	False
False	True	False
True	False	False
True	True	True



Operasi	Maksud
a && b	logik AND
a b	logik OR
a&b	logik Boolean AND
a b	logik Boolean OR
!a	logik NOT



True/False
goo.gl/xEPKUd



Tahukah Anda?

Cara yang mudah untuk mengingati logik AND dan logik OR.

AND : Apabila anda inginkan semua menjadi betul

OR: Apabila anda hanya inginkan satu sahaja menjadi betul.

Cuba cara ini untuk mengingati logik AND dan logik OR.

Operator Logikal OR

Operator logik OR digunakan apabila dua atau lebih syarat *boolean* perlu digabungkan dan hanya salah satu syarat *boolean* perlu benar. Operator logikal OR ditulis menggunakan simbol '`||`'. Mari kita bandingkan operator ini dengan operator logikal AND.

Contoh (33)

Baju yang dijemur perlu diangkat apabila hari sudah malam atau akan hujan. Tulis atur cara bagi situasi yang diberikan.

```
public class Contoh33{  
    public static void main(String[] args){  
        boolean malam=true;  
        boolean hujan=false;  
        if (malam || hujan)  
            System.out.println("Angkat baju.");  
    }  
}
```

Contoh output:

```
Command Prompt  
C:\Users\Tukmanrahim\workspace\JavaForm4Book\bin>java Contoh33  
Angkat baju.
```

Jelaslah di sini bahawa cuma salah satu syarat *boolean* diperlukan dan tidak semestinya kedua-dua syarat *boolean* perlu dipenuhi. Lihat Jadual 1.11.

Jadual 1.11 Contoh penggunaan operator OR

markah \geq 0	markah \leq 100	markah \geq 0 markah \leq 100
False	False	False
False	True	True
True	False	True
True	True	True

Operator Logikal NOT

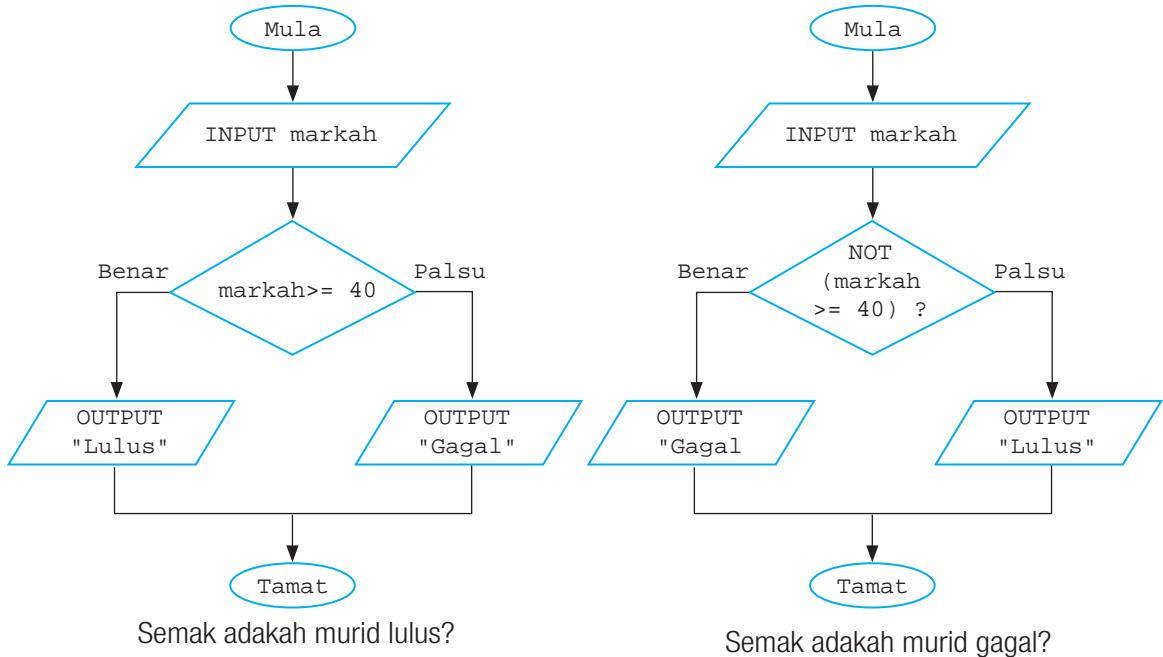
Jadual 1.12 Jadual kebenaran operator NOT

x	NOT x
TRUE	FALSE
FALSE	TRUE

Operator logikal NOT menukar nilai *boolean* kepada lawannya. Katakan boleh ubah x ialah jenis *boolean*. Nilai x adalah *False*. Oleh itu, NOT x adalah *True*. Bagaimanakah pula jika nilai asal x adalah *True*? Ini bermaksud NOT x adalah *False*. Lihat jadual kebenaran pada Jadual 1.12 yang merumuskan kegunaan operator NOT.

Operator logikal NOT diletakkan sebelum syarat *boolean* untuk menukar status kepada lawannya. Operator NOT ditulis menggunakan simbol '`!`'. Sebagai contoh, markah ≥ 40 menunjukkan murid lulus. Apabila NOT ditambah (`!markah >=40`), maksud pernyataan tersebut berubah dan akan menunjukkan murid gagal.





Rajah 1.33 Operator NOT memberikan maksud lawannya

Contoh 34

Markah lulus ialah 40 dan ke atas. Jika status lulus murid adalah tidak benar, murid tersebut terpaksa mengulangi ujian.

```

public class Contoh34{
    public static void main(String[] args){
        boolean lulus;
        int markah=83;
        if(markah>=40)
            lulus=true;
        else
            lulus=false;
        if(!lulus)
            System.out.println("Anda perlu
                mengulangi ujian");
        else
            System.out.println("Anda lulus");
    }
}

```

Contoh output:

```

C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh34Baru
Anda Lulus

```



*Logical operations and Truth
Tables*
goo.gl/nVNLCz



Logic Operators in Java
goo.gl/xVYszQ



Logical operators
goo.gl/gBi4iM





Uji Minda

Apakah perkaitan antara Hukum DeMorgan dengan operator logik?

Penggabungan Operator Hubungan dan Operator Logikal dalam Struktur Kawalan Pilihan

Operator hubungan dan operator logikal boleh digabungkan dalam struktur kawalan pilihan. Dalam contoh di atas, operator hubungan $<=$ dan $>=$ telah digabungkan dengan operator logikal AND dalam ungkapan syarat boolean:

```
Markah >=0 && Markah <= 100
```

Satu lagi contoh gabungan melibatkan operator hubungan dan operator logikal OR adalah seperti yang berikut:

Contoh (35)

Jika suhu badan adalah lebih tinggi daripada 37°C ataupun 98.6°F , maka seseorang itu mungkin menghidap demam panas.



Tuliskan atur cara bagi situasi yang digambarkan.

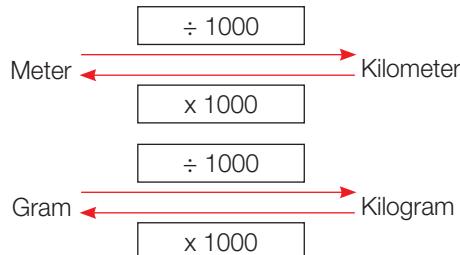
```
public class Contoh35{
    public static void main(String[] args){
        double celcius=39, fahrenheit=97;
        if (celcius>37 || fahrenheit>98.6)
            System.out.println("Anda mungkin
demam panas.");
    }
}
```

Contoh output:

```
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh35
Anda mungkin demam panas.
```



Anda diberikan tugas oleh guru anda untuk menukar unit meter (m) kepada kilometer (km) dan kilogram (kg) kepada gram (g). Rumus bagi setiap pertukaran unit yang dinyatakan adalah seperti yang berikut:

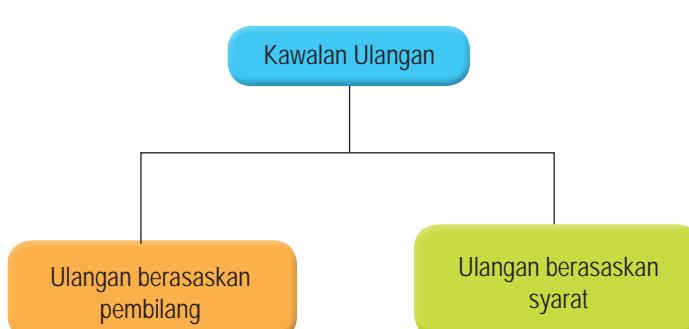


Berdasarkan rumusan yang diberikan, tulis atur cara bagi hasil pertukaran bagi nilai yang berikut:

- 500 g kepada kg
- 0.8 km kepada m
- 1.7 kg kepada g
- 400 m kepada km

1.4.3 Struktur Kawalan Ulangan

Sesetengah pernyataan diletakkan dalam blok yang perlu diulang. Hal ini berlaku kerana ulangan dibuat untuk bilangan kali tertentu atau sehingga syarat *boolean* tertentu tidak lagi benar. Lihat Rajah 1.34.



Repetition statements
goo.gl/vC6jWD

Rajah 1.34 Jenis-jenis struktur kawalan ulangan

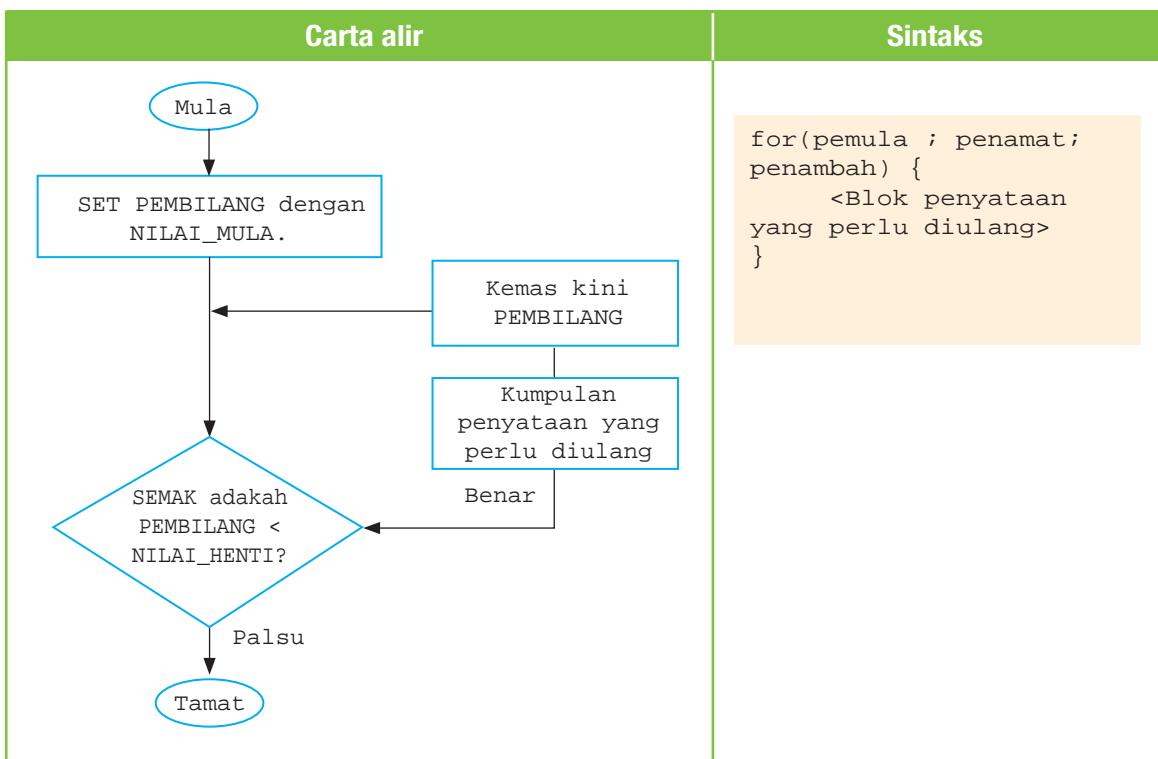
Dalam kehidupan harian, anda boleh memerhatikan tindakan alat-alat elektronik yang berulang. Contoh struktur kawalan ulangan ialah alat penyaman udara. Penyaman udara akan beroperasi apabila suhu bilik naik dan berhenti apabila suhu bilik sejuk. Suhu bilik merupakan penentu, sama ada proses penyejukan perlu berulang ataupun tidak.



Conditionals and Loops
goo.gl/rSPeZY

Kawalan Ulangan For

Kawalan *for* berulang untuk bilangan kali tertentu. Ini ditentukan oleh pemboleh ubah pembilang yang bermula dengan nombor indeks tertentu seperti 0 atau 1. Nombor indeks akan ditambah secara automatik pada akhir blok penyataan. Penambahan biasanya bernilai 1 tetapi boleh ditetapkan di bahagian penambahan. Penambahan ini akan dibuat setiap kali blok kenyataan telah diulang dan akan berlanjut sehingga syarat *boolean* berulang menjadi tidak benar.



Rajah 1.35 Carta alir dan sintaks untuk kawalan ulangan *for*

- **pemula**: Pengisytiharan pemboleh ubah pembilang dengan nilai mula. Pembilang mengesan bilangan ulangan. Sebagai contoh, `int i=1;` mengisytiharkan *i* sebagai pemboleh ubah pembilang dan pengiraan bermula dengan 1.
- **penamat**: Ungkapan yang menjadi syarat untuk menghentikan gelung. Sekiranya pemboleh ubah yang diisyiharkan dalam pemula adalah *i*, maka dalam penamat, tuliskan ungkapan `i <= 10` sekiranya ulangan dibenarkan sehingga pusingan ke-10. Bagaimanakah sekiranya ulangan dibenarkan sebanyak 100 kali?
- **penambah**: Ungkapan untuk mengemaskinikan nilai pemboleh ubah pembilang pada setiap gelungan. Suatu nilai integer, biasanya 1, ditambahkan kepada pembilang selepas setiap ulangan. Sebagai contoh, ungkapan `i++`; menambah 1 kepada pembilang. Ini juga boleh ditulis sebagai `i=i+1`;

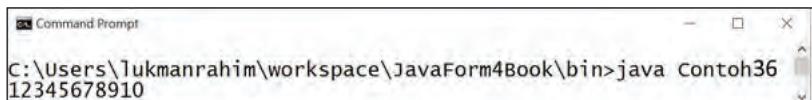
CONTOH (36)

Tuliskan atur cara untuk gelung *for* mencetak nombor 1 hingga 10.

Penyelesaian

```
public class Contoh36{
    public static void main(String[] args){
        int i;
        for(i=1;i<=10;i+=1){
            System.out.print(i);
        }
    }
}
```

Contoh output:



```
Command Prompt
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh36
12345678910
```

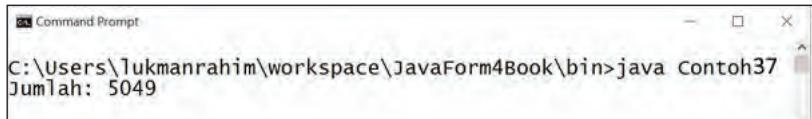
CONTOH (37)

Gunakan gelung *for* untuk menentukan hasil jumlah $2+3+4+5+\dots+99+100$.

Penyelesaian

```
public class Contoh37{
    public static void main(String[] args){
        int jumlah=0;
        for(int i=2;i<=100;i+=1){
            jumlah = jumlah + i;
        }
        System.out.println("Jumlah: " + jumlah);
    }
}
```

Contoh output:



```
Command Prompt
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh37
Jumlah: 5049
```

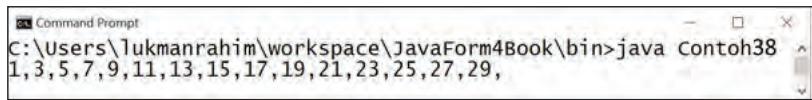
CONTOH (38)

Tuliskan atur cara untuk mempamerkan semua integer ganjil dari -30 sehingga 30.

Penyelesaian

```
public class Contoh38{
    public static void main(String[] args){
        for(int i=0;i<=30; i+=1){
            if ((i % 2) == 1)
                System.out.print(i + ", ");
        }
    }
}
```

Contoh output:



```
Command Prompt
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh38
1,3,5,7,9,11,13,15,17,19,21,23,25,27,29,
```

Tahukah Anda?

Langkah pelaksanaan kawalan *for*:

- 1 Pengumpulan nilai awal kepada pembilang
- 2 Ungkapan <ujian> akan mula dinilai
- 3 Jika nilai tersebut adalah benar, maka <pernyataan> akan dilaksanakan
- 4 Pembilang dikemas kini menggunakan <pengemasikan>
- 5 Ulang langkah dua hingga empat
- 6 Tamat gelung



Uji Minda

- (a) Gunakan gelung *for* untuk menentukan hasil jumlah $50+51+\dots+59+60$
- (b) Tuliskan atur cara untuk pamerkan semua integer genap dari -10 sehingga 20.





Zaman dunia di hujung jari pada hari ini telah memudahkan semua urusan kehidupan manusia. Jika dahulu, semua urusan bank memerlukan seseorang individu untuk pergi ke bank, kini semuanya hanya boleh dilakukan melalui urusan perbankan atas talian. Atas sumbangan pakar pemprograman, sistem perbankan atas talian memudahkan orang ramai untuk membuat urusan perbankan jika tidak mempunyai masa untuk ke bank.

CONTOH 39

Aida telah menyimpan wangnya sebanyak RM500 di dalam bank. Setiap tahun, Aida menerima kadar faedah sebanyak 10% ke atas baki dalam akaun. Berapakah baki wang simpanan Aida selepas lima tahun sekiranya dia tidak pernah mengeluarkan wang simpanannya di dalam bank? Anda mesti menggunakan kawalan ulangan *for* untuk menentukan jawapan.



Penyelesaian

```
public class Contoh39{  
    public static void main(String[] args){  
        double baki = 500.0;  
        for(int i=1;i<=5;i+=1){  
            baki = baki + (0.1 * baki);  
        }  
        System.out.println("Baki 5 tahun: " + baki);  
    }  
}
```

Contoh output:

```
Command Prompt  
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh39  
Baki 5 tahun: 805.255
```



21

Struktur Kawalan Ulangan *For*

Kerja Berkumpulan

Lakukan aktiviti yang berikut.

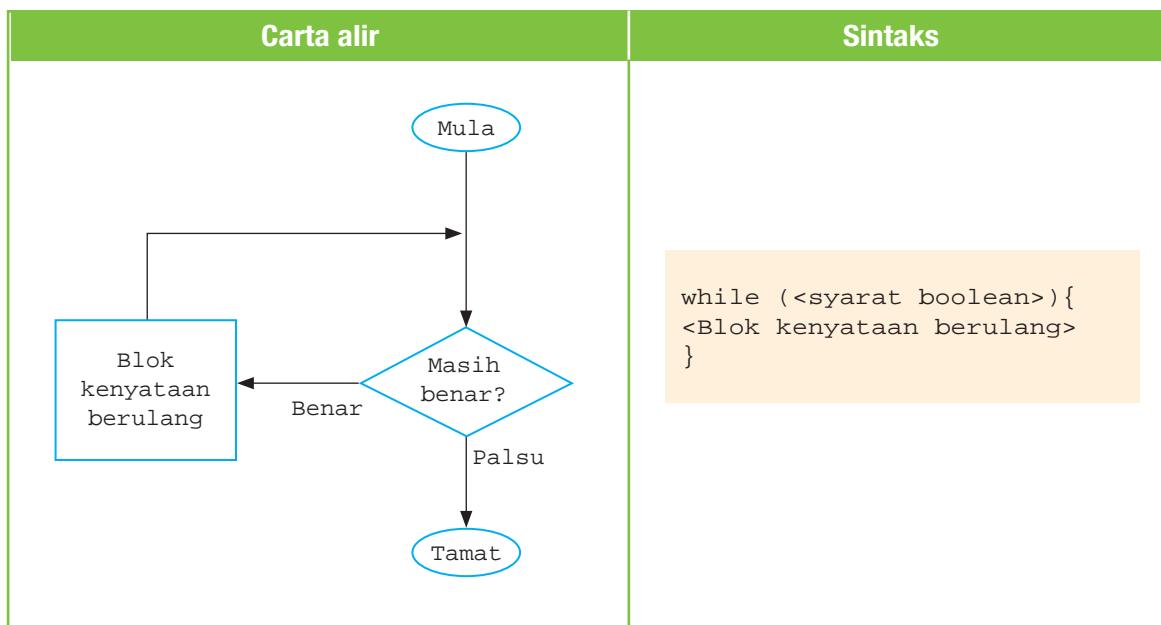
- 1 Pilih seorang murid. Murid diminta berdiri di satu penjuru papan putih. Berikan sebatang pen "marker".
- 2 Guru akan berdiri di antara murid dan papan putih. Umumkan kepada kelas bahawa guru ingin melihat murid mengulang untuk menulis satu perkataan yang sama sebanyak enam kali. Minta murid-murid kelas membilang ulangan yang dibuat.
- 3 Di satu penjuru yang lain, letakkan sebuah kerusi.



- 4 Murid diminta berjalan ke papan putih, menulis "Hello dunia!", berjalan ke arah kerusi dan kemudian berpatah balik ke arah guru.
- 5 Murid akan bertanya kepada guru jika dia dibenarkan berhenti. Guru akan bertanya kepada kelas berapa ulangan yang sudah dilakukan murid tersebut.
- 6 Jika murid lain tidak membenarkan untuk berhenti, guru akan mengarahkan murid tersebut mengulang setiap langkah sebelumnya semula.
- 7 Apabila kelas bersetuju bahawa murid ini sudah mencapai bilangan kali yang dikehendaki, benarkan murid balik ke tempat duduk dan permainan ditamatkan. Persoalannya:
 - (a) Berapakah bilangan minimum ulangan yang dapat dibuat?
 - (b) Berapakah bilangan maksimum ulangan yang dapat dibuat?
 - (c) Apakah syarat yang membenarkan pengulangan?
 - (d) Dalam keadaan apakah murid tidak dibenarkan mengulang dan permainan ditamatkan?
 - (e) Adakah bilangan ulangan perlu diketahui terlebih dahulu?

Kawalan Ulangan While

Kawalan ulangan *while* membuat ujian terlebih dahulu ke atas input. Jika input memenuhi syarat *boolean*, blok arahan dalam gelung akan dilaksanakan. Jika syarat tidak dipenuhi, kawalan ulangan akan berhenti dan kawalan akan bergerak ke baris arahan selepas kawalan ulangan. Rajah 1.36 menunjukkan kawalan ulangan *while*.



Rajah 1.36 Carta alir dan sintaks untuk kawalan ulangan *while*



While loop in Java with examples
goo.gl/qCyMB0

Kawalan ulangan mengandungi kawalan pilihan tetapi cuba perhatikan arah anak panah untuk keputusan "Benar". Adakah anak panah berpusing kembali kepada anak panah yang memasuki rombus kawalan pilihan? Tujuan berbuat demikian adalah supaya syarat *boolean* yang sama akan diuji berkali-kali.

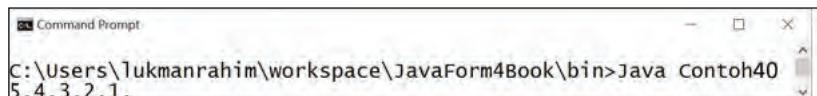
Setelah penyataan-penyataan dalam blok berulang dilaksanakan, kenal pasti sama ada atur cara perlu mengemas kini nilai dalam syarat. Hal ini kerana, jika langkah ini tidak dibuat, maka syarat *boolean* yang diuji akan mempunyai nilai *boolean* "Benar" selama-lamanya.

CONTOH (40)

Tentukan output bagi kod Java berikut:

```
public class Contoh40{
    public static void main(String[] args){
        int n=5;
        while(n>0){
            System.out.print(n + " , ");
            n-=1;
        }
    }
}
```

Contoh output:



```
Command Prompt
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>Java Contoh40
5,4,3,2,1.
```

CONTOH (41)

Tahukah Anda?

Langkah perlaksanaan kawalan *while*:

- 1 <ungkapan> dinilai
- 2 Jika <ungkapan> bernilai benar, <penyataan> dilaksanakan.
- 3 Jika <ungkapan> bernilai palsu, tamat gelung.
- 4 Ulang langkah 3 dan 4

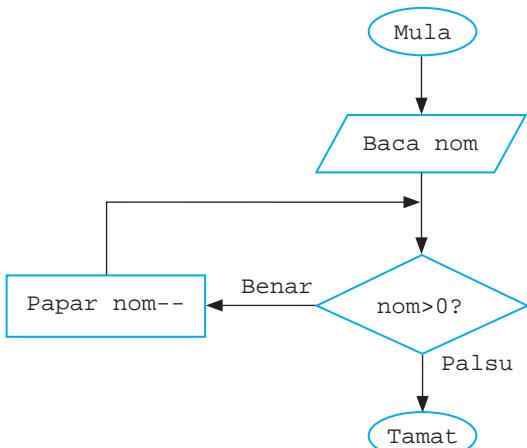
Gunakan kawalan ulangan *while* untuk mempamerkan semua nombor dalam turutan menurun daripada integer positif nombor yang dimasukkan pengguna sehingga sifar.

- (a) Tulis pseudokod untuk penyelesaian bagi masalah di atas.
- (b) Lukiskan carta alir untuk penyelesaian masalah di atas.
- (c) Tulis atur cara Java menggunakan carta alir yang telah dilukis sebagai rujukan.

Penyelesaian:

- (a) Baca input nom
Selagi (nom > 0) mula_selagi
 Papar nom=nom-1;
 tamat_selagi

(b)



Berdasarkan Contoh 42, gunakan kawalan ulangan *while* untuk memperkenan semua nombor dalam turutan menurun daripada integer negatif nombor yang dimasukkan pengguna sehingga -10.

(c)

```
public class Module1{
    public static void main(String args[]){
        Scanner scanner=new Scanner(System.in);
        int nom=scanner.nextInt();
        while (nom>0){
            nom -= 1;
            System.out.println(nom + " ");
        }
    }
}
```

CONTOH 42

Tuliskan sebuah aratur cara yang memaparkan arahan kepada pengguna untuk memasukkan kata laluan. Kata laluan pengguna mestilah sama dengan nilai dalam rekod, iaitu nilai dalam pemboleh ubah RekodLaluanRahsia. Aratur cara akan memaparkan arahan kepada pengguna sekali lagi jika ujian kata laluan gagal.

Penyelesaian

```
public class Contoh42{
    public static void main(String[] args){
        String strPassword = new String();
        Scanner scanner=new Scanner(System.in);
        final String RekodLaluanRahsia="Pisang";

        while(!strPassword.equals(RekodLaluanRahsia)){
            System.out.print("Sila masukkan password: ");
            strPassword=scanner.next();
            System.out.println();
        }
    }
}
```

Contoh output:

C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh42
Sila masukkan password:Pisang



Lakukan aktiviti yang berikut.

- 1 Pilih seorang murid. Murid diminta berdiri di satu penjuru papan hitam. Berikan sebatang kapur putih.
- 2 Guru akan berdiri di antara murid dan papan hitam. Fikirkan syarat baharu dan umumkan kepada kelas. Sebagai contoh, umumkan bahawa guru ingin melihat ayat "Hello dunia!" 6 kali di atas papan hitam.
- 3 Di satu penjuru yang lain, letakkan sebuah kerusi.
- 4 Murid akan meminta kebenaran untuk menulis "Hello dunia!" daripada guru. Guru boleh bertanya kelas adakah ayat "Hello dunia!" belum mencukupi.
- 5 Jika kelas berkata ya, murid tadi dibenarkan berjalan ke papan hitam, menulis "Hello dunia!", dan kemudian berjalan mengelilingi kerusi dan kemudian berpatah balik ke penjuru asal.
- 6 Guru akan menyemak syarat sekali lagi. Jika penyemakan adalah benar, murid diminta mengulangi langkah-langkah sebelumnya.
- 7 Sekiranya tidak mematuhi syarat, murid diminta balik ke tempat duduk dan permainan ditamatkan.
- 8 Minta murid-murid memikirkan perkara berikut dan membuat kesimpulan:
 - (a) Bilangan minimum ulangan yang boleh dibuat. Adakah mungkin tiada langsung?
 - (b) Bilangan maksimum ulangan yang boleh dibuat.
 - (c) Apakah syarat yang membenarkan pengulangan?
 - (d) Dalam keadaan apakah murid tidak dibenarkan mengulang dan permainan ditamatkan?
 - (e) Adakah bilangan ulangan perlu diketahui terlebih dahulu?

Tahukah Anda ?

Gelang Infiniti/Gelung Tanpa Henti

Adakah terdapat sebarang kemungkinan bahawa blok kenyataan dalam gelung akan berulang tanpa henti? Selagi syarat *boolean* yang diuji menghasilkan keputusan ya, maka gelung akan berulang lagi. Oleh hal yang demikian, gelung tersebut dipanggil gelang infiniti. Gelang infiniti bukan perkara yang baik dan perlu dielakkan.

Pastikan syarat *boolean* yang diuji sentiasa dikemas kini dalam blok kenyataan yang diulang supaya ujian syarat *boolean* akan menghasilkan keputusan tidak benar kelak.

Contoh gelang infiniti:

Carta alir	Sintaks
<pre> graph TD Benar{Benar} --> Blok[Blok kenyataan berulang] Blok --> Benar </pre>	<pre> while(benar) System.out.println("Gelung enggan berhenti"); </pre>

Namun begitu, kata kunci *break* dan kawal pilihan *if-then* boleh digunakan untuk menghentikan gelang infiniti.



Kawalan Ulangan Do-While

Kawalan ulangan *do-while* menyerupai kawalan ulangan *while*. Keadaan yang membezakan kawalan ulangan *do-while* ialah kawalan ini membuat ujian selepas blok arahan dalam gelung dilaksanakan. Oleh itu, kawalan ulangan *do-while* memberikan jaminan bahawa blok arahan akan dilaksanakan walaupun sekali.

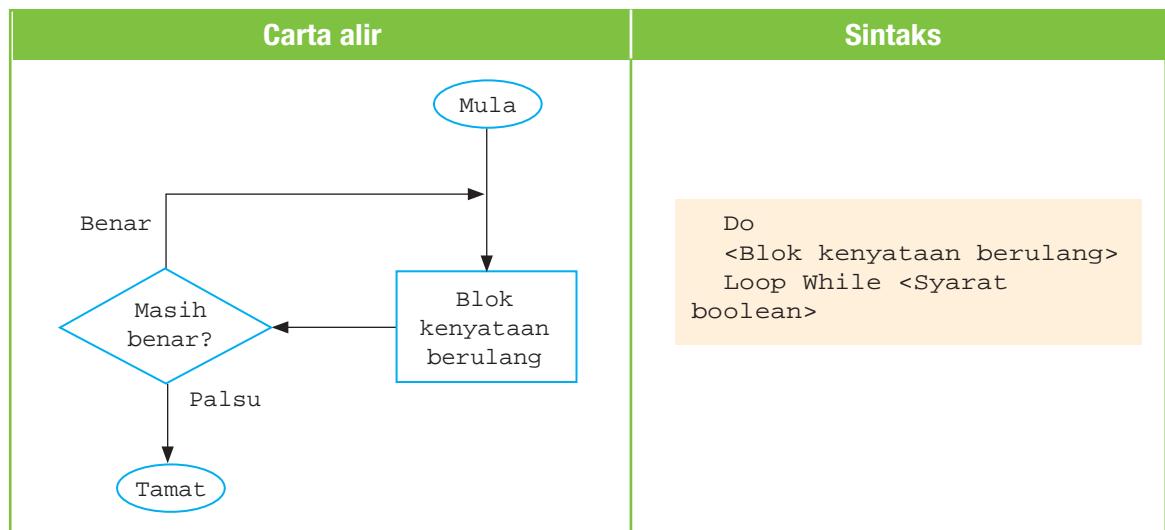
Perhatikan Rajah 1.37. Susunan blok arahan berulang terletak sebelum simbol ujian syarat *boolean*. Anak panah "Benar" akan diikuti sekiranya ujian syarat *boolean* menghasilkan keputusan benar. Anak panah "Benar" akan menghala kepada garis aliran sebelum blok arahan berulang. Sebaliknya, dalam kawalan ulangan *while*, anak panah "Benar" daripada simbol ujian syarat *boolean* akan menghala kepada blok arahan ulangan sebelum berbalik kepada simbol ujian syarat *boolean*.

Jika "Palsu", kawalan ulangan akan berhenti dan kawalan akan bergerak ke baris arahan yang baharu.

Tahukah Anda?

Langkah pelaksanaan kawalan *do while*:

- 1 <pernyataan> dilaksanakan
- 2 <ungkapan> dinilai
- 3 Jika <ungkapan> bernilai benar, ulang langkah 1 hingga 2
- 4 Jika <ungkapan> bernilai palsu, tamat gelung



Rajah 1.37 Carta alir dan sintaks untuk kawalan ulangan *do-while*

Contoh (43)

Apakah output untuk kawalan ulangan yang berikut?

```
int no = 1;
do{
    System.out.println(no + " x 3 = " + no * 3);
    no = no + 1;
} while(no <= 12);
```

Tahukah Anda?

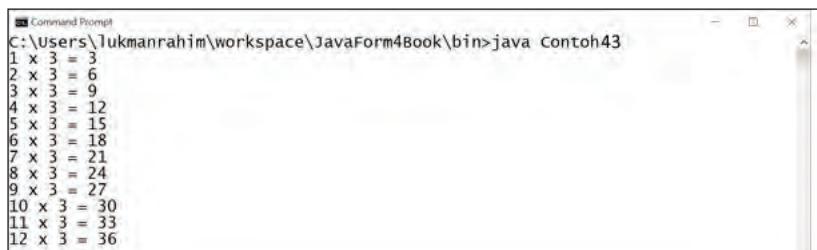
Ringkasan untuk pernyataan kawalan *for*, *while* dan *do while*.

- Bentuk *for*
`for (<awalan>; <ujian>;
 <pengemaskinian>;
 <pernyataan>;)`
- Bentuk *while*
`<awalan>;
while (<ujian>){
 <pernyataan>;
 <pengemaskinian>;
}`
- Bentuk *do while*
`<awalan>;
Do {
 <pernyataan>;
 <pengemaskinian>;
} while (<ujian>);`

Penyelesaian

```
public class Contoh43{  
    public static void main(String[] args){  
        int no = 1;  
        do{  
            System.out.println(no + " x 3 = " + no  
                * 3);  
            no = no + 1;  
        }while(no <= 12);  
    }  
}
```

Contoh output:



```
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java contoh43  
1 x 3 = 3  
2 x 3 = 6  
3 x 3 = 9  
4 x 3 = 12  
5 x 3 = 15  
6 x 3 = 18  
7 x 3 = 21  
8 x 3 = 24  
9 x 3 = 27  
10 x 3 = 30  
11 x 3 = 33  
12 x 3 = 36
```

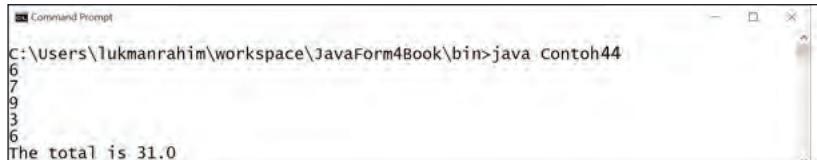
CONTOH (44)

Tulis gelung kawalan ulangan *do-while* yang akan meminta nombor integer daripada pengguna sebanyak 5 kali sebelum mempamerkan jumlah nombor yang telah dimasukkan.

Penyelesaian

```
public class Contoh44{  
    public static void main(String[] args){  
        double no;  
        double sum;  
        int counter = 1;  
        Scanner scanner=new Scanner(System.in);  
        do{  
            no = scanner.nextDouble();  
            sum += no;  
            counter = counter + 1;  
        }while(counter <= 5);  
        System.out.println("The total is " + sum);  
    }  
}
```

Contoh output:



```
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java contoh44  
6  
7  
9  
3  
6  
The total is 31.0
```



Contoh 45

Sebuah aplikasi menerima dua nombor integer daripada pengguna dan memaparkan perbezaan dua nombor tersebut. Setelah itu, aplikasi akan bertanya kepada pengguna jika ingin mengulanginya. Pengguna memasukkan perkataan "Ya" jika ingin mengulangi, atau lain-lain jika tidak mahu mengulanginya. Tuliskan atur cara Java untuk aplikasi ini.

Penyelesaian

```
public class Contoh45{  
    public static void main(String args[]){  
        String input;  
        Scanner scanner=new Scanner(System.in);  
        int no1;  
        int no2;  
        do {  
            no1 = scanner.nextInt();  
            no2 = scanner.nextInt();  
            System.out.println("Beza: " + Math.  
                abs(no2 - no1));  
            System.out.println("Taip Ya untuk  
                teruskan ..");  
            input=scanner.next();  
        }while(input.equals("Ya"));  
    }  
}
```

Contoh output:

```
Command Prompt  
C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh45  
5  
7  
Beza: 2  
Taip Ya untuk teruskan ..  
Ya  
5  
9  
Beza: 4  
Taip Ya untuk teruskan ..  
t
```



Info tambahan:
Penyataan *continue* merupakan satu penyataan kawalan yang membolehkan untuk menyambung terus kepada gelung seterusnya tanpa perlu melaksanakan penyataan lain di dalam gelung. Penyataan ini boleh digunakan dalam gelung *for*, *while* dan *do while*.
Penyataan *break* pula digunakan untuk keluar daripada gelung tersebut. Penyataan *finally* digunakan bagi memastikan gelung dilaksanakan walaupun pengecualian tidak dijangkakan.



23

Struktur Kawalan Ulangan Do-While

Kerja Berkumpulan

- 1 Tulis 10 angka dari 0 hingga 9 di atas papan hitam.
- 2 Pilih seorang murid. Murid diminta berdiri di satu penjuru papan hitam. Berikan sebatang kapur putih.
- 3 Di satu penjuru yang lain, letakkan sebuah kerusi. Guru berada di sebelah kerusi. Guru akan menulis satu operator perbandingan (lebih kecil, lebih besar atau sama dengan) diikuti satu angka di antara 3 hingga 8 di atas sekeping kertas A4. Sebagai contoh, < 5. Kertas ini ialah syarat rahsia yang dirahsiakan daripada murid tadi.
- 4 Murid diminta berjalan ke papan hitam, membulatkan satu angka, dan kemudian berjalan ke arah guru.
- 5 Sebelum dibenarkan berjalan mengelilingi kerusi, guru akan bertanya kepada kelas sama ada pilihan

- murid mematuhi syarat rahsia. Seluruh kelas boleh melihat syarat itu kecuali murid tadi. Jika ya, guru akan membenarkan murid berjalan mengelilingi kerusi dan kemudian berpatah balik ke penjuru asal.
- 6 Murid diminta mengulangi langkah-langkah sebelumnya dengan memilih satu nombor baharu.
 - 7 Sekiranya pilihan tidak mematuhi syarat, murid diminta balik ke tempat duduk dan permainan ditamatkan.
 - 8 Minta murid-murid memikirkan perkara berikut dan membuat kesimpulan:
 - (a) Bilangan minimum ulangan yang boleh dibuat.
 - (b) Bilangan maksimum ulangan yang boleh dibuat.
 - (c) Apakah syarat yang membenarkan pengulangan?
 - (d) Dalam keadaan apakah murid tidak dibenarkan mengulang dan permainan ditamatkan?
 - (e) Adakah bilangan ulangan perlu diketahui terlebih dahulu?

Tahukah Anda ?

Semuanya membawa maksud yang sama!

- $i = i + 1$
- $i += 1$
- $i++$

1.4.4 Struktur Kawalan Ulangan yang Melibatkan Operator Increment dan Decrement, Class Math dan Boolean

Operator Increment (++) dan Decrement (--)

Operator *increment*(++) dan operator *decrement*(--) lazimnya digunakan dalam kawalan ulangan sebagai pembilang. Operator *increment* (++) ialah penambahan nilai pemboleh ubah bagi bilangan nombor tertentu. Sebaliknya, operator *decrement* (--) mengurangkan nilai pemboleh ubah bagi bilangan nombor tertentu.

Jadual 1.13 Ungkapan tokok

Ungkapan	Makna	Contoh
$i+=1$	$i = i + 1$	<pre>while (<syarat boolean>){ <Blok kenyataan berulang> <kemas kini nilai dalam syarat> }</pre> <p>Katakan i mengandungi 9. $i += 1$ akan menambah 1 kepada 9. Jadi, nilai baharu i ialah 10.</p>
$i+=2$	$i = i + 2$	<pre>while (<syarat boolean>){ <Blok kenyataan berulang> <kemas kini nilai dalam syarat> }</pre> <p>Katakan i mengandungi 5. $i += 2$ akan menambah 2 kepada 5. Jadi, nilai baharu i ialah 7.</p>





Ungkapan	Makna	Contoh
i+=3	i = i + 3	<pre>while (<syarat boolean>){ <Blok kenyataan berulang> <kemas kini nilai dalam syarat> }</pre> <p>Katakan i mengandungi 2. $i += 3$ akan menambah 3 kepada 2. Jadi, nilai baharu i ialah 5.</p>

Tahukah Anda?

++i;
--i;

Tokok dan susut berlaku sebelum nilai boleh ubah digunakan lebihan ungkapan.

i++;
i--;

Tokok dan susut berlaku selepas nilai pemboleh ubah digunakan dalam lebihan ungkapan.

Ungkapan	Makna	Contoh
i-=1	i = i - 1	<pre>while (<syarat boolean>){ <Blok kenyataan berulang> <kemas kini nilai dalam syarat> }</pre> <p>Katakan i mengandungi 9. $i -= 1$ akan berkurang 1 daripada 9. Jadi, nilai baharu i ialah 8.</p>
i-=2	i = i - 2	<pre>while (<syarat boolean>){ <Blok kenyataan berulang> <kemas kini nilai dalam syarat> }</pre> <p>Katakan i mengandungi 5. $i -= 2$ akan berkurang 2 daripada 5. Jadi, nilai baharu i ialah 3.</p>
i-=3	i = i - 3	<pre>while (<syarat boolean>){ <Blok kenyataan berulang> <kemas kini nilai dalam syarat> })</pre> <p>Katakan i mengandungi 2. $i -= 3$ akan berkurang 3 daripada 2. Jadi, nilai baharu i ialah -1.</p>

Math.random()

Math.random() ialah subatur cara Java untuk menjana nombor secara rambang. *Math.random()* menggunakan waktu sistem sebagai nilai benih untuk memulakan penjanaan nombor rambang. Subatur cara ini digunakan setiap kali diperlukan untuk menjana suatu nombor rambang antara 0 dan 1.

Sekiranya nombor rambang diperlukan adalah di antara 1 hingga 10, gunakan ungkapan $(int)(Math.random() * 10) + 1$. Sekiranya nombor rambang diperlukan adalah di antara 1 hingga 100, maka ungkapan $(int)(Math.random() * 100) + 1$ digunakan.

Bagaimana pula jika nombor rambang diperlukan dalam julat 1 hingga 6?



Increment and decrement
goo.gl/RZqgGH



CONTOH (46)

Tulis kod Java untuk menjana 20 nombor rambang bagi dadu.
Dadu cuma mempunyai enam kombinasi, iaitu 1, 2, 3, 4, 5 dan 6.



Penyelesaian

```
public class Contoh46{
    public static void main(String[] args){
        int i;
        Scanner scanner=new Scanner(System.in);
        boolean flag=true;
        do{
            for (i = 1;i<=20;i++){
                System.out.print((int)((Math.random() * 6) + 1) + " ");
            }
            System.out.println();
            System.out.println("-----");
            System.out.println("Taip ya untuk teruskan. Tidak untuk
henti");
            if(scanner.next().equals("ya")){
                flag=true;
            }else{
                flag=false;
            }
        }while(flag);
    }
}
```

Bendera Boolean

Bagaimana jika atur cara membenarkan pengguna menentukan sama ada ingin meneruskan satu lagi set 20 nombor rambang? Dalam hal ini, kod di atas boleh diletakkan dalam kawalan ulangan *do-while* dan satu pemboleh ubah *Boolean* digunakan untuk mengawal ulangan.

Pemboleh ubah ini beroperasi sebagai *flag*, atau bendera penentu. Sebelum memasuki gelung seterusnya, pengguna ditanya sama ada ingin meneruskan atur cara itu lagi. Jika "Benar", *flag* diset sebagai *True*. Jika sebaliknya, *flag* diset sebagai *False*.

A screenshot of a Windows Command Prompt window titled "Command Prompt". The command entered is "C:\Users\lukmanrahim\workspace\JavaForm4Book\bin>java Contoh46". The output shows 20 random numbers between 1 and 6 generated by the program, followed by a prompt asking if the user wants to continue. The user types "ya" (yes) twice, causing the loop to repeat, and then types "Tidak" (no), which exits the loop.

Rajah 1.38 Output 20 nombor dengan opsyen untuk mengulang

Contoh 47

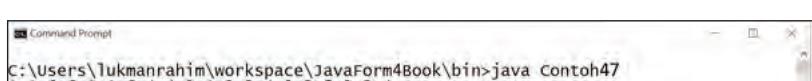
Tuliskan kod Java yang menjana 20 nombor dadu secara rambang dan pengguna diberikan pilihan untuk mengulang proses itu.

Penyelesaian

```
public class Contoh47{
    public static void main(String[] args){
        int i;
        int diceNo;
        int count1=0,count2=0,count3=0,count4=0,count5=0,count6=0;
        Scanner scanner=new Scanner(System.in);
        boolean flag=true;
        do{
            for (i = 1;i<=20;i++){
                diceNo= (int)(Math.random() * 6 + 1);
                System.out.print(diceNo+ " ");
                switch(diceNo){
                    case 1: count1++; break;
                    case 2: count2++; break;
                    case 3: count3++; break;
                    case 4: count4++; break;
                    case 5: count5++; break;
                    case 6: count6++;
                }
            }
            System.out.println();
            System.out.println("* Dice number 1 = " + count1 + " %");
            System.out.println("* Dice number 2 = " + count2 + " %");
            System.out.println("* Dice number 3 = " + count3 + " %");
            System.out.println("* Dice number 4 = " + count4 + " %");
            System.out.println("* Dice number 5 = " + count5 + " %");
            System.out.println("* Dice number 6 = " + count6 + " %");
            System.out.println();
            System.out.println("-----");
            System.out.println("Taip ya untuk teruskan. Tidak untuk henti");

            if(scanner.next().equals("ya")){
                flag=true;
            }else{
                flag=false;
            }
        }while(flag);
    }
}
```

Contoh output:



Pembilang

Selain pembilang i yang biasa digunakan dalam kawalan ulangan *for*, pembilang lain boleh diisyiharkan untuk membuat pengiraan dalam penyelesaian masalah. Pembilang ini diisyiharkan oleh pengatur cara di luar kawalan ulangan sebagai pemboleh ubah jenis int dan diumpukkan nilai asal 0. Dalam badan blok kawalan ulangan, pembilang ini akan ditokok atau dikurangkan bergantung kepada jenis permasalahan menggunakan operator *Increment* atau *Decrement*. Syarat yang melibatkan perubahan nilai pembilang diuji menggunakan struktur kawalan pilihan seperti *switch-case* atau *if-else*.

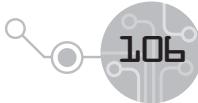
Untuk memahami peranan pembilang dengan lebih lanjut lagi, lihat contoh yang berikut.

CONTOH (48)

Eksperimen menentukan kekerapan nombor dadu 1, 2, 3, 4, 5 dan 6 hasil 100 kali lambungan rambang agak sukar dibuat secara manual. Katakan hasil eksperimen adalah peratusan kekerapan setiap nombor dadu dalam 100 lambungan.

Penyelesaian

```
1 package dadurambang;
2 import java.util.Scanner;
3
4 public class DaduRambang {
5     public static void main(String[] args) {
6         int i;
7         int diceNo;
8         int kiral=0,kira2=0,kira3=0,kira4=0,kira5=0,kira6=0;
9         double persen1=0,persen2=0,persen3=0,persen4=0,persen5=0,persen6=0;
10        int jumlahKiraan=0;
11        Scanner scanner=new Scanner(System.in);
12        boolean flag=true;
13        do{
14            for (i=1; i<=100; i++){
15                diceNo= (int)(Math.random() * 6 + 1);
16
17                switch(diceNo){
18                    case 1: kiral++; break;
19                    case 2: kira2++; break;
20                    case 3: kira3++; break;
21                    case 4: kira4++; break;
22                    case 5: kira5++; break;
23                    case 6: kira6++;
24                }
25            }
26
27            jumlahKiraan = kiral + kira2 + kira3 + kira4 + kira5 + kira6;
28
29            persen1 = (double)kiral/jumlahKiraan * 100;
30            persen2 = (double)kira2/jumlahKiraan * 100;
31            persen3 = (double)kira3/jumlahKiraan * 100;
32            persen4 = (double)kira4/jumlahKiraan * 100;
33            persen5 = (double)kira5/jumlahKiraan * 100;
34            persen6 = (double)kira6/jumlahKiraan * 100;
35
36            System.out.println();
37            System.out.println("*Nombor 1 dadu = " + persen1 + "%");
38            System.out.println("*Nombor 2 dadu = " + persen2 + "%");
39            System.out.println("*Nombor 3 dadu = " + persen3 + "%");
40            System.out.println("*Nombor 4 dadu = " + persen4 + "%");
41            System.out.println("*Nombor 5 dadu = " + persen5 + "%");
42            System.out.println("*Nombor 6 dadu = " + persen6 + "%");
43            System.out.println();
44            System.out.println("-----");
45            System.out.println("Taip ya untuk teruskan:");
46            if(!scanner.next().equals("ya")){
47                flag=false;
48            }
49        }while(flag);
50    }
51 }
```



Contoh output:

```
C:\WINDOWS\system32\cmd.exe -java -jar DaduRambang.jar  
D:\Java Projects\DaduRambang\dist>java -jar DaduRambang.jar  
*Nombor 1 dadu = 16.0 %  
*Nombor 2 dadu = 13.0 %  
*Nombor 3 dadu = 13.0 %  
*Nombor 4 dadu = 15.0 %  
*Nombor 5 dadu = 20.0 %  
*Nombor 6 dadu = 23.0 %  
  
-----  
Taip ya untuk teruskan:  
ya
```

Latihan Formatif 1.4

- 1 Indeks Jisim Badan atau lebih dikenal sebagai BMI (*Body Mass Index*) ialah suatu ukuran yang menunjukkan hubungan antara berat badan dan ketinggian seseorang. BMI boleh diukur dengan mengambil kira berat badan seseorang (dalam unit kilogram) dan dibahagi dengan dua kali ketinggian seseorang (dalam unit meter).

$$\text{BMI} = \frac{\text{berat (kg)}}{\text{tinggi (m)} \times \text{tinggi (m)}}$$

Setelah mendapat keputusan BMI, anda bolehlah melihat carta dalam jadual yang ditunjukkan di bawah bagi menentukan kategori anda.

BMI	Status
Kurang daripada 20.6	Kurus
20.7 – 26.4	Normal
26.5 – 30.9	Gemuk
31.0 – 45.2	Obesiti
Lebih daripada 45.3	Bahaya

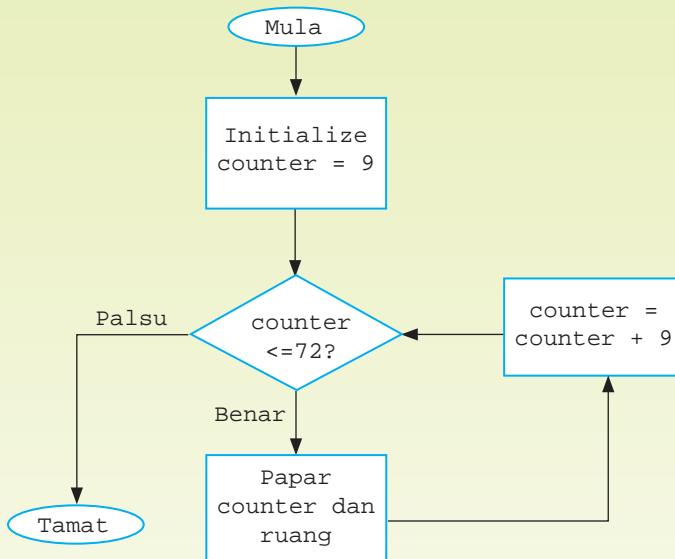
Tuliskan atur cara Java yang dapat,

- menerima nilai berat seseorang (dalam kg) dan nilai tinggi seseorang (dalam m).
- mengira BMI berdasarkan nilai berat dan tinggi yang diperolehi pada soalan (a).
- memaparkan status seperti dalam jadual di atas berdasarkan nilai BMI yang diperoleh.

- 2 Tentukan output untuk kawalan ulangan yang berikut.

```
(a) for(n = 2;n<=20; n+=3){  
    System.out.println(n + " ");  
}  
(b) for(n = 150;n<=40;n-=15){  
    System.out.println(n + " ");  
}
```

- 3 Tulis atur cara Java yang meminta pengguna memasukkan dua integer, iaitu noMula dan noTamat. Gunakan struktur kawalan ulangan *for* untuk memperlihatkan semua integer genap di antaranya dan kedua-dua nombor tersebut.
- 4 Faktorial ialah hasil pendaraban nombor dengan nombor berikutnya yang lebih kecil sehingga nombor 1. Faktorial 5 ialah $5 \times 4 \times 3 \times 2 \times 1 = 120$. Gunakan kawalan ulangan *do-while* untuk mencari faktorial bagi integer positif yang dimasukkan oleh pengguna.
- 5 Kaji carta alir yang berikut.



- (a) Tulis atur cara menggunakan bahasa pengatur cara Java berdasarkan carta alir di atas. Tunjukkan bahagian kawalan sahaja.
(b) Apakah output yang anda jangka?
- 6 Tulis sebuah atur cara yang memaparkan arahan kepada pengguna untuk memasukkan kata laluan. Kata laluan pengguna mestilah sama dengan nilai dalam rekod, iaitu nilai dalam boleh ubah RekodLaluanRahsia. Atur cara akan memaparkan arahan sekali lagi jika ujian kata laluan gagal. Pengguna hanya boleh mencuba sebanyak tiga kali sebelum aplikasi ditutup sendiri.
- (a) Tulis pseudokod untuk penyelesaian masalah ini.
(b) Lukis carta alir.
(c) Tulis atur cara Java berdasarkan algoritma yang anda reka sebelum ini.

- 7 Tulis pernyataan *if...else...if* berdasarkan jadual yang ditunjukkan di bawah.

Kod ujian	Paparan
“KERANI”	“Capaian data sahaja”
“JURUTEKNIK”	“Capaian sistem sahaja”
“PENTADBIR”	“Capaian sistem dan data”
“PENGURUS”	“Capaian keseluruhan”
Lain-lain	“Capaian tidak diterima”

- 8 Tulis semula atur cara di bawah dengan menggunakan arahan *while*.

```
int p, i;
for (i = 0; i<=9; i++)
{
    p = i + 1;
    System.out.println (i + " - " + p);
}
if (i>=9)
    System.out.println ("tamat\n");
```



- 9 Tulis satu atur cara yang dapat membaca data Kod buku dan Bilangan buku seperti yang ditunjukkan dalam rajah yang ditunjukkan di bawah. Paparkan juga kod yang mempunyai bilangan buku yang tertinggi dan terendah.

Kod buku	Bilangan buku
100	12
101	5
111	2
113	16
145	25

- 10 Tulis semula pernyataan *if* berikut menggunakan pernyataan *switch...case*.



```
(a)      int kiraan = 29;
if (kiraan % 10 == 0)
    System.out.println ("Hasil kiraan = " + kiraan);
else
    System.out.println("Hasil kiraan = " + null);
```

```
(b) int poin = 3, markah;
    if (poin == 1)
        markah += 5;
    else if (poin == 2)
        markah += 10;
    else if (poin == 3)
        markah += 26;
    else if (poin == 4)
```

-  11 Tulis atur cara Java yang dapat menerima nombor di antara 1 dan 100 secara rawak menggunakan subatur cara `Math.random()`. Contoh output ditunjukkan seperti di bawah:

```
Nombor rawak di antara 1 dan 100 adalah:
Nombor Rawak [1] : 23
Nombor Rawak [2] : 56
Nombor Rawak [3] : 27
Nombor Rawak [4] : 96
Nombor Rawak [5] : 90
```

-  12 Tulis atur cara Java yang dapat menentukan sama ada seseorang itu layak untuk mengundi atau tidak berdasarkan umur dan status pendaftaran (telah membuat pendaftaran atau tidak pernah membuat pendaftaran) dan memaparkan mesej seperti di bawah:

Umur	Status	Mesej Kelayakan
Bawah 18 tahun	Mendaftar	"Anda terlalu muda dan tidak layak mengundi"
	Tidak Mendaftar	
Sekurang-kurangnya 18 tahun	Mendaftar	"Anda layak mengundi"
Sekurang-kurangnya 18 tahun	Tidak Mendaftar	"Anda perlu mendaftar dahulu sebelum mengundi"

-  13 Pengisyntihsaran pemboleh ubah diberikan seperti di bawah:

```
boolean pass = true;
```

Tulis atur cara yang dapat menentukan sama ada transaksi anda boleh diteruskan atau tidak menggunakan pernyataan `if...else...if` :-

Nilai	Mesej transaksi
Benar	"Anda boleh teruskan transaksi berikutnya"
Palsu	"Anda perlu ulang semula kemasukan kata laluan"



STANDARD PEMBELAJARAN

- 1.5.1 Membezakan jenis ralat dalam atur cara (sintaks, masa larian dan logik)
- 1.5.2 Mengesan, mengenal pasti, menterjemah mesej ralat dan membaiki ralat
- 1.5.3 Mengenal pasti nilai bagi pemboleh ubah pada bahagian tertentu atur cara
- 1.5.4 Menghasilkan atur cara yang mudah dibaca dengan menggunakan gaya yang baik (komen, nama pemboleh ubah yang bermakna, inden)

Amalan Terbaik Pengaturcaraan

Amalan terbaik secara umum ialah teknik atau metodologi yang telah dibuktikan melalui suatu pengalaman atau kajian yang boleh dipercayai, untuk mendapatkan hasil yang diinginkan. Amalan terbaik dalam sesuatu bidang merupakan satu komitmen untuk menggunakan semua pengetahuan dan teknologi yang ada untuk memastikan keberhasilan yang baik.

Amalan terbaik pengaturcaraan ialah apabila pengatur cara dapat mempraktikkan amalan-amalan yang biasa diikuti untuk menghasilkan atur cara yang baik. Rajah di bawah menunjukkan perbezaan yang boleh dilihat antara kod atur cara yang dibangunkan menggunakan amalan terbaik pengaturcaraan dan kod atur cara yang dibangunkan dengan tidak menerapkan amalan terbaik pengaturcaraan.

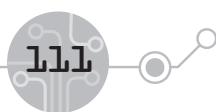
Rajah yang manakah membuatkan anda lebih memahami apa yang ingin disampaikan oleh pengatur cara?

```
/* Langkah 1: Baca nombor1 dan nombor2
// Langkah 2: Hitung hasil tolak nombor1 - nombor2
// Langkah 3: Papar hasil*/
// Program Mengira Hasil Tolak Dua Nombor
public class Tolak_Dua_Nombor {
    public static void main(String[] args){
        //Pengisytiharan input
        int nombor1 = 20;
        int nombor2 = 13;
        //Pengisytiharan output
        double hasil;

        //Proses menolak nombor1 - nombor2
        hasil = nombor1 - nombor2;

        //Output yang akan dipaparkan
        System.out.println("Hasil tolak ialah " +
        hasil );
    }
}
```

Rajah 1.39 Atur cara yang menggunakan amalan terbaik pengaturcaraan





Aplikasi Harian



Penggunaan Java
Netbeans dalam
Pembinaan Permainan
Telefon Pintar
<https://goo.gl/uwbGeQ>

```
public class Tolak_Dua_Nombor {
    public static void main(String[] args){
        int nombor1 = 20;
        int nombor2 = 13;
        double hasil;

        hasil = nombor1 - nombor2;

        System.out.println("Hasil tolak ialah " + hasil );
    }
}
```

Rajah 1.40 Atur cara yang tidak menggunakan amalan terbaik pengaturcaraan

Berikut merupakan amalan-amalan terbaik pengaturcaraan yang perlu diikuti oleh seorang pengatur cara dalam menghasilkan suatu kod atur cara yang baik, atau boleh juga dikatakan faktor-faktor yang mempengaruhi kebolehbacaan kod atur cara.

1 Inden yang konsisten



- Penggunaan inden yang konsisten membuatkan kod atur cara mudah dibaca dan difahami oleh pengguna lain. Cara menulis inden yang konsisten perlu dilakukan dari awal kod atur cara hingga ke akhir kod.

2 Jenis data



- Pilih jenis data yang bersesuaian supaya saiz pemboleh ubah tidak terlampaui kecil atau besar dan memulihara sumber (*conserve resources*). Sebagai contoh, gunakan jenis data integer bagi nombor bulat dan bukan *double*.

3 Pemboleh ubah yang bermakna



- Skop biarlah kecil untuk mengelakkan kekeliruan dan mudah diselenggara iaitu:
 - (a) Tidak bermula dengan nombor. Contohnya, gunakan *cara1* dan bukan *1cara*.
 - (b) Tiada ruang kosong antara perkataan. Jika terdapat lebih daripada satu perkataan, gunakan *underscore* (*_*) atau rapatkan perkataan tersebut menjadi satu perkataan. Contohnya, *cara 1* atau *caraPertama* atau *cara_1*.
 - (c) Tidak sama dengan kata kekunci dalam Java. Contohnya, penggunaan *integer* atau *double* sebagai pemboleh ubah.
 - (d) Penggunaan huruf kecil dengan huruf besar seperti kata laluan, pemboleh ubah bersifat sensitif kes/huruf. Contohnya, *caraPertama* berbeza dengan *CaraPertama*.
- Nama yang bermakna dan mudah difahami. Penggunaan perkataan singkatan yang tidak jelas maksudnya atau penggunaan huruf seperti *x* dan *y* tidak digalakkkan.
- Tidak boleh terdiri daripada ruang kosong dan perkataaan rizab/khas seperti '*print*' dan '*value*'.

4 Komen



- Komen perlu ditulis dengan jelas dalam dua hingga tiga baris pendek untuk menerangkan fungsi kod dan memenuhi ruang lajur pengekodan.

Rajah 1.41 Amalan terbaik dalam pengaturcaraan

1.5.1 Jenis Ralat dalam Atur Cara

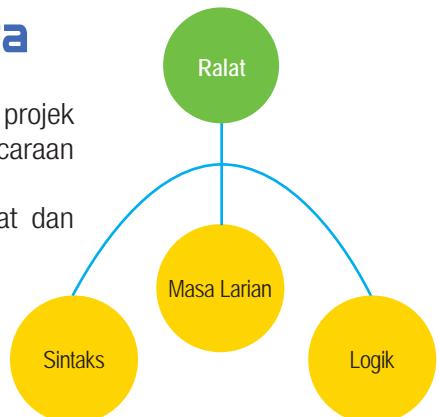
Ralat mungkin ditemui apabila anda menjalankan atur cara bagi projek pertama anda. Ralat yang sering berlaku semasa proses pengaturcaraan terbahagi kepada tiga jenis iaitu sintaks, masa larian dan logik.

Berikut akan diterangkan dengan lebih terperinci jenis-jenis ralat dan faktor penyebabnya.

Ralat Sintaks

Ralat sintaks merujuk ralat yang berlaku akibat perkara yang berikut:

- a) Kesalahan tatabahasa seperti salah ejaan dan tatatanda.
- b) Penggunaan objek atau aksara yang tidak dikenali.



Rajah 1.42 Jenis-jenis ralat dalam pengaturcaraan

Contoh 49

```
public class HelloMalaysia {  
    public static void main (String[] args){  
        System.out.println("Hello Malaysia!");  
    }  
}
```

Exception in thread "main" java.lang.Error: Unresolved compilation problem:
Syntax error, insert ")" to complete ClassBody
at HelloMalaysia.main(HelloMalaysia.java:9)

Contoh atur cara yang mengandungi ralat sintaks

Untuk rajah di atas, apakah yang menyebabkan atur cara tersebut mempunyai ralat sintaks? Cuba anda perhatikan dengan teliti kod atur cara tersebut. Fahami mesej ralat yang cuba disampaikan. Apabila anda perhatikan, satu tatatanda “ } ” telah hilang. Tatatanda “ } ” yang ada hanya untuk menutup satu tatatanda “ { ” pada “**public class** HelloMalaysia”. Manakala pada kaedah utama “**public static void** main (String[] args)” tidak diakhiri dengan tatatanda “}”. Rajah di bawah telah dilengkapkan dengan tatatanda “{”.

```
public class HelloMalaysia {  
    public static void main (String[] args){  
        System.out.println("Hello Malaysia!");  
    }  
}
```

Tatatanda telah dilengkapkan

Contoh atur cara mengandungi ralat sintaks tatatanda yang telah dibetulkan



Uji Minda



Cuba uji minda anda dengan menjawab kuiz pengaturcaraan.

CONTOH (50)

Rajah di bawah menunjukkan ralat sintaks. Semasa pengisytiharan pemboleh ubah atau pemalar dilakukan, jenis data juga perlu diisyiharkan bersama-sama nama pemboleh ubah tersebut. Cuba anda perhatikan atur cara di bawah. Pemboleh ubah "Tinggi" tidak dinyatakan bersama jenis datanya. Sepatutnya pemboleh ubah tinggi diisyiharkan sebagai "int Tinggi;".

```

public class LuasSegitiga {
    public static void main (String[] args){
        int Tapak = 6;
        Tinggi = 4; // Ralat Sintaks
        double Luas;

        Luas = (1.0 / 2) * Tapak * Tinggi;

        System.out.println ("Luas Segitiga ialah : "
        + Luas);
    }
}

```



Kerjaya Sains Komputer



Artikel ini memaparkan nasihat dan tips kerjaya daripada pakar dalam bidang pengaturcaraan.
<http://goo.gl/vrDCrk>

Exception in thread "main" java.lang.Error: Unresolved compilation problems:

Tinggi cannot be resolved to a variable
Tinggi cannot be resolved to a variable

at LuasSegitiga.main([LuasSegitiga.java:12](#))

```

public class LuasSegitiga {
    public static void main (String[] args){

        int Tapak = 6;
        int Tinggi = 4;
        double Luas;

        Luas = (1.0 / 2) * Tapak * Tinggi;

        System.out.println ("Luas Segitiga ialah : "
        + Luas);
    }
}

```

Ralat Masa Larian

Ralat masa larian ialah ralat yang ditemui ketika atur cara yang sedang berjalan terganggu akibat beberapa faktor. Ralat ini berlaku sekiranya pengatur cara cuba untuk melaksanakan operasi aritmetik yang mustahil. Contohnya seperti yang berikut:

- a Pengiraan data bukan berangka (*non-numerical*)
- b Pembahagian dengan digit 0
- c Mencari punca kuasa dua bagi nombor negatif

Contoh (51)

Pengiraan data bukan berangka (*non-numerical*)

Ah Chong sedang menulis atur cara untuk mengira luas sebuah segi tiga. Semasa membuat pengisytiharan pemboleh ubah, dia secara tidak sengaja telah mengumpukkan pemboleh ubah "Tinggi" dengan *char* x. Semasa atur cara dijalankan, pengkompil telah mengeluarkan ralat masa larian. Ah Chong memeriksa atur caranya dan mendapati pemboleh ubah "Tinggi" telah diumpukkan dengan "x" dan bukannya nombor.

```
public class LuassSegitiga {  
    public static void main (String[] args){  
        // Pengisytiharan pemboleh ubah input  
        // dan output  
        int Tapak = 6;  
        int Tinggi = x; // ← Ralat Sintaks  
        double Luas;  
  
        // Proses yang terlibat dalam penghitungan  
        // luas segitiga  
        Luas = (1.0 / 2) * Tapak * Tinggi;  
  
        //Paparan output  
        System.out.println ("Luas Segitiga ialah : "  
        + Luas);  
    }  
}
```

Contoh atur cara bukan bebas ralat masa larian (Pengiraan data bukan berangka)

```
Exception in thread "main" java.lang.Error: Unresolved  
compilation problem:  
  x cannot be resolved to a variable  
  
  at LuassSegitiga.main(LuassSegitiga.java:12)
```

Contoh mesej ralat masa larian (Pengiraan data bukan berangka)



Inovasi Sains
Komputer



Robot direka oleh ASUS
untuk menjadi teman
digital kepada warga tua.
<https://www.zenbo.asus.com>



Run-time Errors
<http://goo.gl/6pMi06>

CONTOH 52

Rajah di bawah menunjukkan kod arus cara bukan bebas ralat untuk mengira purata tiga nombor. Bolehkah anda kenal pasti pada bahagian mana yang mungkin akan berlakunya ralat masa larian? Mari kita semak bersama-sama.

```
public class HelloWorld {  
    public static void main(String[] args) {  
  
        int a, b, c, d;  
        a = 3;  
        b = 5;  
        c = 10;  
        d = (a + b + c)/0;  
  
        System.out.println(d);  
    }  
}
```

Contoh arus cara bukan bebas ralat masa larian (Pembahagian dengan digit 0)

Selepas arus cara ini dijalankan, mesej ralat seperti rajah di bawah akan terpapar. Apakah yang berlaku? Cuba anda perhatikan pada formula yang digunakan. Pembahagian tiga nombor tersebut adalah dengan digit 0, iaitu merupakan salah satu faktor kepada berlakunya ralat masa larian.

```
Exception in thread "main" java.lang.ArithmeticException: / by zero  
at Purata_Tiga_Nombor.main(Purata_Tiga_Nombor.java:10)
```

Contoh mesej ralat masa larian (Pembahagian dengan digit 0)

CONTOH 53

Ralat masa larian juga berlaku sekiranya kesalahan dilakukan semasa mencari punca kuasa dua untuk nombor negatif. Seperti yang kita sedia maklum nombor negatif tidak mempunyai punca kuasa. Lihat rajah di bawah. Output yang akan keluar adalah seperti yang ditunjukkan.

```
public class Punca_Kuasa_Dua {  
    public static void main(String[] args) {  
  
        int x = 9;  
        int y = -25;  
        System.out.println("\n" + Math.sqrt(x));  
        System.out.println(Math.sqrt(y));  
    }  
}
```

Contoh arus cara bukan bebas ralat masa larian (Mencari punca kuasa dua nombor negatif)



```
C:\ Command Prompt  
C:\Users\fatisma\Desktop\WORK SPACE ECLIPSE JUNO\Tutoria  
l1\src>java Punca_Kuasa_Dua  
3.0  
NaN
```

Contoh output bukan bebas ralat masa larian (Mencari punca kuasa dua nombor negatif)

Ralat Logik

Ralat ini berlaku apabila atur cara tidak berfungsi seperti yang diingini. Ralat logik tidak dapat dikesan atau jarang dikesan oleh pengkompil. Pengatur cara sahaja yang boleh mengesan ralat logik melalui output yang dihasilkan.

Sekiranya output yang dihasilkan tidak menepati apa yang diinginkan, pengatur cara perlu memeriksa semua aspek output projek tersebut seperti pengiraan, teks dan penjarakan (*spacing*).



Logical errors in Java
<http://goo.gl/OYMZS>

CONTOH 54

Azian dan Mei Ling ingin menghasilkan atur cara yang boleh menghitung hasil tolak dua nombor. Selepas menulis kod atur cara, hasil data output yang diperoleh salah. Oleh sebab pengkompil tidak dapat mengesan ralat tersebut, Azian perlu memeriksa atur cara yang telah ditulisnya. Namun begitu, Azian boleh menganggarkan tempat berlakunya kesilapan dalam kod atur cara dengan berpandukan data output yang dipaparkan seperti rajah di bawah.

```
//Program Mengira Hasil Tolak Dua Nombor  
public class Tolak_Dua_Nombor {  
    public static void main(String[] args){  
  
        int nombor1 = 20;  
        int nombor2 = 13;  
        double hasil;  
  
        hasil = nombor1 + nombor2; // ← Ralat Logik  
  
        System.out.println("Hasil tolak ialah " + hasil );  
    }  
}
```

Formula matematik yang salah digunakan untuk menghitung hasil tolak dua nombor

```
C:\ Command Prompt  
C:\Users\fatisma\Desktop\WORK SPACE ECLIPSE JUNO\Tutoria  
l1\src>java Tolak_Dua_Nombor  
Hasil tolak ialah 33.0
```

Contoh output yang tidak tepat menunjukkan ralat logik berlaku

Lakukan penyelidikan ini secara berkumpulan dua hingga tiga orang. Bincang dengan rakan anda. Atur cara di bawah ialah sebuah program ringkas untuk mengira berat dan tinggi ideal bagi individu.

Sekiranya individu itu mempunyai berat kurang atau sama dengan 55 dan tinggi kurang atau sama dengan 145, maka paparan output ialah "Awak langsing" dan sebaliknya.

Walau bagaimanapun atur cara di bawah bukan atur cara bebas ralat. Senaraikan jenis ralat yang mungkin berlaku dalam kod atur cara di bawah dan nyatakan bagaimana ralat tersebut akan muncul apabila sesebuah atur cara dijalankan. Gunakan kemudahan enjin carian untuk mencari lebih maklumat. Bezakan jenis ralat dan dan nyatakan juga sebarang persamaannya, jika ada. Persembahkan hasil carian anda kepada rakan sekelas. (Cadangan – gunakan peta *i-Think* sekiranya sesuai).

```

import java.util.Scanner;

public class UsingJikaTidakScanner {
    public static void main(String[] args)

        Scanner input = new Scanner (System.in);

        System.out.println("Sila masukkan berat: ");
        int berat = input.nextInt();

        System.out.println("Sila masukkan tinggi: ")
        int tinggi = input.nextInt();

        if(berat <= 55 && tinggi >= 155){
            System.out.println("Awak langsing");
        }
        else {
            System.out.println("Awak gempal");
        }
    }
}
    
```

```

C:\Users\fatisma\Desktop\WORK SPACE ECLIPSE JUNO\projekUtama\src>java Mengira_Berat_Tinggi_Ideal
Sila masukkan berat:
45
Sila masukkan tinggi:
146
Awak gempal
    
```

1.5.2 Pengesahan, Pengenalpastian dan Penterjemahan Mesej Ralat serta Pembaikan Ralat



Common runtime error
solution
<http://goo.gl/g2924s>

Setelah memahami perbezaan yang diterangkan bagi ketiga-tiga ralat atur cara dalam pengaturcaraan, anda akan boleh mengesahkan, mengenal pasti dan menterjemahkan mesej ralat serta membaiki ralat. Rajah 1.43(a) menunjukkan atur cara mengira luas bulatan bukan bebas ralat. Mari kita kesan ralat yang berlaku dan buat pembaikan!

```
public class LuasBulatan {  
    public static void main(String[] args){  
  
        int jejari = 5;  
        double luasBulatan;  
        final double pi= 3.142;  
  
        luasBulatan = pi * jejari;  
  
        System.out.println("Luas bulatan ialah " + luasbulatan);  
    }  
}
```

Rajah 1.43 (a) Atur cara menghitung luas bulatan bukan bebas ralat

```
Exception in thread "main" java.lang.Error: Unresolved compilation  
problems:  
  Syntax error, insert ";" to complete BlockStatements  
  luasbulatan cannot be resolved to a variable  
  
  at LuasBulatan.main(LuasBulatan.java:5)
```

Rajah 1.43 (b) Pengkompil mengesahkan ralat dalam kod atur cara Rajah 1.43(a)

Apabila atur cara pada Rajah 1.43(a) dijalankan, kotak dialog ralat pada Rajah 1.43(b) dipaparkan. Apakah yang berlaku?

1. Semak semula atur cara pada bahagian pengisytiharan.
2. Pastikan semua tatatanda ditaip dengan lengkap.

3. Pastikan nama pemboleh ubah yang diisyiharkan adalah sama dengan nama pemboleh ubah yang akan dipanggil balik dalam atur cara. Yang dimaksudkan adalah sama dari segi ejaan dan penggunaan huruf kecil dan huruf besar. (INGAT! Dalam pengaturcaraan Java, penggunaan huruf kecil dan huruf besar adalah kes sensitif.)

Kita dapat mengesan kesalahan dalam atur cara seperti contoh yang berikut:

Kenal pasti Jenis Ralat	Pembaikan Ralat
1 Ralat Sintaks <code>int jejari = 5</code> <code>double luasBulatan;</code> <code>System.out.println("Luas bulatan ialah "</code> <code>+ luasBulatan);</code>	<code>int jejari = 5;</code> <code>System.out.println("Luas bulatan ialah "</code> <code>+ luasBulatan);</code>

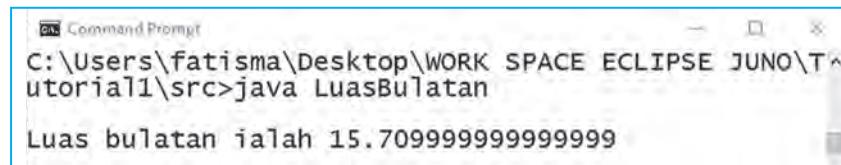
Penerangan

- Setiap baris atur cara yang lengkap akan diakhiri dengan simbol “;”. Pengisyiharan pemboleh ubah `int jejari` = 5 tidak diakhiri dengan simbol “;”.
- Luas bulatan diisyihar sebagai `double` luasBulatan; iaitu ejaan Bulatan berhuruf besar, tetapi di dalam baris atur cara output, `System.out.println("Luas bulatan ialah " + luasBulatan)`, Bulatan dieja menggunakan huruf kecil.


Aplikasi Harian

Pembelian tiket LRT di mesin tiket telah memudahkan para penumpang. Selain itu, tiket LRT juga boleh dibeli secara atas talian. Pembelian atas talian telah diprogramkan untuk mengelakkan kesesakan di mesin tiket.

4. Setelah ralat dikenal pasti, ralat sintaks boleh dibaiki.
5. Setelah mengesan dan membaiki ralat sintaks, atur cara dapat dijalankan tetapi jawapan pada paparan output tidak tepat. Lihat Rajah 1.43(c).



```
C:\Users\fatisma\Desktop\WORK SPACE ECLIPSE JUNO\Tutorial1\src>java LuasBulatan
Luas bulatan ialah 15.709999999999999
```

Rajah 1.43 (c) Paparan data output luas bulatan yang tidak tepat

Jawapan tepat bagi luas bulatan dalam Rajah 1.43(a) ialah 78.55 tetapi sistem memberikan jawapan 15.709 (Rajah 1.43(c)). Apakah yang berlaku?

1. Semak semula atur cara pada bahagian penggunaan formula.
2. Daripada Rajah 1.43(a), kita dapat mengesan kesalahan dalam formula mengira luas bulatan.

Ralat Logik	Pembaikan Ralat Logik
<code>luasBulatan = pi * jejari;</code>	<code>luasBulatan = pi * jejari * jejari;</code>

- Ralat logik sukar dikesan pada peringkat awal. Ralat ini selalunya dapat dikesan apabila paparan hasil yang dikehendaki tidak menepati apa yang diingini.
- Setelah ralat dikenal pasti, ralat logik boleh dibaiki.

```
public class LuasBulatan {
    public static void main(String[] args) {
        int jejari = 5;

        double luasBulatan;
        final double pi= 3.142;

        luasBulatan = pi * jejari * jejari;

        System.out.println("Luas bulatan ialah " +
        luasBulatan);
    }
}
```

Tahukah Anda?

Sintaks “\n” -digunakan untuk mewujudkan jarak antara baris (*line spacing*). “\n” juga menandakan untuk memasukkan baris teks yang baharu.

Rajah 1.44 Atur cara menghitung luas bulatan bebas ralat



25

Kenal pasti dan baiki ralat

Kerja Berkumpulan

Secara berpasangan, taip semula atur cara di bawah dalam persekitaran platform Java. Jalankan atur cara anda. Buat “Screenshot” atur cara yang telah anda taip dan paparan ralat yang mungkin anda peroleh.

Kesan ralat tersebut. Bincang dengan rakan anda apakah jenis ralat yang mungkin berlaku. Kenal pasti ralat tersebut dan baiki ralat. Masukkan kod atur cara yang mengandungi ralat dan kod selepas dibaiki dalam jadual di bawah.

```
public class kilometerKepadaMeter {

    public static void main(String[] args) {
        int Kilometer = 20;
        int meter;

        System.out.println("\n" + kilometer + " Kilometer"
    }
}
```

Jenis ralat	Kenapa anda memilih jenis ralat tersebut? Justifikasikan jawapan anda	Kod atur cara bukan bebas ralat	Kod atur cara bebas ralat



Uji Minda

Sekiranya anda memasukkan nilai input 8 sebagai nilai jejari untuk sebuah silinder dengan ketinggian 12 cm, berapakah nilai pemboleh ubah output bagi isi padu silinder tersebut?

1.5.3 Mengenal Pasti Nilai bagi Pemboleh Ubah pada Bahagian Tertentu Atur Cara

Pemboleh ubah digunakan untuk menyimpan data input dan data output. Dalam pengaturcaraan, pemboleh ubah digunakan sebagai pemudah cara bagi komputer untuk menerima, menghantar, memproses dan menyebabkan operasi terhadap sesuatu input.

Sekiranya pengguna memasukkan nilai pemboleh ubah yang diingini, pemproses akan memproses nilai data input itu dan mengeluarkan suatu nilai baharu iaitu output. Untuk mengira luas bulatan, murid boleh memasukkan sebarang nilai sebagai pemboleh ubah input ke dalam formula πr^2 . Sekiranya murid memasukkan nilai input 5 sebagai nilai jejari, pemproses akan melaksanakan pengiraan iaitu πr^2 atau $3.142 * 5 * 5$ dan output yang terhasil ialah 78.55. Oleh itu, nilai bagi pemboleh ubah input ialah 5 dan nilai bagi pemboleh ubah output ialah 78.55.

CONTOH (55)

```
public class LuasBulatan {
    public static void main(String[] args) {

        int jejari = 5;
        double luasBulatan;
        final double pi= 3.142;

        luasBulatan = pi * jejari * jejari;

        System.out.println("Luas bulatan ialah " + luasBulatan);
    }
}
```

Atur cara menghitung luas bulatan

Pemboleh ubah	Input	Ouput
Item pemboleh ubah	jejari	luas
Nilai (Data pemboleh ubah)	5	78.55

CONTOH (56)

Azliza dan dua orang rakananya ingin membeli tiket menaiki bas dari Melaka ke Johor menggunakan mesin tiket di stesen bas Melaka. Harga perjalanan bagi seorang penumpang ialah RM45.00. Nyatakan pemboleh ubah input dan output bagi pembelian tiket bas di mesin tiket. Berikut ialah kod atur cara yang digunakan untuk menghitung jumlah harga tiket yang perlu dibayar oleh Azliza dan dua orang rakananya.

```
import java.text.DecimalFormat;
public class HargaTiket {

    public static void main(String[] args) {
        int bilangan_tiket = 3;
        double harga_tiket = 45.00;
        double jumlah_harga;
        DecimalFormat df = new DecimalFormat("#.00");
        System.out.println("\n" + "Harga tiket untuk " +
            "seorang penumpang = " + "RM" +
            df.format(harga_tiket));
    }
}
```

Atur cara menghitung jumlah harga tiket bas yang dibeli

Penyelesaian

Pemboleh ubah	Input	Ouput
Item pemboleh ubah	bilangan_tiket harga_tiket	Jumlah_harga
Nilai (Data pemboleh ubah)	3 45.00	135.00



Permainan *Alien Hive* telah direka dan dibangunkan oleh *Appxplore*, syarikat pembangunan permainan video yang beroperasi di Kuala Lumpur. Sejak dilancarkan pada tahun 2013, permainan video ini telah dimuat turun sebanyak tiga juta kali.



Tahukah
Anda ?

```
import java.text.*;
DecimalFormat;
DecimalFormat
df = new
DecimalFormat
("#.00");
```

Sintaks ini digunakan untuk menetapkan nilai tempat perpuluhan



Uji Minda

Anda dan rakan anda telah pergi ke Zoo Negara untuk mengisi masa pada hujung minggu. Harga tiket bagi seorang individu ialah RM25.00. Sekiranya terdapat potongan harga sebanyak 30% untuk murid sekolah, nyatakan pemboleh ubah input dan output bagi jumlah harga bagi jumlah harga tiket anda dan rakan anda.



<https://goo.gl/tYR8YL>



BAB 1 PENGATURCARAAN



Kenal pasti nilai pemboleh ubah dalam bahagian tertentu subatur cara

Bincang dengan pasangan anda. Kenal pasti nilai pemboleh ubah data bagi atur cara di bawah. Berikan justifikasi bagi jawapan anda.

```

import java.text.DecimalFormat;
public class Mengira_Jumlah_Belian {
    public static void main(String[] args){

        DecimalFormat df = new DecimalFormat("#.00");

        double milo;
        double susu_cair;
        double gula;
        double jumlah_belian;

        milo = 19.90 * 3 * (100 - 5) / 100;
        susu_cair = 2.80 * 6 * (100 - 2) / 100;
        gula = 3.00 * 5 * (100 - 3) / 100;

        jumlah_belian = milo + susu_pekat + tepung;
    }
}
    
```



Aplikasi Harian



<https://repl.it>

Laman sesawang ini merupakan pengkompil dalam talian yang mudah digunakan. Anda perlu menaip kod atur cara dan anda akan dibantu untuk menguji dan memastikan kod anda bebas ralat. Laman sesawang ini boleh digunakan untuk pelbagai bahasa pengaturcaraan.

1.5.4 Penghasilan Atur Cara yang Mudah Dibaca dengan Menggunakan Gaya yang Baik (Komen, Nama Pemboleh Ubah yang Bermakna dan Inden)

Untuk menghasilkan atur cara yang baik dan mudah difahami, pengatur cara perlu menggunakan gaya yang baik seperti yang berikut:

- Meletakkan komen pada setiap fungsi yang dibuat.
- Menggunakan nama pemboleh ubah yang bermakna.
- Menggunakan inden yang selesa untuk dibaca.

Apakah yang dimaksudkan dengan gaya yang baik? Mari kita terokai!

Komen

Komen merujuk penanda yang dibuat oleh pengatur cara untuk setiap atur cara yang dibina. Setiap bahasa pengaturcaraan mempunyai kod yang khusus untuk komen. Dalam pengaturcaraan Java, kod komen perlu menggunakan sintaks yang telah ditetapkan. Sekiranya sintaks tersebut tidak dimasukkan pada baris komen dalam atur cara tersebut, sistem akan mengeluarkan ralat sintaks/kompil.

Tatatanda komen dalam Java perlu dilakukan dengan betul. Terdapat **tiga** gaya tatabanda komen yang boleh digunakan iaitu:

Jadual 1.15 Jenis komen dan huraiannya

Jenis komen	Huraian
// Pengisyiharan pemboleh ubah	Pengkompil mengabaikan semua teks bermula dengan tatabanda // hingga ke teks terakhir dalam baris yang sama.
/* Pengisyiharan pemboleh ubah*/	Pengkompil mengabaikan semua teks yang berada dalam /* hingga ke */ walaupun dalam baris yang berlainan.
/** Pengisyiharan pemboleh ubah*/	Komen ini ialah komen dokumentasi. Pengkompil mengabaikan komen jenis ini sama seperti komen yang menggunakan /* dan */.

CONTOH 57

Penggunaan tatabanda “//” , “/* */” dan “/** */” sebagai sintaks komen. Penyataan yang berwarna hijau merujuk komen.

```
/* Program ringkas untuk mengira Luas Segitiga
Langkah 1: Baca Tapak
Langkah 2: Baca Tinggi
Langkah 3: Hitung Luas
Langkah 4: Papar Luas
*/
public class LuasSegitiga {
    public static void main (String[] args){
        /**Pengisyiharan pemboleh ubah input dan output*/
        int Tapak = 6;
        int Tinggi = 4;
        double Luas;

        // Proses yang terlibat dalam penghitungan
        // luas segitiga
        Luas = (1.0 / 2) * Tapak * Tinggi;

        //Paparan output
        System.out.println ("Luas Segitiga ialah : " + Luas);
    }
}
```

Atur cara dengan penggunaan komen



Java documentation
comments
<http://goo.gl/iu1SMo>



Anda boleh layari laman sesawang untuk menetapkan inden bagi kod atur cara anda
<https://goo.gl/CNlMC8>

Pemboleh ubah yang Bermakna

Semasa menulis atur cara, pengatur cara perlu memikirkan nama pemboleh ubah yang mempunyai ejaan yang ringkas dan bermakna dalam atur cara yang sedang dilaksanakan. Rujuk contoh 58.

Inden

Inden merujuk cara penulisan atur cara yang memudahkan pembacaan. Pembacaan atur cara akan dimulakan dengan inden iaitu barisan teks berada di beberapa kedudukan aksara ke dalam, dari jidar kiri atau kanan halaman. Rujuk rajah pada contoh 58.

CONTOH (58)

Untuk menghitung luas segi tiga, formula yang digunakan ialah “ $1/2 * p * q$ ”. Sekiranya pengatur cara hanya mengisyiharkan pemboleh ubah yang perlu digunakan sebagai p dan q sahaja, nama pemboleh ubah tersebut tidak bermakna. Tetapi sekiranya pengatur cara menggunakan nama pemboleh ubah tapak dan tinggi, maka pengatur cara lain senang untuk memahaminya.

Inden

```
/* Program menghitung luas segitiga*/
public class LuasSegitiga {
    public static void main (String[] args){
        /*Pengisyiharan pemboleh ubah input dan output*/
        int Tapak = 6;
        int Tinggi = 4; → Pemboleh ubah yang bermakna
        double Luas;

        // Proses yang terlibat dalam penghitungan
        // luas segitiga
        Luas = (1.0 / 2) * Tapak * Tinggi;

        //Paparan output
        System.out.println ("Luas Segitiga ialah : "
        + Luas);
    }
}
```

Penggunaan nama pemboleh ubah yang bermakna dan penggunaan inden

Komen, Pemboleh ubah yang bermakna dan Inden

Bincangkan. Rancang dan hasilkan atur cara bersama-sama rakan anda. Adan, Samy dan Tan telah diminta oleh guru untuk membangunkan satu atur cara mudah untuk mengira isi padu sebuah silinder. Atur cara tersebut mestilah boleh menerima data input daripada pengguna dan dapat memaparkan data output kepada pengguna. Atur cara tersebut mestilah bebas ralat dan menggunakan amalan terbaik pengaturcaraan seperti penggunaan komen, inden dan nama pemboleh ubah yang bermakna. Anda boleh gunakan senarai semak seperti yang dicadangkan di bawah.

Kriteria	Input	Ouput
Pemboleh ubah bermakna		
Nilai data pemboleh ubah		
Tanda Ada (/) atau Tiada (X)		
	Ada	Tiada
Komen		
Inden		

Latihan Formatif 1.5

- 1 Nyatakan pernyataan berikut sama ada benar atau palsu.

(a)	Pemboleh ubah yang mempunyai dua perkataan boleh diterima dalam penulisan atur cara tanpa menggunakan garis bawah (<i>underscore</i> "_")	
(b)	Penggunaan nama pemboleh ubah yang ringkas dan bermakna digalakkkan dalam penulisan atur cara.	
(c)	Panjang aksara pemboleh ubah adalah antara 1 hingga 256 aksara.	

- 2 Apakah yang anda faham tentang pemboleh ubah dalam suatu atur cara komputer? Jelaskan.
- 3 Senaraikan gaya penulisan atur cara yang perlu dipraktikkan oleh seorang pengatur cara. Justifikasikan jawapan anda.
- 4 Apakah yang anda faham dengan penggunaan inden dalam pengaturcaraan komputer?
- 5 Senaraikan tiga jenis ralat yang sering berlaku dalam pengaturcaraan Java. Terangkan secara ringkas dua daripada tiga jenis ralat tersebut.
- 6 Senaraikan faktor yang mempengaruhi kebolehbacaan kod atur cara. Justifikasikan jawapan anda.



- 7 Di bawah merupakan contoh atur cara untuk menghitung operasi bahagi bagi dua nombor. Tulis semula atur cara tersebut. Anda perlu mengemas kini kod atur cara tersebut bagi memudahkan pengatur cara lain membaca dan memahami atur cara tersebut. Berikan penekanan kepada penggunaan komen dan inden dalam atur cara tersebut.

```
import java.text.DecimalFormat;

public class Hasil_Bahagi_Dua_Nombor {
    public static void main(String[] args) {
        int nombor1 = 5;
        int nombor2 = 6;
        double hasil;
        DecimalFormat df = new DecimalFormat("#.00");
        hasil = (nombor1 + nombor2) / 2.0;
        System.out.println("Hasil bahagi ialah " +
            df.format(hasil));
    }
}
```



- 8 Di bawah merupakan keratan kod atur cara bukan bebas ralat. Anda perlu mengesan dan membaiki ralat dalam atur cara tersebut dengan menaip semula atur cara tersebut dalam persekitaran platform Java. Nyatakan jenis ralat tersebut dan cadangkan cara paling sesuai untuk menyelesaikannya.

```
public class HelloWorld {
    public static void main(String[] args) {

        String x, y, z;

        x = "Hello ";
        y = "World "
        z = "Malaysia";
        System.out.println(x + y + "kepada" + "\n" + z);

    }
}
```



- 9 Anda perlu tulis sebuah program yang dapat menghitung dan membuat penukaran unit bagi suhu daripada *Fahrenheit* kepada *Celsius*. Formula penukaran adalah seperti di bawah.

$$\text{Celsius} = (5.0 / 9) * (\text{Fahrenheit} - 32)$$

Program anda mestilah berupaya membaca darjah *Fahrenheit* jenis data *double* daripada pengguna. Kemudian, program ini boleh menukar nilai tersebut kepada *Celsius* dan memaparkan keputusan betul kepada dua titik perpuluhan. Berikan penekanan kepada penggunaan komen dan inden dalam atur cara anda.

STANDARD PEMBELAJARAN

- 1.6.1 Menerangkan struktur tatasusunan (*array*) satu dimensi:
 - (ii) Isytihar nilai
 - (ii) Umpukan nilai awal
- 1.6.2 Menggunakan subatur cara dan memahami konsep menghantar parameter ke subatur cara dan mengembalikan data
- 1.6.3 Membezakan antara *function* dan *procedure* pada subatur cara
- 1.6.4 Menulis atur cara bermodular yang mengandungi struktur tatasusunan

Struktur Data dan Modular

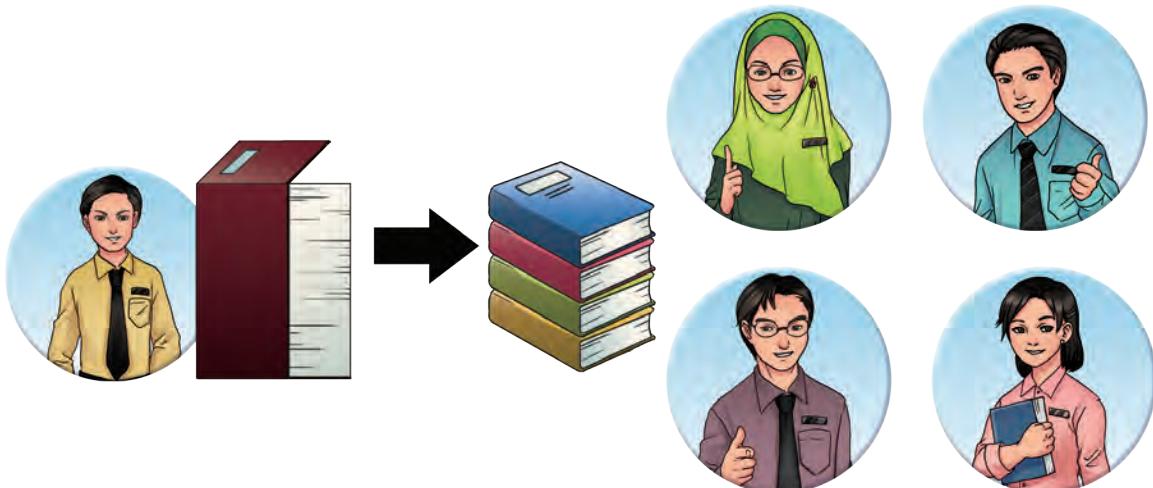
Atur cara mengandungi kod komputer untuk memproses data kepada maklumat. Data perlu disimpan dalam pemboleh ubah supaya boleh diproses. Arahan kod komputer dan pemboleh ubah disimpan dalam fail kod sumber (*source code*).

Atur cara yang ringkas mungkin mengandungi beberapa baris arahan dan pemboleh ubah sahaja. Untuk atur cara yang kompleks seperti sistem pinjaman buku sekolah dan sistem-sistem lain pengurusan maklumat, mungkin ratusan ribu baris arahan dan ratusan pemboleh ubah diperlukan. Dapatkan bayangan kesukaran untuk menulis kod komputer untuk sistem yang kompleks?

Kod komputer seharusnya mudah ditulis, mudah dibaca dan mudah dikemas kini. Bagaimana?

Gunakan struktur yang lebih sistematik untuk pemboleh ubah dan arahan. Pemboleh ubah boleh dipecahkan kepada "kumpulan-kumpulan kecil" yang dipanggil **tatasusunan**. Arahan-arahan komputer juga boleh dipecahkan kepada beberapa "kumpulan kecil" yang dipanggil fungsi. Apabila diperlukan sahaja, kumpulan yang berkait akan dipanggil.

Sebagai analogi, bayangkan terdapat sebuah buku berjudul "Semua Mata Pelajaran Tingkatan 4" dan hanya seorang guru yang akan mengajarkan semua mata pelajaran tersebut. Dalam buku tersebut, tiada bab-bab berasingan mahupun subtopik. Semuanya ditulis dalam satu perenggan sahaja dari muka surat satu hingga muka surat akhir. Adakah ini satu kaedah yang baik dalam pengajaran dan pembelajaran?



Rajah 1.45 "Pecah-dan-Urus" adalah lebih berkesan





Uji Minda

Apakah kelebihan yang terdapat pada tatasusunan (*array*)?



Apakah perbezaan antara algoritma dengan struktur data?

Algoritma ialah kaedah melakukan proses tertentu terhadap data seperti menyusun (*sorting*), mencari (*searching*), mencari nombor prima dan memproses nombor rawak dan grafik.

Struktur data merupakan satu kaedah tertentu untuk menyimpan secara tersusun data-data dalam ingatan supaya senang dicapai untuk diproses bagi menghasilkan maklumat mengikut kehendak pengguna. Data boleh disusun dalam bentuk tatasusunan (*array*) dan vektor (*vector*), senarai pautan (*linked lists*), timbunan (*stack*) dan giliran (*queue*).

Lihat Rajah 1.45. Apa kata sekiranya kandungan sebuah buku dibahagikan kepada bab-bab berasingan dan dijilidkan menjadi beberapa buah buku berlainan? Bukankah pengetahuan-pengetahuan dalam buku tebal menjadi lebih mudah diurus dan dibaca? Adakah anda perlu membawa semua buku tersebut sekiranya subjek tersebut tidak diajar pada hari-hari tertentu?

Demikian halnya dengan guru-guru yang mengajar pengetahuan tersebut. Guru-guru yang ada pengkhususan dalam subjek-subjek tertentu adalah lebih arif dalam subjek tersebut sahaja. Sekiranya murid ingin bertanyakan permasalahan Matematik, murid boleh mencari guru matematik untuk mendapat penjelasan lanjut.

Demikian halnya dengan atur cara komputer. Buku-buku dalam analogi yang ditunjukkan pada Rajah 1.45 adalah ibarat pemboleh ubah manakala guru-guru yang ada pengkhususan dalam subjek tersebut adalah ibarat arahan-arahan atur cara dalam fail kod sumber.

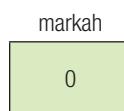
1.6.1 Struktur Tatasusunan (*array*) Satu Dimensi

Sistem pengurusan maklumat melibatkan pemprosesan suatu jumlah data yang banyak. Bagi menguruskan data agar lebih tersusun dan senang dicapai, maka diperkenalkan konsep struktur data. Sebaik manapun algoritma, jika data tidak tersusun, capaian dan pemprosesan data menjadi kurang efisien. Malah, penulisan kod komputer menjadi amat sukar.

Salah satu struktur data yang penting ialah **tatasusunan** (*array*). Tatasusunan ialah pemboleh ubah yang membolehkan koleksi beberapa nilai data (*elemen*) dalam satu-satu masa dengan menyimpan setiap elemen dalam ruang memori berindeks.

Dalam subtopik ini, anda akan mengkaji pemboleh ubah mudah ataupun biasa semula sebagai sel memori dan perbezaannya dengan tatasusunan. Anda akan melihat kelebihan-kelebihan tatasusunan sebagai senarai data berbanding dengan pemboleh ubah mudah.

Pemboleh ubah ialah slot memori yang telah dikhaskan untuk menyimpan data. Kebiasaannya, pemboleh ubah mudah cuma menyimpan satu nilai data dalam satu-satu masa. Lihat ruang memori untuk pemboleh ubah mudah di bawah.



Rajah 1.46 Pemboleh ubah mudah menggunakan hanya satu kotak

Dalam kod Java, nilai boleh ubah diisytihar seperti sintaks seperti yang berikut:

```
jenisData namaPbolehUbah;
```

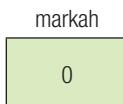
Contoh 1:

```
int markah;
```

Jika Java perlu menyimpan data bernilai 34, umukkan sahaja nilai 34 ke dalam boleh ubah markah. Kod Java boleh ditulis seperti yang berikut.

Contoh 2:

```
markah = 34
```



Rajah 1.47 Lukisan blok memori untuk pemboleh ubah mudah dengan nilai 34

Kedua-dua langkah di atas boleh dibuat sekali gus dalam satu arahan.

Contoh 3:

```
int markah = 34
```

Dalam analogi yang berikut, boleh ubah yang telah diisytiharkan diibaratkan sebuah bilik kosong yang boleh diduduki oleh satu penghuni sahaja pada satu-satu masa. Hal ini bermaksud, sekiranya jenis data boleh ubah ialah integer, maka nilai yang boleh disimpan ialah integer sahaja. Dalam analogi rumah kosong yang berikut, nama rumah tersebut ialah markah dan rumah ini hanya membenarkan tetamu jenis integer sahaja untuk datang menginap.

Analogi pemboleh ubah mudah:

Rajah 1.48 menunjukkan analogi bagi sebuah rumah yang kosong. Rumah ini dinamakan "Bilik Bujang". Rumah ini hanya mempunyai satu bilik untuk seorang penghuni sahaja. Penghuni tersebut mestilah khusus, misalnya untuk kakitangan sekolah yang berjawatan guru dan bujang.

Sekarang, cuba anda bayangkan rumah tersebut ialah boleh ubah mudah yang bernama markah dan hanya membenarkan satu nilai disimpan serta nilai tersebut mestilah berjenis int sahaja.

Bagaimana sekiranya terdapat banyak data jenis integer yang perlu disimpan? Sebagai contoh, data-data jenis integer 34, 56, 78, 89, 56 dan 95 perlu disimpan dalam boleh ubah. Kita hanya perlu isytihar lebih banyak boleh ubah sahaja, bukan?



Fundamental Programming
Structures in Java
<http://goo.gl/FQql8g>

Tahukah Anda ?

Komputer mempunyai tatasuhan yang sangat besar. Hal ini adalah kerana tatasuhan merupakan cara terbaik untuk memproses data yang sangat besar terdapat di dalam komputer.



Rajah 1.48 Pemboleh ubah mudah diibaratkan sebagai rumah dengan satu bilik untuk tetamu khusus





Java Arrays
<http://goo.gl/YrU7Qm>

Contoh 4:

```
int markah1 = 34
int markah2 = 56
int markah3 = 78
int markah4 = 89
int markah5 = 56
int markah6 = 95
```

Bagaimana pula jika terdapat 100 markah yang perlu disimpan? Penggunaan pemboleh ubah nampaknya kurang ekonomik jika bilangan data yang disimpan adalah besar. Cuba fikirkan. Adakah wajar mengisytiharkan 100 pemboleh ubah?

Contoh-contoh di atas melibatkan penggunaan jenis data mudah. Jenis data mudah menggunakan satu sel ingatan sebagai pemboleh ubah untuk menyimpan suatu data.

Tahukah Anda?

Tatasusunan bukan sahaja terdapat dalam satu dimensi tetapi juga dua dimensi. Bagaimanakah cara untuk membezakan antara dimensi ini? Andaikan anda menyenaraikan semua makanan yang anda makan ketika waktu tengah hari. Misalnya, kentang, ikan, ayam dan nasi. Makanan yang disenaraikan di atas dianggap sebagai satu dimensi. Tetapi, jika anda mengelaskannya mengikut piramid makanan seperti protein, karbohidrat, lemak dan vitamin, senarai tersebut akan menjadi dua dimensi. Misalnya,
Protein:
(ikan dan ayam)
Karbohidrat:
(nasi dan kentang)

Pengisytiharan Tatasusunan

Dalam kehidupan harian, biasanya data yang sama akan disimpan dalam senarai. Dalam konteks atur cara komputer, senarai data dirujuk sebagai tatasusunan.

Tatasusunan ialah koleksi satu atau lebih data yang dipanggil elemen. Setiap tatasusunan mempunyai jenis data tertentu. Tatacusunan *int* menyimpan elemen jenis integer sahaja. Demikian juga halnya dengan *double*, *String* dan jenis-jenis data yang lain.

Dalam kod *Java*, sintaks pengisytiharan struktur tatasusunan adalah seperti yang berikut:

```
jenisData [] namaTatasusunan;
namaTatasusunan = new jenisData [saizTatasusunan];
```

Sintaks yang ditunjukkan di atas terdiri daripada dua baris. Baris pertama mengisytiharkan nama tatasusunan. Baris kedua pula mengisytiharkan saiz tatasusunan.

- Mengisytiharkan nama tatasusunan

```
jenisData [] namaTatasusunan;
```

jenisData merujuk jenis data tatasusunan. Simbol “tanda kurung siku” [], ialah simbol khas untuk menunjukkan bahawa pemboleh ubah ialah jenis tatasusunan dan bukannya pemboleh ubah mudah, misalnya *int* [] untuk tatasusunan integer, *double* [] untuk tatasusunan *double* ataupun *String* [] untuk tatasusunan *String*. *namaTatasusunan* ialah nama yang sesuai diberikan oleh pengatur cara.



Contoh:

```
int [] senaraiMarkah;
```

Dalam contoh ini, jenis tatasusunan `int` ialah integer. Simbol `[]` merujuk jenis pemboleh ubah iaitu **tatasusunan**. Nama tatasusunan ialah `senaraiMarkah`.

- Saiz tatasusunan:

```
namaTatasusunan = new jenisData [saizTatasusunan];
```

Dalam baris kedua, bilangan elemen dalam tatasusunan diisyiharkan. Kata kunci `new` digunakan untuk set saiz tatasusunan. Ini diikuti oleh jenis data `jenisData` dan saiz tatasusunan dalam tanda kurung `[]`.

Contoh:

```
senaraiMarkah = new int[6];
```

Dalam contoh ini, pemboleh ubah tatasusunan boleh mengandungi enam elemen pemboleh ubah jenis integer, seperti yang dinyatakan dalam `new int[6]`. Enam ruang memori yang kosong diperuntukkan untuk menyimpan data jenis integer dan semuanya adalah untuk tatasusunan bernama `senaraiMarkah`.

Baris pertama mesti ditulis dahulu sebelum baris kedua. Pengisytiharan tatasusunan bernama `senaraiMarkah` dibuat seperti susunan yang berikut:

```
int [] senaraiMarkah;  
senaraiMarkah = new int[6];
```

Lihat ruang memori untuk tatasusunan dalam jadual berikut dan bandingkan dengan ruang memori pemboleh ubah mudah.

	[0]	[1]	[2]	[3]	[4]	[5]
senaraiMarkah	0	0	0	0	0	0

Rajah 1.49 Ruang memori untuk tatasusunan

Perhatikan bahawa tatasusunan ialah koleksi-koleksi sel memori dengan satu nama dan indeks. Indeks tatasusunan bermula daripada sifar. Jadi, dalam contoh enam elemen, indeks ialah 0, 1, 2, 3, 4 dan 5. Perhatikan bahawa nilai-nilai elemen ialah 0 dalam setiap sel. Hal ini demikian kerana tatasusunan ini masih belum diberikan sebarang nilai.

Jadual 1.16 Ruang memori pemboleh ubah mudah berbanding ruang memori tatasusunan

Pemboleh ubah Mudah	Pemboleh ubah Tatasusunan
markah 0	senaraiMarkah [0] 0 [1] 0 [2] 0 [3] 0 [4] 0 [5] 0

Lihat analogi dalam Rajah 1.50. Bayangkan sebuah bangunan yang panjang dengan nama Asrama Bujang. Terdapat enam bilik untuk seorang guru yang berstatus bujang bagi setiap satunya. Perhatikan bahawa setiap bilik mempunyai nama bilik yang menggunakan indeks. Indeks ialah senarai nombor yang bermula daripada sifar. Indeks memudahkan pemilik bangunan “AsramaBujang” untuk mengesan bilik mana yang kosong dan mengemaskinkan status bilik. Sebagai contoh, nombor-nombor bilik ialah no[0], no[1], no[2], no[3], no[4] dan no[5].



Rajah 1.50 Analogi pemboleh ubah jenis tatasusunan ialah sebuah bangunan dengan banyak bilik

Tahukah Anda?

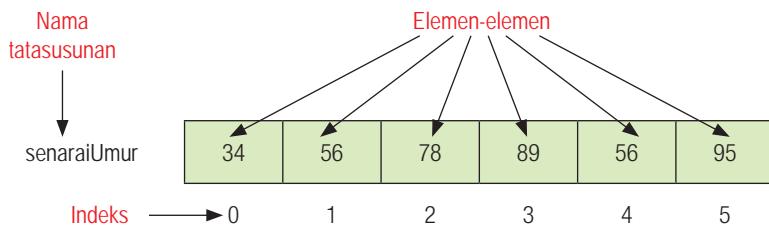
- Mari fikirkan sejenak.
- Bilik-bilik dengan nama no[0] - no[5] ialah jenis bangunan Asrama Bujang.
 - Bilik pertama ialah no[0].
 - Nama bilik no[5] ialah bilik keenam.
 - Nama bilik no[6] tidak wujud dalam contoh ini.

Pengumpukan Tatasusunan

Pengisytiharan tatasusunan menyediakan ruang memori yang masih kosong. Oleh sebab itu, nilai perlu diberikan melalui proses pengumpukan. Selepas diisyihar, nilai diumpuk dengan memanggil elemen-elemen tatasusunan satu-persatu:

```
senaraiMarkah[ 0 ] = 34 ;
senaraiMarkah[ 1 ] = 56 ;
senaraiMarkah[ 2 ] = 78 ;
senaraiMarkah[ 3 ] = 89 ;
senaraiMarkah[ 4 ] = 56 ;
senaraiMarkah[ 5 ] = 95 ;
```

Dalam memori komputer, elemen-elemen dalam tatasusunan sentiasa diletakkan sebelah-menyebelah dalam satu blok memori.



Rajah 1.51 Sel memori menunjukkan elemen-elemen dalam tatasusunan

Pengumpukan Nilai Awal Tatasusunan

Dalam pengumpukan nilai awal, umpukan dibuat ketika melakukan pengisytiharan. Sebagai contoh:

```
int [ ] senaraiMarkah = { 34 , 56 , 78 , 89 , 56 , 95 }
```

↓ ↓
Pengisytiharan Pengumpukan nilai awal



Arrays
<http://goo.gl/JiHxCA>

Kombinasi jenis data dan tanda kurung `int []` adalah untuk mengisyiharkan nama `senaraiMarkah` sebagai pemboleh ubah tatasusunan integer. Oleh sebab umpukan awal digunakan, saiz tatasusunan tidak perlu dimasukkan dalam tanda kurungan `[]` pengisytiharan.

Saiz tatasusunan ditentukan secara automatik oleh pengumpukan nilai-nilai berdasarkan bilangan data dalam tanda kurung dakap "`{`" dan "`}`". Semua data yang hendak disimpan ialah satu jenis yang sama.

Jadual 1.17 Lukisan blok memori pemboleh ubah mudah berbanding lukisan blok memori tatasusunan

Pemboleh ubah Mudah	Pemboleh ubah Tatasusunan
markah 34	senaraiMarkah [0] 34 [1] 56 [2] 78 [3] 89 [4] 56 [5] 95

Perbezaan Struktur Memori antara Pemboleh Ubah Mudah dengan Memori Tatasusunan

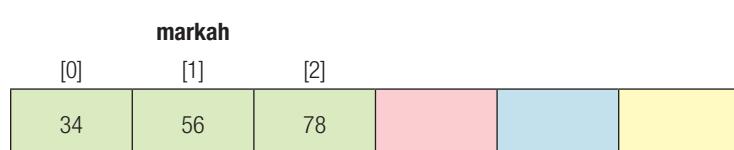
Katakan seorang guru ingin merekodkan markah bagi subjek Teknologi Maklumat (TM) yang diajarinya. Dalam segmen atur cara di bawah menunjukkan contoh perbezaan pengisytiharan pemboleh ubah mudah dengan pengisytiharan tatasusunan:

```
public static void main(String [ ] args)  
{  
int [ ] markah = { 56 , 78 , 34 } ; // tatasusunan  
}
```



Uji Minda

Apakah yang terjadi sekiranya pemboleh ubah tidak diisyiharkan terlebih dahulu?





Uji Minda

Kelas tatususunan terdiri daripada pelbagai *static methods* yang bertujuan untuk mencari tatususunan, membandingkan tatususunan dan mengisi elemen tatususunan. Apakah kelas tatususunan tersebut? Cuba anda dan rakan-rakan berusaha untuk mencari jawapannya. Selamat Mencuba!

Semua data markah disimpan dalam satu nama pemboleh ubah sahaja, iaitu pemboleh ubah `markah`. Jika berlaku penambahan data markah, guru hanya perlu menambah nilai data tanpa mengisytiharkan nama pemboleh ubah yang baharu. Data disimpan berjujukan di dalam ruang memori.

Hal ini berbeza dengan pengisytiharan tanpa menggunakan tatususunan. Setiap data akan disimpan dalam pemboleh ubah yang berbeza dan ditempatkan di dalam ruang memori pada lokasi yang berbeza seperti yang ditunjukkan dalam rajah di bawah:

```
public static void main(String [] args)
{
    int markah1 = 56, markah2 = 78, markah3 = 34;
}
```

Di dalam memori komputer, maklumat ini disimpan seperti yang berikut:

markah1	markah2	markah3
56	78	34

Data disimpan dalam alamat yang berbeza di dalam memori. Apakah yang berlaku jika guru ingin menambahkan data markah? Adakah guru perlu mengisytiharkan pemboleh ubah tambahan yang berbeza bagi memegang nilai-nilai markah tersebut? Bayangkan jika data yang hendak ditambah itu banyak dan data akan disimpan secara rawak di mana-mana lokasi dan tidak tersusun. Hal ini akan menyebabkan proses mencapai data dari memori untuk diproses bagi menghasilkan maklumat akan menjadi lambat kerana data ditempatkan pada ruang alamat yang berbeza di dalam ingatan.

Atur cara yang Menggunakan Tatususunan

Elemen boleh dipanggil menggunakan indeks tatususunan. Sebagai contoh, diisytiharkan tatususunan berikut:

```
int [] senaraiMarkah = {86,78,80};
```

Tatususunan di atas masing-masing mempunyai tiga elemen dengan indeks 0, 1 dan 2. Oleh itu, gunakan sintaks berikut untuk memanggil elemen-elemen tersebut.

```
nama_tatususunan[indeks]
```

Dalam contoh `senaraiMarkah`, setiap elemen dipanggil dengan kod-kod atur cara yang berikut:

- `senaraiMarkah[0]` akan memberikan nilai 86
- `senaraiMarkah[1]` akan memberikan nilai 78
- `senaraiMarkah[2]` akan memberikan nilai 80

Contoh 59 menerangkan tatasusunan hariMinggu. Tatasusunan ini mengandungi data jenis rentetan (*String*). Elemen-elemen dalam tatasusunan ialah nama hari dalam seminggu. Perhatikan bahawa susunan elemen adalah penting kerana mengikut turutan indeks. Masih ingatkah nombor indeks yang pertama dalam tatasusunan?

CONTOH 59

Diberi tatasusunan hariMinggu seperti yang berikut:

```
String [] HariMinggu = {"Isnin", "Selasa", "Rabu", "Khamis", "Jumaat", "Sabtu",  
"Ahad"};
```

- (a) Tuliskan kod Java untuk mencetak elemen "Jumaat".
- (b) Apakah saiz tatasusunan hariMinggu.?
- (c) Apakah nilai yang dicetak oleh kod berikut?
 - (i) `System.out.println(hariMinggu[0])`
 - (ii) `System.out.println(hariMinggu[3])`
 - (iii) `System.out.println(hariMinggu[6])`
 - (iv) `System.out.println(hariMinggu[4])`
- (d) Mengapakah kod-kod berikut tidak boleh digunakan?
 - (i) `System.out.println(hariMinggu[7])`
 - (ii) `System.out.println(hariMinggu[10])`
 - (iii) `System.out.println(hariMinggu["3"])`
 - (iv) `System.out.println(hariMinggu[Selasa])`
- (e) Lukis rajah blok memori untuk tatasusunan di atas. Pastikan lukisan mengandungi nama tatasusunan, indeks dan elemen-elemen.

Penyelesaian:

- (a) `System.out.println(hariMinggu[4]);`
- (b) 7
- (c) (i) Isnin (iii) Ahad
(ii) Khamis (iv) Jumaat
- (d) (i) Ralat masa larian hariMinggu[7] di luar batasan tatasusunan.
(ii) Ralat masa larian hariMinggu[10] di luar batasan tatasusunan.
(iii) Ralat sintaks. Panggilan elemen tidak dibenarkan kerana indeks bukan integer tetapi *String*.
(iv) Ralat sintaks. Panggilan elemen tidak dibenarkan kerana tidak menggunakan nombor indeks.
- (e) Elemen-elemen dalam tatasusunan sentiasa diletakkan sebelah-menyebelah dalam blok memori yang sama. Nama tatasusunan merujuk blok memori. Dalam contoh di atas, nama blok ialah hariMinggu. Dalam blok memori tatasusunan, susunan elemen adalah dalam turutan berjujukan. Setiap elemen dirujuk menggunakan nama blok diikuti nombor indeks.

	[0]	[1]	[2]	[3]	[4]	[5]	[6]
hariMinggu	"Isnin"	"Selasa"	"Rabu"	"Khamis"	"Jumaat"	"Sabtu"	"Ahad"

CONTOH (60)

Diberikan kod atur cara Java yang penuh.

```
package perisianSaya;
public class MyClass {
    public static void main(String[] args){
        String [] senaraiNama = new String[4];
        senaraiNama[0] = "Adam";
        senaraiNama[1] = "Alia";
        senaraiNama[2] = "Wong";
        senaraiNama[3] = "Devi";

        int [] senaraiUmur = {16, 17, 16, 17};

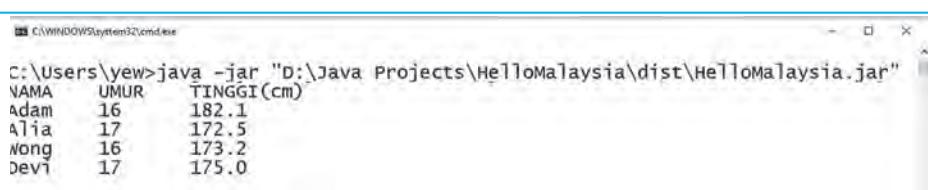
        double [] senaraiTinggi = {182.1,172.5,173.2,175.0};
        System.out.println("NAMA\tUMUR\tTINGGI(cm)");
        for(int i=0;i<4;i++){
            System.out.print(senaraiNama[i] + "\t");
            System.out.print(senaraiUmur[i] + "\t");
            System.out.print(senaraiTinggi[i]);
            System.out.println();
        }
    }
}
```

- (a) Senaraikan tiga pemboleh ubah tatasusunan dalam atur cara di atas.
- (b) Tulis elemen bagi kod berikut:
 - (i) senaraiNama[2]
 - (ii) senaraiUmur[1]
 - (iii) senaraiTinggi[3]
- (c) Tentukan output bagi atur cara di atas.

Penyelesaian:

- (a) senaraiNama, senaraiUmur, senaraiTinggi
- (b) (i) Wong
 - (ii) 17
 - (iii) 175.0

(c)



C:\Windows\system32\cmd.exe

```
C:\Users\yew>java -jar "D:\Java Projects\HelloMalaysia\dist\HelloMalaysia.jar"
NAMA      UMUR      TINGGI(cm)
Adam      16        182.1
Alia      17        172.5
Wong      16        173.2
Devi      17        175.0
```



Ambil sekeping kertas A4. Lukis blok memori untuk pernyataan-pernyataan yang berikut:

- String negaraku = "Malaysia";
- String [] negaraASEAN = {"Brunei", "Kemboja", "Filipina", "Indonesia", "Laos", "Malaysia", "Singapura", "Thailand", "Vietnam"};
- String [] atlet = new String [6];
atlit[0] = "Misbun";
atlit[2] = "Chong Wei";
atlit[4] = "Pandeleta";
atlit[5] = "Rajagopal";

Setelah blok memori anda siap dilukis, bandingkan jawapan anda dengan jawapan di bawah.

Jawapan:

(a) negaraku "Malaysia"

(b)

[0]	[1]	[2]	[3]	[4]	[5]	[6]	[7]	[8]	
negaraASEAN	"Brunei"	"Kemboja"	"Filipina"	"Indonesia"	"Laos"	"Malaysia"	"Singapura"	"Thailand"	"Vietnam"

(c)

[0]	[1]	[2]	[3]	[4]	[5]	
atlet	"Misbun"	Null	"Chong Wei"	Null	"Pandeleta"	"Rajagopal"

1.6.2 Menggunakan Subatur Cara, Memahami Konsep Menghantar Parameter ke Subatur Cara dan Mengembalikan Data

Dalam kehidupan harian, tanggungjawab yang berat biasanya dibahagikan kepada tugas-tugas untuk dilaksanakan secara berasingan. Tugasan kecil lebih mudah untuk dipantau dan dilaksanakan. Sebagai contoh, tanggungjawab mendidik murid merupakan tugasan pihak sekolah. Pengetua akan membahagikan tugas mengajar kepada guru-guru di bawah pengurusan beliau.



Uji Minda

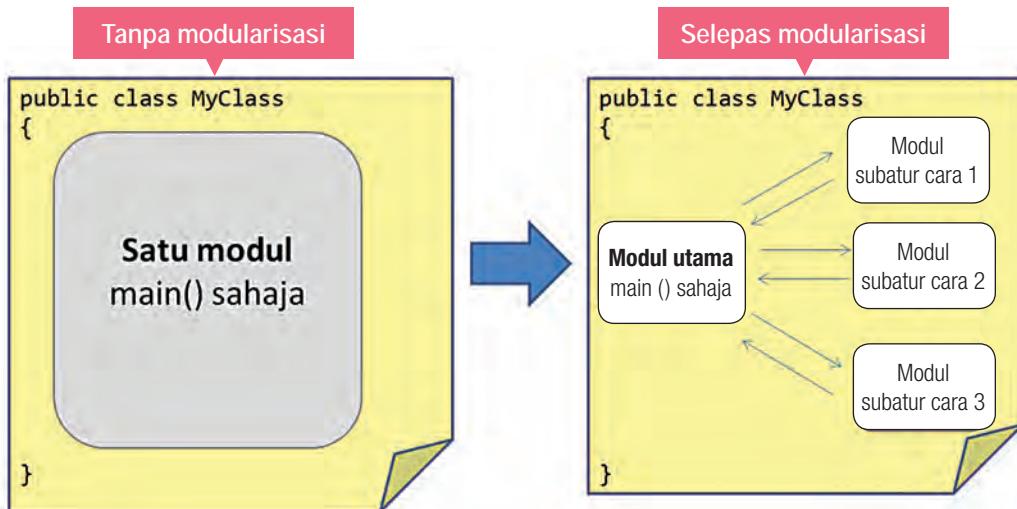
Selain daripada tatasusunan yang telah kita pelajari, terdapat juga beberapa jenis tatasusunan yang lain. Apakah jenis tatasusunan tersebut? Nyatakan perbezaan yang boleh anda dapatkan bagi tatasusunan tersebut dengan tatasusunan yang telah anda pelajari.





Rajah 1.52 Subatur cara ialah kaedah modularisasi, sama seperti pembahagian tugas dalam kalangan guru

Demikian halnya dengan atur cara komputer. Fail kod komputer yang panjang, mungkin puluhan ke ribuan baris, adalah sukar ditulis, dibaca, diulang kaji atau dikemas kini. Oleh itu, baris-baris kod komputer yang berkait boleh dihimpunkan dalam satu modul. Dengan itu, kod-kod komputer yang panjang dapat dibahagi-bahagikan kepada modul-modul. Setiap modul adalah lebih pendek dan mengkhusus kepada tujuan tertentu.



Rajah 1.53 Modularisasi kod komputer

Modul-modul ini dipanggil subatur cara. Subatur cara ialah struktur untuk himpunan kod komputer. Antara senarai pernyataan yang terdapat di dalam suatu subatur cara ialah,

- menerima input data
- memproses data
- memaparkan maklumat



Rajah 1.54 Kelebihan menggunakan modul atau subatur cara

Sebuah subatur cara mempunyai kepala dan badan. Contoh sebuah aratur cara adalah seperti berikut.

```
Pengepala (Header) ← static void subAtur01()
                    {
Badan (Body)   ←     System.out.println("Hello dunia.");
                    }
```



Tahukah Anda?
Subatur cara `main()` digunakan untuk melaksanakan semua jenis aratur cara Java.

Pengepala subatur cara mempunyai komponen-komponen yang berikut:

```
static void subAtur01 ()
    ↓    ↓    ↓    ↓
  Kata  Jenis  Nama  Bekas
  kunci data subatur parameter
  pulangan cara
```

Kata kunci khas	Kata kunci <i>static</i> diletakkan di hadapan nama subatur cara. Tanpa kata kunci ini, subatur cara tidak dapat digunakan secara langsung. Tanpa <i>static</i> , subatur cara memerlukan penghasilan objek sebelum data digunakan.
Jenis data pulangan	Subatur cara biasanya akan memulangkan hasil setelah badan subatur cara selesai memproses data. Jenis data pulangan di "pengepala" bergantung kepada jenis data yang ingin dipulangkan oleh "badan". Ini termasuk <i>int</i> , <i>double</i> , <i>String</i> dan <i>char</i> . Jika tidak ada keperluan memulangkan data, gunakan kata kunci <i>void</i> .
Nama subatur cara	Nama ini diberikan oleh pengatur cara. Sebarang nama yang sesuai boleh diberikan. Nama mestilah bermula dengan huruf (biasanya huruf kecil) dan boleh mengandungi angka tetapi bukan simbol.
Bekas parameter	Bekas parameter dikepil oleh tanda kurung "(" dan ")". Jika bekas parameter adalah kosong, simbol ")" digunakan. Jika bekas menerima parameter, maka jenis data dan nama parameter akan dikepilkann. Contoh: (<i>int kuantiti</i>) menerima parameter integer bernama <i>kuantiti</i> . Parameter ialah pemboleh ubah untuk menyimpan nilai hantaran kepada subatur cara.

Badan subatur cara mengandungi kod komputer. Badan subatur cara bermula dengan simbol kurung dakap pembuka "{" dan diakhiri simbol kurung dakap penutup "}". Penyataan komputer adalah sebaris kod komputer yang diakhiri simbol koma bertindih ";". Semua penyataan komputer mesti terkandung dalam subatur cara. Penyataan-penyataan komputer ialah algoritma komputer yang telah diterjemahkan kepada bahasa pengaturcaraan seperti *Java*. Penyataan-penyataan ini akan dilaksanakan oleh komputer dari baris awal hingga ke baris akhir dalam turutan atas dan bawah (*top-down*).

Subatur Cara `main()`

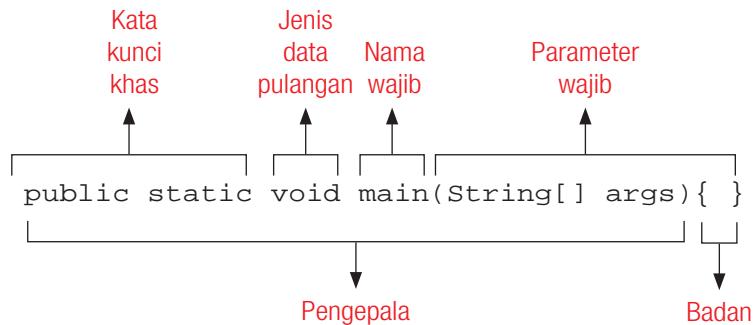
Subatur cara `main()` ialah sejenis subatur cara yang istimewa. Hal ini demikian kerana:

- subatur cara wajib dengan nama `main()`.
- boleh wujud tanpa subatur cara yang lain.
- mengandungi baris pertama penyataan yang mesti dilaksanakan komputer.
- mengandungi baris terakhir penyataan yang mesti dilaksanakan komputer.

Subatur cara `main()` mempunyai sintaks yang serupa dengan sintaks umum subatur cara tetapi dengan reka bentuk kepala yang unik:



`main()` method
<http://goo.gl/1dVWYl>



Rajah 1.55 Subatur cara `main()`

Dalam Java khususnya, pengepala subatur cara `main()` mesti mempunyai ciri-ciri yang berikut:

Kata kunci khas	Mempunyai kata kunci khas <code>public</code> dan <code>static</code> :
	<ul style="list-style-type: none">• <code>public</code> ialah skop akses subatur cara. <i>Public</i> membolehkan subatur cara ini diakses dari mana-mana kod projek.• Semua subatur cara mempunyai <code>static</code> supaya dapat digunakan secara langsung tanpa objek
Jenis data pulangan	Subatur cara <code>main</code> tidak memulangkan data, maka gunakan <code>void</code> .
Nama subatur cara	Nama wajib ialah <code>main</code> .
Bekas parameter	Bekas parameter mesti mengandungi parameter tatasusunan <code>String</code> dengan nama "args".

Pengepala subatur cara `main()` jarang diubah. Kekalkan sintaks subatur cara `main()` seperti yang ditunjukkan dalam Rajah 1.55 untuk mengelakkan ralat sintaks. Badan subatur cara `main()` mengandungi kod-kod komputer.

Penyataan-penyataan dalam subatur cara `main()` akan menentukan sifat atur cara. Oleh itu, penyataan-penyataan ini seharusnya ditulis berasaskan algoritma yang telah diuji. Contoh yang berikut ialah atur cara ringkas yang memaparkan rentetan "Hello Malaysia.".

Contoh:



Ingatlah bahawa penyataan-penyataan kod komputer tidak boleh ditulis di mana-mana tempat melainkan di dalam subatur cara sahaja. Jika tidak, kod tersebut akan menimbulkan ralat apabila dikompil.

Subatur cara `main()`

Output

```
1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args){
5         System.out.println("Hello Malaysia.");
6     }
7
8 }
9
10
```

C:\WINDOWS\system32\cmd.exe

D:\Java Projects\HelloMalaysia\dist>java -jar HelloMalaysia.jar

Hello Malaysia.

Rajah 1.56 Contoh subatur cara `Main`

Dalam rajah di atas, subatur cara `main()` terletak pada baris empat hingga enam. Subatur cara mesti berada dalam badan *class*, *MyClass*. Badan *MyClass* adalah dari baris tiga hingga lapan.

Tahukah Anda ?

Ralat sintaks ialah kesilapan menggunakan simbol kurung dakap. Murid sama ada terlupa meletakkan simbol atau terlebih meletakkan simbol. Pastikan pasangan adalah betul. Setiap simbol kurung dakap pembuka "{" mesti mempunyai simbol penutup "}". Untuk mencegah kecualian, buat indentasi yang sama bagi pasangan yang sama. Indentasi adalah untuk menambahkan ruang kosong sebelum simbol menggunakan kekunci "Tab".



Defining methods
<http://goo.gl/S1XKyT>

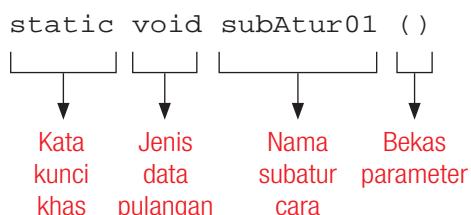
Baris pertama dalam `main()` adalah pada baris lima. Penyataan kod `Java System.out.println("Hello Malaysia.");` akan mencetak "Hello Malaysia." pada skrin output. Baris enam ialah simbol kurung dakap penutup ")" yang menandakan pengakhiran subatur cara `main()` program Java.

Perhatian!

- Simbol kurung dakap penutup ")" pada baris enam ialah pasangan kepada simbol kurung dakap pembuka "{" pada baris empat. Pasangan ini ialah penanda kepada badan subatur cara `main()`.
- Simbol kurung dakap penutup ")" pada baris lapan, ialah pasangan kepada simbol kurung dakap pembuka "{" pada baris dua. Pasangan ini ialah penanda kepada badan class `MyClass`.

Subatur Cara Lain

Selain subatur cara `main()`, pengatur cara boleh menulis subatur cara yang lain. Aktiviti ini disebut sebagai pentakrifan subatur cara. Subatur cara lain adalah serupa dengan `main()` tetapi lebih ringkas:



Pengatur cara mentakrifkan subatur cara baharu dengan merujuk sintaks subatur cara di atas. Contoh suatu subatur cara ringkas ialah subatur cara yang memaparkan teks "Hello dunia.". Lihat contoh yang berikut:

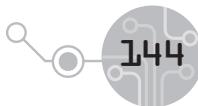
Contoh:

```
static void hello()  
{  
System.out.println("Hello dunia.");  
}
```

Dalam fail kod sumber, subatur cara baharu ini terletak diluar subatur cara `main()`. Seperti mana subatur cara `main()`, subatur cara lain juga mesti berada dalam badan *class*. Lihat rajah 1.57.



Java Methods Calling
<http://goo.gl/hKk5lw>



```

1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args) {
5
6         }
7
8         static void hello()
9         {
10             System.out.println("Hello dunia.");
11         }
12
13     }

```



Uji Minda

Berdasarkan contoh kedua, cuba bina subatur cara bernama masukUmur dan meminta pengguna untuk memasukkan input umur murid-murid di dalam kelas anda.

Rajah 1.57 Subatur cara lain yang dalam badan *class*

Contoh kedua ialah subatur cara bernama masukMarkah. Subatur cara ini meminta pengguna input markah bagi mata pelajaran Sains Komputer. Markah tersebut diumpukan kepada pemboleh ubah bernama markahKomputer.

Contoh:

```

1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args) {
5
6         }
7
8         static void masukMarkah()
9         {
10             double markahKomputer;
11             System.out.println("Input markah Sains Komputer:");
12             java.util.Scanner scanner = new java.util.Scanner(System.in);
13             markahKomputer = scanner.nextDouble();
14             System.out.println("Markah Sains Komputer adalah: " + markahKomputer);
15         }
16
17     }

```

Rajah 1.58 Subatur cara masukMarkah

Dalam contoh di atas, subatur cara masukMarkah terletak pada baris kod 9 hingga 15. Dalam badan kod, terdapat penggunaan prosedur `java.util.Scanner` untuk membaca input pengguna. Untuk menggunakannya, isytiharkan pemboleh ubah `scanner` sebagai jenis `java.util.Scanner` dan umpukan dengan objek `new java.util.Scanner(System.in)`. Biasanya, pengumpukan objek `java.util.Scanner(System.in)` cuma perlu dibuat satu kali. Pada baris berikutnya, baris 13, pemboleh ubah `scanner` digunakan untuk membaca input jenis `double` daripada skrin menggunakan prosedur `nextDouble()`.



Selain membaca data jenis `double`, `java.util.Scanner` mempunyai prosedur-prosedur lain seperti

- `nextInt()` untuk membaca input jenis `int`,
- `next()` untuk membaca input satu perkataan,
- `nextLine()` untuk membaca input satu baris teks.



Tahukah Anda ?

Ingat lima perkara ini!

- (i) Apabila subatur `main()` dipanggil, semua pernyataan yang merupakan sebahagian daripada subatur akan dilaksanakan.
- (ii) Perkataan `public` sebelum nama subatur boleh dipanggil dari pelbagai lokasi yang terdiri daripada *class* yang lain.
- (iii) `static` bermaksud subatur ialah kepunyaan *class*, bukannya objek.
- (iv) Perkataan terakhir sebelum subatur ialah `void`.
- (v) Apabila memanggil subatur yang memulangkan sesuatu, anda boleh menggunakan apa yang telah dipulangkan.

Memanggil Subatur `main()`

Subatur cara boleh menggunakan subatur cara lain. Tujuannya adalah supaya kod penyataan-penyataan dalam subatur lain turut dilaksanakan.

Umumnya, hubungan dua subatur cara adalah pemanggil dan dipanggil. Pemanggil memanggil nama subatur cara dipanggil dalam badan subatur cara pemanggil.

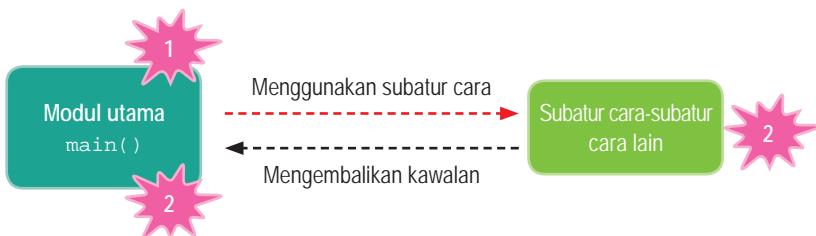


Rajah 1.59 Subatur cara boleh memanggil ataupun dipanggil

Pentakrifan atau pendefinisan subatur cara perlu dibuat sebelum subatur cara dapat dipanggil daripada `main()` ataupun mana-mana subatur cara lain, yakni pemanggil.

Memanggil Subatur daripada `main()`

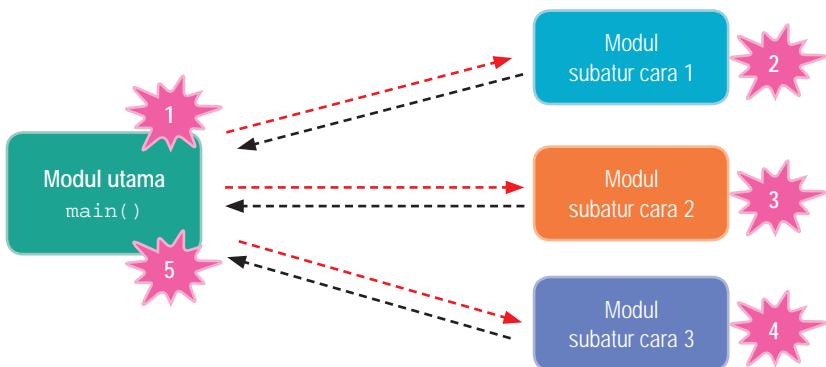
Kebiasaannya, subatur cara `main()` menggunakan subatur cara-subatur cara lain. Lihat Rajah 1.60(a).



Rajah 1.60(a) Perhubungan antara subatur cara `main()` dengan subatur cara lain

Sebagai modul utama, kawalan mutlak berada pada `main()`. Apabila `main()` memerlukan bantuan subatur cara lain untuk melakukan proses tertentu, kawalan dipindahkan kepada subatur cara tersebut. Setelah subatur cara tersebut selesai, kawalan dikembalikan kepada subatur cara `main()`.

Apabila terdapat lebih daripada dua subatur cara, yang manakah akan dilaksanakan terlebih dahulu? Bagaimanakah subatur cara ini disusun dan diselaras?



Creating objects and calling methods
<http://goo.gl/502bir>

Penunjuk
 - - → Menggunakan subatur cara
 ← - Mengembalikan kawalan

Rajah 1.60(b) Sub atur cara `main()` sebagai modul utama

Dalam rajah 1.60(b), contoh menunjukkan modul utama `main()` menggunakan modul subatur cara 1 diikuti subatur cara 2 dan subatur cara 3. Akhirnya, kawalan dipulangkan kembali kepada modul utama `main()`. **Atur cara bermula dengan `main()` dan diakhiri `main()`.**

Contoh yang berikut menunjukkan bagaimana subatur cara `hello()` dipanggil daripada `main()`. Panggilan dibuat dalam prosedur `main()` menggunakan nama dipanggil `hello()`. Cuba kenal pasti subatur cara `main()`, definisi subatur cara `hello()` dan panggilan subatur cara `hello()`.

```

1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args){
5         hello(); // Panggil hello()
6     }
7
8     static void hello(){
9         System.out.println("Hello dunia.");
10    }
11 }

```

Pemanggil
main()

Panggil
hello()

Definisi
hello()



Method calling
<http://goo.gl/yxNojd>

Rajah 1.61(a) Panggilan `hello()` dan definisi `hello()`



Calling Java Methods
<http://goo.gl/GUP954>

Dalam hal ini, panggilan subatur cara `hello()` dibuat dari badan pemanggil `main()`. Seterusnya, kawalan dipindahkan kepada definisi subatur cara `hello()` untuk melaksanakan pernyataan-pernyataan dalam badannya.

```

1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args) {
5         hello(); // 1. Panggil
6     }
7
8     static void hello() { // 2. Laksana
9         System.out.println("Hello dunia.");
10    }
11 }
12
13 }
```

The diagram illustrates the execution flow between two methods. It starts at line 5 where the `hello();` call is made from `main()`. This is labeled '1. Panggil' (Call). A red dashed box encloses the `hello()` method definition (lines 8-11). An arrow points from the `hello();` call to the start of the `hello()` method. Inside the `hello()` method, a green arrow points from the opening brace to the `System.out.println` statement, which is labeled '2. Laksana' (Execute). After the `hello()` method completes, a red curved arrow labeled '3. Pulang' (Return) points back to the closing brace of the `main()` method.

Rajah 1.61(b) Memanggil `hello()` daripada `main()`

Tahukah Anda?

Apabila memanggil subatur yang mengambil satu atau lebih argumen, pastikan nilai yang sesuai disediakan. Hal ini adalah kerana argumen sering diperlukan dan subatur pemanggil mesti mempunyai nilai yang betul apabila memanggil suatu subatur. Penerangan lebih lanjut mengenai argumen ialah pada muka surat yang seterusnya.

Satu lagi contoh menunjukkan subatur cara `main()` memanggil dua subatur cara yang lain.

```

1 package perisianSaya;
2 public class MyClass {
3
4     static java.util.Scanner scanner = new java.util.Scanner(System.in);
5
6     public static void main(String[] args) {
7
8         mintaNama();
9         mintaMarkah();
10    }
11
12    public static void mintaNama() {
13        String nama;
14        System.out.println("Masukkan nama: ");
15        nama = scanner.nextLine();
16        System.out.println("Terima kasih " + nama);
17    }
18
19    public static void mintaMarkah() {
20        int markah;
21        System.out.println("Masukkan markah: ");
22        markah = scanner.nextInt();
23        System.out.println("Markah anda: " + markah);
24    }
25
26 }
27 }
```

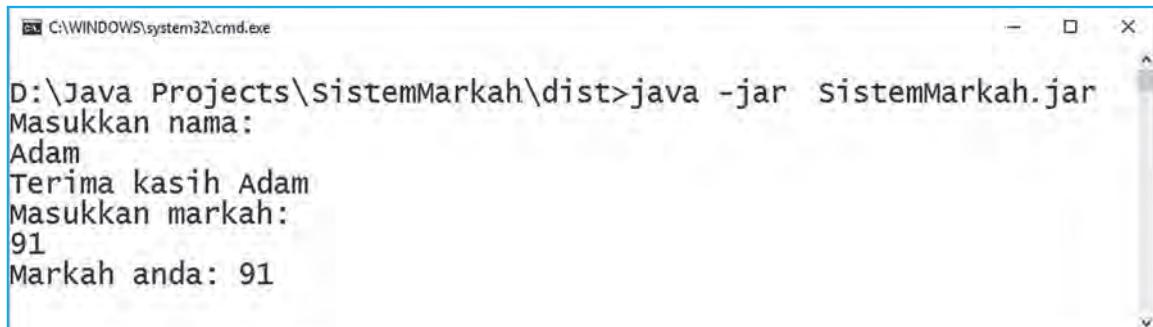
The diagram shows the execution flow starting from line 6 of the `main()` method. A red curved arrow labeled '1' points to the `mintaNama();` call. Another red curved arrow labeled '2' points to the `mintaMarkah();` call. Both calls lead to their respective method definitions (lines 12-17 and 20-24). A green arrow points from the closing brace of the `main()` method back to the end of the `mintaMarkah()` method.

Rajah 1.62(a) Memanggil dua subatur cara daripada `main()`

Rajah 1.62(a) mentakrifkan dua definisi subatur cara `mintaNama()` dan `mintaMarkah()`. Setiap satu menerima input dan memaparkan output untuk nama dan markah masing-masing. Perhatikan di baris 4, pemboleh ubah `scanner` ialah objek `new java.util.Scanner(System.in)` yang digunakan untuk membaca teks di baris 17 dan membaca integer di baris 24 menggunakan kaedah fungsi `nextLine()` dan `nextInt()` masing-masing. Perhatikan pemboleh ubah objek `scanner` terletak di dalam badan `class` tetapi tidak terletak di bawah mana-mana subatur cara. Ini kerana pemboleh ubah objek `scanner` dikongsi oleh semua subatur cara dalam `class` di atas. Apabila dikompil dan dijalankan, output atur cara di atas adalah seperti yang ditunjukkan. Perhatikan susunan output bergantung kepada susunan panggilan subatur cara dalam `main()`.

Tahukah Anda ?

Penggunaan kata kunci khas `public` ialah skop akses. Bagi subatur cara-subatur cara lain dalam contoh yang ditunjukkan, penggunaan `public` boleh diabaikan tanpa apa-apa kesan.

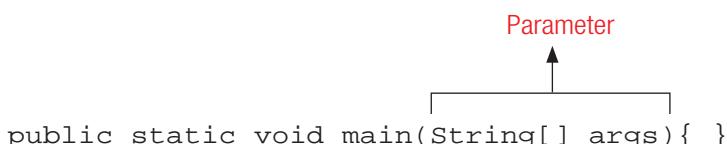


```
C:\WINDOWS\system32\cmd.exe
D:\Java Projects\sistemMarkah\dist>java -jar SistemMarkah.jar
Masukkan nama:
Adam
Terima kasih Adam
Masukkan markah:
91
Markah anda: 91
```

Rajah 1.62(b) Output

Parameter

Sebahagian subatur cara boleh menerima parameter. Ingatkah lagi parameter wajib "args" dalam subatur cara `main()`?



Apakah itu parameter? **Parameter** ataupun argumen ialah pemboleh ubah yang membolehkan subatur cara menerima nilai daripada pemanggil. Dengan demikian, subatur cara-subatur cara masih dapat berkongsi nilai-nilai pemboleh ubah melalui parameter.

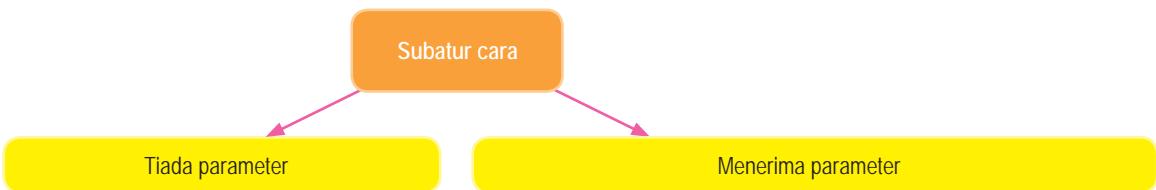
Ungkapan parameter rasmi (*formal parameter*) digunakan untuk merujuk parameter bagi subatur cara.

Ungkapan parameter sebenar (*actual parameter*) pula digunakan untuk merujuk pemboleh ubah di dalam subatur cara pemanggil.

Tahukah Anda ?

Bahasa pengaturcaraan *Java* tidak seperti bahasa pengaturcaraan yang lain. Hal ini adalah kerana *Java* tidak mempunyai mekanisme untuk mengubah nilai parameter sebenar.





<pre>static void subAtur01 (){} static void subAtur02 (){} static void subAtur03 (){}</pre>	<pre>static void subAtur01 (int x){} static void subAtur02 (int x, double y){} static void subAtur03 (int [] x, String z){}</pre>
---	---

Rajah 1.63 Sesetengah subatur cara boleh menerima parameter

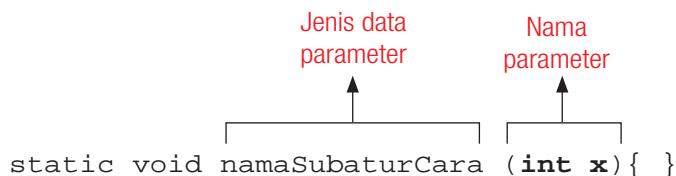


Berdasarkan Contoh 1 dan 2, labelkan pengisyiharan parameter bagi soalan yang berikut.

(a) static void kuasaTiga (int no) {
double jawapan = nom * nom * nom;
System.out.print(jawapan);
}

(b) static void cariJumlah (int a, int b, int c) {
int jawapan = a + b + c;
System.out.print(jawapan);
}

Penggunaan parameter perlu diisytiharkan sewaktu pentakrifan subatur cara-subatur cara:



Jika parameter diperlukan, parameter perlu diisytiharkan dalam kurungan bekas parameter dalam subatur cara. Pengisyiharan parameter adalah sama seperti pengisyiharan pemboleh ubah. Tiada had untuk bilangan parameter dan juga turutan parameter bergantung kepada pengatur cara.

Contoh 1:

Pengisyiharan parameter nom

```
static void kuasaDua(int nom){  
double jawapan = nom * nom;  
System.out.print(jawapan);  
}
```

Contoh 2:

Pengisyiharan parameter x, y dan z

```
static void cariJumlah(int x, int y, int z){  
int jawapan = x + y + z;  
System.out.print(jawapan);  
}
```

Namun begitu, panggilan subatur cara mestilah menggunakan nilai-nilai yang serasi dengan pengisytiharan parameter dalam subatur cara. Susunan nilai adalah penting.

Contoh 3:

```
public static void main(String [] args){  
    cariJumlah(1,2,7);  
}  
  
static void cariJumlah(int x, int y, int z){  
    int jawapan = x + y + z;  
    System.out.print(jawapan);  
}
```

Pengumpukan nilai parameter kepada parameter

Dalam contoh 3, panggilan memberikan tiga nombor integer yang serasi dengan definisi cariJumlah. Setiap nombor integer 1, 2 dan 7 akan diumpukan kepada parameter x, y dan z mengikut urutan.

Contoh 4:

```
public static void main(String [] args){  
    paparHarga("telefon pintar",1500.0);  
}  
  
static void paparHarga(String item, double h){  
    System.out.print("Harga " + item + " adalah " +  
h);  
}
```

Pengumpukan nilai parameter kepada parameter

Dalam contoh di atas, senarai parameter mesti berada dalam susunan yang betul. Ini bermakna `paparHarga(1500.0, "telefon pintar")`; adalah salah kerana umpukan nilai kepada jenis data yang tidak betul. Bilangan nilai yang dihantar mestilah tepat, iaitu tidak lebih ataupun kurang. Penggunaan `paparHarga("telefon pintar")`; adalah salah kerana tiada maklumat harga.



Berdasarkan Contoh 3 dan 4, bina suatu sintaks yang bersesuaian untuk mengisytiharkan parameter. Pastikan sintaks tersebut mempunyai susunan nilai yang betul dan terdapat pengumpukan nilai parameter kepada parameter.

Tahukah Anda?

Nama yang diberikan pada parameter mestilah nama yang unik mengikut parameter tersebut.

Nama parameter tidak boleh sama dengan nama parameter yang lain dalam subatur yang sama dan nama boleh ubah setempat di dalam subatur tersebut.



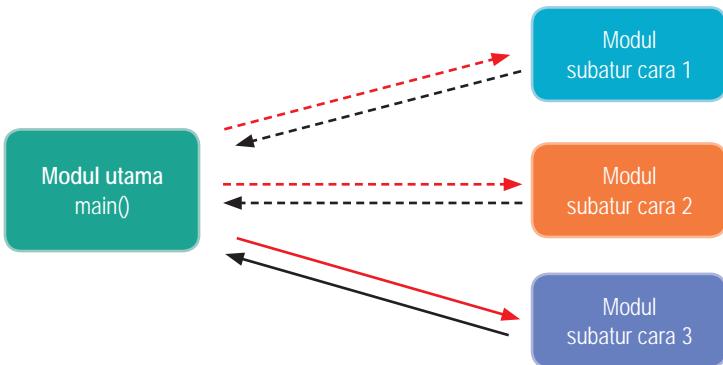
Tahukah Anda?

Nota ringkas untuk mengingati cara mengembalikan data.

- (a) Menentukan sama ada jenis data subatur cara yang dipulangkan ialah jenis data primitif atau nama *class*.
- (b) Jenis data boleh menjadi *void* yang bermaksud tidak memulangkan sebarang nilai.
- (c) Subatur cara boleh memulangkan data dengan menggunakan perkataan "return".

Mengembalikan data

Semua subatur cara akan mengembalikan kawalan kepada pemanggil. Sesetengah subatur cara bukan sahaja mengembalikan kawalan tetapi juga data. Dalam rajah yang berikut, contoh subatur cara 3 mengembalikan data kepada pemanggil `main()`. Bagi yang memulangkan kawalan sahaja, seperti subatur cara 1 dan 2, kata kunci *void* digunakan sewaktu pengisytiharan subatur cara.



Penunjuk

- - - → Menggunakan subatur cara
- Menggunakan subatur cara DAN menghantar parameter
- ← - - - Mengembalikan kawalan
- ← - - Mengembalikan kawalan DAN parameter

Rajah 1.64 Jenis-jenis parameter dan pulangan daripada subatur cara dipanggil

Sekiranya subatur cara mengembalikan data, baris akhir dalam badan subatur cara mempunyai pernyataan *return* dan data yang ingin dipulangkan kepada pemanggil. Nama subatur cara tidak mempunyai *void* sebagai jenis data pulangan. Sebaliknya, jenis data yang bersesuaian dengan data dipulang diisytiharkan dalam sintaks.

Data boleh dipulangkan sebagai hasil ungkapan, nilai dalam pemboleh ubah, nilai pemalar ataupun nilai data itu sendiri. Setiap subatur cara cuma boleh mengembalikan satu jenis data sahaja. Rujuk Jadual 1.18.

Jadual 1.18 Subatur cara dan penjelasannya

Definisi subatur cara	Penjelasan
static void subAturCara(){}	Tidak memulangkan data. Cuma kawalan dipulangkan
static int subAturCara(){}	Memulangkan data jenis integer
static double subAturCara(){}	Memulangkan data jenis <i>double</i>
static String subAturCara(){}	Memulangkan data jenis <i>String</i>

Nama subatur cara yang memulangkan data biasanya diberikan *prefix get*. Sebagai contoh, *getNama*, *getAlamat* dan *getTelefon*. Contoh berikut menunjukkan subatur cara *main()* yang menggunakan subatur cara untuk mengembalikan data. Lihat baris kod 9, 10 dan 11 untuk panggilan. Baris-baris kod untuk definisi subatur cara *getNama*, *getAlamat* dan *getTelefon* adalah masing-masing 20-23, 25-28 dan 30-33.

```

1 package perisianSaya;
2 public class MyClass {
3
4     static java.util.Scanner scanner = new java.util.Scanner(System.in);
5
6     public static void main(String[] args){
7         String nama, alamat, telefon;
8
9         nama = getNama();
10        alamat = getAlamat(nama);
11        telefon = getTelefon(nama);
12
13        System.out.println();
14        System.out.println("Terima kasih. Sila semak.");
15        System.out.println("Nama: " + nama);
16        System.out.println("Alamat: " + alamat);
17        System.out.println("Telefon: " + telefon);
18    }
19
20    static String getNama(){
21        System.out.print("Masukkan nama: ");
22        return scanner.nextLine();
23    }
24
25    static String getAlamat(String n){
26        System.out.print(n + ", sila masukkan alamat: ");
27        return scanner.nextLine();
28    }
29
30    static String getTelefon(String n){
31        System.out.print(n + ", sila masukkan telefon: ");
32        return scanner.nextLine();
33    }
34 }
```

Tahukah Anda?

Terdapat dua perkara penting yang perlu diberikan perhatian ketika mengembalikan data iaitu,

- (a) Jenis data pulangan mestilah bersesuaian dengan jenis data yang ditentukan oleh subatur cara. Contohnya, apabila jenis data pulangan ialah daripada subatur cara integer, anda tidak boleh memulangkan *String*.
- (b) Parameter mestilah mengikut susunan dan diterima oleh subatur cara dengan susunan yang sama.

Rajah 1.65 Menggunakan subatur cara untuk mengembalikan data



1.6.3 Perbezaan antara Fungsi (Function) dan Prosedur (Procedure) pada Subatur cara

Dua jenis subatur cara ialah prosedur dan fungsi. Perbezaan dirumuskan seperti yang berikut:

Jadual 1.19 Perbandingan antara prosedur dan fungsi

Aspek perbandingan	Prosedur	Fungsi
Persamaan	Mengembalikan kawalan. Perbezaan: Tidak mengembalikan data. Jenis data pulangan dalam sintaks adalah <i>void</i> . Badan tidak diakhiri dengan pernyataan <i>return</i> .	Mengembalikan kawalan. Perbezaan: Mengembalikan data. Jenis data pulangan dalam sintaks adalah <i>int</i> , <i>double</i> , <i>char</i> , <i>String</i> , tatasusunan atau objek <i>Java</i> . Badan diakhiri dengan pernyataan <i>return</i> diikuti data yang dipulangkan.
Sintaks definisi	<pre>static void namaProsedur([jenisData namaParameter]){};</pre>	<pre>static jenisData namaFungsi([jenisData namaParameter]){};</pre>
Contoh definisi	<pre>static void cariJumlah(int x, int y) { int jawapan = x + y; System.out.print(jawapan); }</pre>	<pre>static int cariJumlah(int x, int y) { int jawapan = x + y; return jawapan; }</pre>
Sintaks panggilan	<code>namaProsedur([jenisData namaParameter]);</code>	<code>jenisData bolehUbah; bolehUbah = namaFungsi([jenisData namaParameter]);</code>
Contoh panggilan	<code>cariJumlah(5, 8);</code>	<code>int jumlah = cariJumlah(5, 8);</code>

Umumnya, sintaks fungsi adalah serupa dengan prosedur. Badan fungsi adalah seperti prosedur kecuali di baris akhirnya terdapat pernyataan *Return* untuk memulangkan data. Data boleh dipulangkan sebagai hasil ungkapan, nilai dalam boleh ubah, nilai pemalar ataupun nilai data itu sendiri.

Bilakah prosedur atau fungsi menjadi pilihan? Petunjuk berikut boleh digunakan untuk memilih fungsi:

Fungsi

- Data yang telah diproses perlu disimpan dalam badan pemanggil.
- Data yang telah diproses akan digunakan lagi oleh pemanggil. Data tersebut mungkin digunakan oleh subatur cara `main()` atau subatur cara-subatur cara lain.
- Subatur cara digunakan untuk meminta input.
- Subatur cara tidak memaparkan hasil atau menyimpan hasil di mana-mana.

Contoh 1:

```
static int mintaNombor() {  
    int nom;  
    java.util.Scanner sc;  
    sc = new java.util.Scanner(System.in);  
    nom = sc.nextInt();  
    return nom  
}
```

Contoh 2:

```
int jumlahNombor(int x, int y) {  
    int jumlah;  
    jumlah = x + y  
    return jumlah;  
}
```

Petunjuk berikut boleh digunakan untuk memilih prosedur iaitu:

Prosedur

- Subatur cara digunakan untuk membuat paparan sahaja, seperti mesej kepada pengguna.
- Hasil proses digunakan dalam subatur cara sekali sahaja dan tidak diperlukan lagi.



Kesilapan umum penggunaan fungsi adalah seperti yang berikut:

- Perlu diingatkan bahawa fungsi cuma memulangkan satu jenis data pada satu-satu masa. Jenis data yang dipulangkan mesti sepadan dengan jenis data pulangan yang telah diisyiharkan.

- Kesilapan umum bagi subatur cara jenis fungsi ialah kegagalan menyambut data yang dipulangkan.

Contoh 1:

```
static void hello(){  
    System.out.print("Hello dunia.");  
}
```

Tahukah Anda?

Fungsi utiliti dalam Java
Fungsi-fungsi utiliti adalah *built-in*, yakni sudah sedia ada dalam bahasa pengaturcaraan Java. Jadual 1.20 menunjukkan sebahagian fungsi utiliti yang lazim digunakan.

Contoh 2:

```
static void hello(String nama){  
    System.out.print("Hello " + nama);  
}
```

Contoh 3:

```
static void cariJumlah(int x, int y)  
{  
    int jawapan = x + y;  
    System.out.print(jawapan);  
}
```

Dalam jadual di bawah, fungsi `mintaNombor()` ialah memulangkan integer. Kaji kod definisi fungsi `mintaNombor()`.

```
static int mintaNombor(){ return 100; }
```

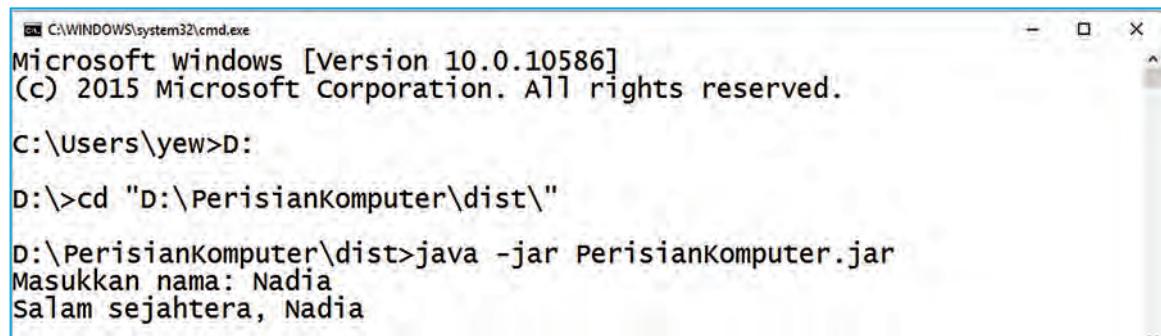
Integer tersebut perlu "disambut" dengan mengumpulkan kepada pemboleh ubah integer. Berdasarkan contoh di dalam jadual, `x` digunakan sebagai pemboleh ubah untuk menyimpan nilai yang dipulangkan.

Jadual 1.20 Panggilan fungsi cara yang betul dan salah

Betul	Salah
<pre>int x; x = mintaNombor() System.out.print(x);</pre>	<pre>int x; mintaNombor() System.out.print(x);</pre>

CONTOH (6)

Tulis satu atur cara PerisianKomputer yang menggunakan subatur cara fungsi `getNama()` untuk meminta nama pengguna. Nama tersebut tidak akan dipaparkan dalam fungsi. Sebaliknya, nilai itu akan dipulangkan kepada prosedur `main()` untuk dipaparkan. Sampel output adalah seperti berikut:



```
C:\WINDOWS\system32\cmd.exe  
Microsoft windows [Version 10.0.10586]  
(c) 2015 Microsoft Corporation. All rights reserved.  
C:\Users\yew>D:  
D:>>cd "D:\PerisianKomputer\dist\"  
D:\PerisianKomputer\dist>java -jar PerisianKomputer.jar  
Masukkan nama: Nadia  
Salam sejahtera, Nadia
```



Penyelesaian:

Untuk menghasilkan jawapan hasilkan projek dengan nama projek PerisianKomputer. Gunakan nama pakej perisianSaya. Gunakan nama class MyClass. Dalam MyClass, masukkan kod berikut:

```
package perisianSaya;
public class MyClass {

    static java.util.Scanner scanner = new java.util.Scanner(System.in);

    public static void main(String[] args){
        String x;
        x = getNama();
        System.out.println("Salam sejahtera, " + x);
    }

    static String getNama(){
        System.out.print("Masukkan nama: ");
        return scanner.nextLine();
    }
}
```

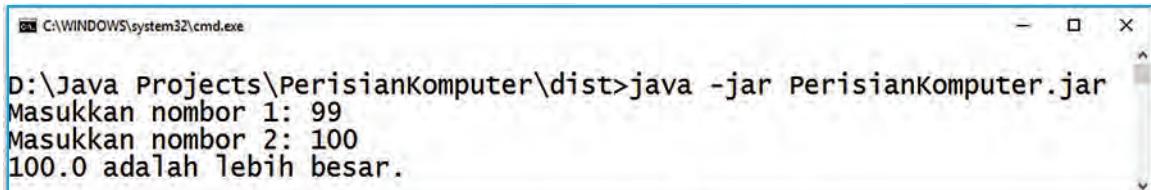
Perhatikan bahawa pengisytiharan `getNama()` mesti mempunyai jenis data pulangan `String`. Badan `getNama()` mempunyai pernyataan `return scanner.nextLine();` yang memulangkan `String`. Dalam `main()`, nilai yang dipulang diumpuk kepada boleh ubah `String x`. Pernyataan umpanan adalah `x = getNama();`.

Jadual 1.21 Contoh fungsi-fungsi utiliti dalam Java

Sintaks Fungsi	Penjelasan
<code>Math.sqrt(double n)</code>	Memulangkan hasil punca kuasa dua untuk nilai n . Contoh: <code>Math.sqrt(100);</code>
<code>Math.floor(double n)</code>	Memulangkan integer paling dekat tetapi kurang atau sama dengan nilai n . Contoh: <code>Math.floor(2.7);</code> memulangkan 2. <code>Math.floor(-2.7)</code> memulangkan -3.
<code>Math.round(double n)</code>	Memulangkan nombor n setelah dibulatkan kepada integer terdekat. Contoh: <code>Math.round(2.7);</code> memulangkan 3. <code>Math.round(2.3);</code> memulangkan 2.
<code>Math.max(double m, double n)</code>	Memulangkan nombor yang lebih besar antara m dan n . Contoh: <code>Math.max(100,10);</code> memulangkan 100.
<code>Math.min(double m, double n)</code>	Memulangkan nombor yang lebih kecil antara m dan n . Contoh: <code>Math.min(100,10);</code> memulangkan 10.

CONTOH (62)

Tulis aratur cara yang menerima dua integer dan tentukan nombor integer yang lebih besar menggunakan fungsi `Math.max(m,n)`.



```
C:\WINDOWS\system32\cmd.exe
D:\Java Projects\PerisianKomputer\dist>java -jar PerisianKomputer.jar
Masukkan nombor 1: 99
Masukkan nombor 2: 100
100.0 adalah lebih besar.
```

Penyelesaian:

```
package perisianSaya;
public class MyClass {

    static java.util.Scanner sc;
    sc = new java.util.Scanner(System.in);

    public static void main(String[] args){
        double nom1,nom2;
        System.out.println("Masukkan nombor 1: ");
        nom1 = scanner.nextInt();
        System.out.println("Masukkan nombor 2: ");
        nom2 = scanner.nextInt();
        System.out.println(Math.max(nom1,nom2) + " adalah lebih besar.");
    }
}
```

Dalam contoh di atas, pemboleh ubah `nom1` dan `nom2` digunakan sebagai parameter dalam `Math.max(nom1, nom2)`. Fungsi *built-in* Java ini akan memulangkan nombor yang lebih besar antara dua pilihan ini. Nombor yang dipulangkan akan dipaparkan terus dalam pernyataan `System.out.println()`;

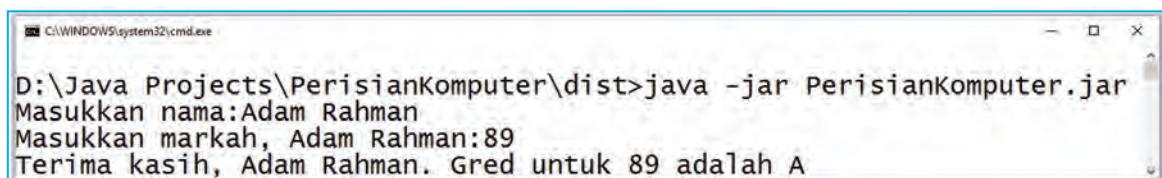


Berdasarkan Contoh 62, tulis aratur cara yang menerima empat integer dan tentukan nombor integer yang lebih besar menggunakan fungsi `Math.max(m,n)`.

1.6.4 Menulis Atur Cara Bermodular yang Mengandungi Struktur Tatasusunan

Atur Cara Bermodular

Atur cara berikut masing-masing meminta nama dan markah murid menggunakan prosedur-prosedur `getNama()` dan `getMarkah()`. Kemudian, `gred` ditentukan menggunakan prosedur `setGred()`. Prosedur `paparInfo()` memaparkan maklumat murid tersebut. Output interaksi dengan pengguna adalah seperti yang berikut:



```
C:\WINDOWS\system32\cmd.exe
D:\Java Projects\PerisianKomputer\dist>java -jar PerisianKomputer.jar
Masukkan nama:Adam Rahman
Masukkan markah, Adam Rahman:89
Terima kasih, Adam Rahman. Gred untuk 89 adalah A
```

Rajah 1.66 Output interaktif pengguna

Atur cara ini memanggil beberapa prosedur daripada `main()`.

```
1 package perisianSaya;
2 public class MyClass {
3
4     static String nama;
5     static int markah;
6     static String gred;
7     static java.util.Scanner scanner = new java.util.Scanner(System.in);
8
9     public static void main(String[] args) {
10
11         getNama();
12         getMarkah();
13         setGred();
14         paparInfo();
15     }
16 }
```

Rajah 1.67 Subatur cara `main()` memanggil beberapa subatur cara lain

Atur cara bagi contoh ini ialah atur cara *console*. Pemboleh ubah-pemboleh ubah nama, markah dan gred diisyiharkan sebagai pemboleh ubah sejagat (*global*). Pemboleh ubah sejagat mempunyai kelebihan dari segi capaian untuk semua subatur cara dalam atur cara. Ini bermakna prosedur `main()`, `getNama()`, `getMarkah()`, `setGred()` dan `paparInfo()` dapat membaca dan menulis kepada set pemboleh ubah-pemboleh ubah yang sama.

Perhatikan bahawa dalam prosedur `main()`, tidak terdapat banyak pernyataan algoritma. Ini kerana pernyataan-pernyataan tersebut telah diletakkan dalam subatur cara yang didefinisikan dalam rajah-rajab yang berikut. Memanggil subatur cara tersebut harus mengikut urutan logik.

```
17     static void getNama() {
18         System.out.print("Masukkan nama:");
19         nama = scanner.nextLine();
20     }
21
22     static void getMarkah() {
23         System.out.print("Masukkan markah, " + nama + ":");
24         markah = scanner.nextInt();
25     }
26
```

Rajah 1.68 Subatur cara `getNama()` dan `getMarkah()`

Tahukah Anda?

Mari kita imbas semula mengenai pemboleh ubah sejagat!

Pemboleh ubah sejagat

- Pemboleh ubah sejagat hanya berfungsi dalam atur cara sahaja.
- Penggunaan pemboleh ubah sejagat adalah semasa dalam fungsi atur cara sahaja. Apabila tatacara itu telah tamat, ruang memori yang diperuntukkan kepada pemboleh ubah dan pemalar juga akan tamat.

Subatur cara dalam Rajah 1.69 meminta pengguna memasukkan nama dan markah yang akan disimpan dalam pemboleh ubah-pemboleh ubah sejagat (*global*), iaitu `nama` dan `markah`.

```
27     static void setGred() {
28         if(markah>=90){gred = "A+";}
29         else if(markah>=80){gred = "A";}
30         else if(markah>=75){gred = "A-";}
31         else if(markah>=70){gred = "B+";}
32         else if(markah>=65){gred = "B";}
33         else if(markah>=60){gred = "C+";}
34         else if(markah>=50){gred = "C";}
35         else if(markah>=45){gred = "D";}
36         else if(markah>=40){gred = "E";}
37         else {gred = "F";}
38     }
39
```

Rajah 1.69 Subatur cara `setGred()`



Prosedur `setGred()` akan menggunakan markah yang dimasukkan untuk menentukan `gred` yang berpadanan. Struktur kawalan pilihan `if-else` digunakan. Keputusan pemilihan disimpan dalam boleh ubah `gred`.

```
40     static void paparInfo() {
41         System.out.print("Terima kasih, " + nama + ", ");
42         System.out.println("Gred untuk " + markah + " adalah " + gred);
43     }
44 }
45 }
```

Rajah 1.70 Prosedur `paparInfo()`

Prosedur `paparInfo()` akhirnya mendapatkan kembali nilai-nilai boleh ubah untuk dipaparkan kembali kepada pengguna. Nilai boleh ubah nama dan markah berasal daripada input pengguna melalui prosedur `getNama()` dan `getMarkah()`. Nilai `gred` ditentukan oleh prosedur `setGred()`.

Struktur Tatasusunan dalam Atur Cara Bermodular

Bolehkah bilangan data yang amat banyak dihantar melalui parameter? Cuba bayangkan penghantaran 100 data daripada subatur cara pemanggil kepada subatur cara dipanggil.

Bagaimanakah pula subatur cara memulangkan bilangan data yang banyak kepada pemanggil? Bukankah subatur cara prosedur tidak mengembalikan data? Bagi subatur cara fungsi, cuma satu nilai boleh dikembalikan.

Jika anda masih ingat penggunaan tatasusunan, anda mungkin dapat menjawab soalan-soalan ini. Tatakusunan ialah satu boleh ubah yang boleh menyimpan banyak elemen menggunakan nama yang sama (tetapi indeks berlainan yang unik mewakili setiap elemen). Bolehkah data yang banyak diletakkan ke dalam satu tatasusunan untuk dihantar kepada subatur cara sebagai satu parameter? Lihat bagaimana 10 integer disimpan dalam boleh ubah tatasusunan `senaraiNombor`.

```
int[] senaraiNombor={1,2,3,4,5,6,7,8,9,10};
```

Untuk menghantar tatasusunan sebagai parameter, pastikan subatur cara mampu menerima parameter tatasusunan. Ini dipastikan daripada pengisytiharan kepala subatur cara yang menerima parameter jenis tatasusunan. Lihat jenis data dalam kepala subatur cara yang berikut:

```
static void paparSenarai(int[] senaraiNombor) { }
```

Bagaimanakah tatasusunan diproses dalam subatur cara? Struktur kawalan ulangan `for` diperlukan untuk mengumpuk atau mengakses nilai elemen-elemen dalam tatasusunan. Jika terdapat 10 elemen dalam tatasusunan, gelung `for` perlu mengulang 10 kali dari indeks 0 sehingga 9.

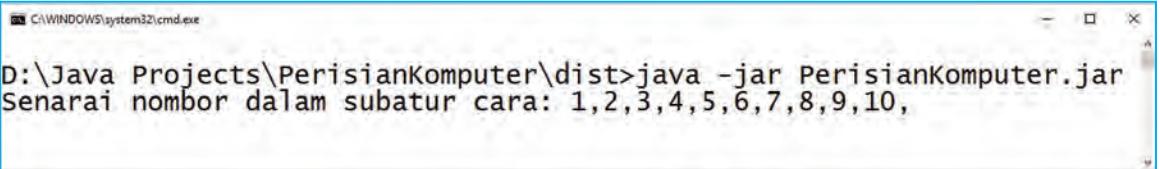
```
for(int i=0; i<10; i++)
{System.out.print(senaraiNombor[i] + ",");}
```

Bagaimanakah pula mengembalikan data-data tatasusunan? Elemen-elemen dalam satu-satu tatasusunan boleh diubah terus daripada mana-mana subatur cara. Oleh itu, pemboleh ubah tatasusunan tidak perlu dikembalikan kepada pemanggil.

Mari kita lihat beberapa contoh penggunaan tatasusunan dalam modul.

CONTOH (63)

Tatasusunan `senaraiNombor` mengandungi 10 integer dari 1 hingga 10. Sebuah atur cara menghantar `senaraiNombor` sebagai parameter kepada subatur cara prosedur `paparSenarai` untuk dipaparkan. Output dipaparkan seperti berikut:



```
C:\WINDOWS\system32\cmd.exe
D:\Java Projects\PerisianKomputer\dist>java -jar PerisianKomputer.jar
Senarai nombor dalam subatur cara: 1,2,3,4,5,6,7,8,9,10,
```

Kod untuk menghasilkan output seperti keperluan di atas adalah seperti yang berikut:

```
1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args){
5         int[] senaraiNombor={1,2,3,4,5,6,7,8,9,10};
6         paparSenarai(senaraiNombor);
7     }
8
9     static void paparSenarai(int[] senaraiNombor)
10    {
11        System.out.print("Senarai nombor dalam subatur cara: ");
12
13        for(int i=0; i<10; i++)
14            {System.out.print(senaraiNombor[i] + ",");}
15    }
16 }
17 }
```

Contoh di atas menunjukkan kesemua elemen dalam tatasusunan dihantar kepada subatur cara `paparSenarai`. Penggunaan tatasusunan memudahkan penghantaran tersebut kerana saiz tatasusunan boleh diubah untuk menghantar lebih banyak elemen. Jadi, adalah mungkin untuk menghantar 100 nombor integer melalui tatasusunan `senaraiNombor` dalam contoh di atas.

CONTOH (64)

Tatasusunan nombor yang kosong diisyiharkan dalam prosedur `main()`. Dari `main()`, panggilan dibuat kepada prosedur `setSenaraiRawak()`. Dalam prosedur `setSenaraiRawak()`, tatasusunan kosong nombor diumpukan dengan nilai-nilai rawak yang dijanakan oleh `Math.random()`. Nilai elemen-elemen dipaparkan pada kedua-dua subatur cara `setSenaraiRawak()` dan prosedur `main()`. Kod untuk contoh ini adalah seperti yang berikut:

```
1 package perisianSaya;
2 public class MyClass {
3
4     public static void main(String[] args){
5         int[] senaraiNombor= new int[10];
6         setSenaraiRawak(senaraiNombor);
7         System.out.println("\n\nDalam subatur cara main:");
8         for(int i=0; i<10; i++)
9             {System.out.print(senaraiNombor[i] + ",");}
10    }
11
12     public static void setSenaraiRawak(int[] senaraiNombor)
13     {
14         System.out.println("Dalam subatur cara setSenaraiRawak: ");
15         for(int i=0; i<10; i++)
16         {
17             senaraiNombor[i] = (int) (Math.random()*10) + 1;
18             System.out.print(senaraiNombor[i] + ",");
19         }
20     }
21 }
```

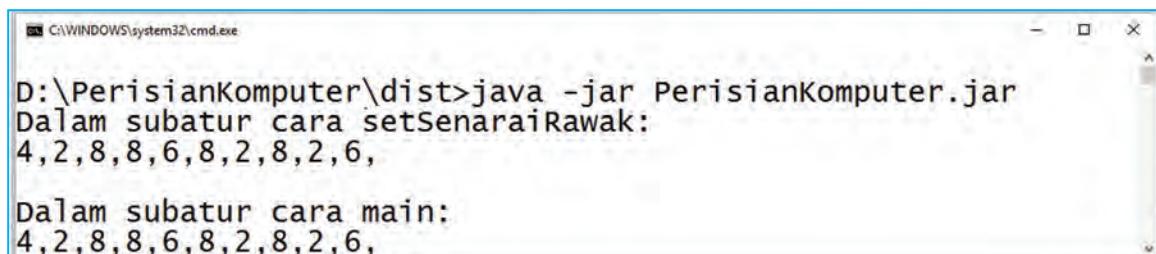
Subatur cara `setSenaraiRawak` dipanggil dari `main()` di baris 6. Tatasusunan `senaraiNombor` yang masih kosong dihantar sebagai parameter kepada subatur cara `setSenaraiRawak` untuk diisi dengan nombor yang rawak.

Lihat baris 17. `Math.random()` ialah subatur cara fungsi Java yang akan menjana nombor perpuluhan rawak di antara 0.0 dan 1.0. Untuk menjana nombor rawak di antara 1 dan 10 (termasuk 1 dan 10), ungkapan berikut digunakan:

```
(int)(Math.random()*10) + 1;
```

Dalam ungkapan ini, penggunaan `(int)` memastikan cuma bahagian integer bagi hasil `double` `(Math.random()*10)` iaitu antara 0 dan 9 diambil. Penambahan dengan integer 1 memastikan julat nombor rawak di antara 1 dan 10. Proses penjanaan nombor rawak dilakukan sebanyak 10 kali dalam struktur kawalan ulangan `for`. Dalam blok ulangan yang sama, nombor rawak dimasukkan ke dalam tatasusunan `senaraiNombor` dan dipaparkan pada skrin.

Output untuk contoh ini adalah seperti yang berikut:



```
C:\WINDOWS\system32\cmd.exe
D:\PerisianKomputer\dist>java -jar PerisianKomputer.jar
Dalam subatur cara setSenaraiRawak:
4,2,8,8,6,8,2,8,2,6,
Dalam subatur cara main:
4,2,8,8,6,8,2,8,2,6,
```

Contoh 64 menunjukkan bahawa penghantaran tatasusunan melalui parameter tidak seperti penghantaran pemboleh ubah. Dalam penghantaran pemboleh ubah biasa melalui parameter, nilai pemboleh ubah mempunyai dua salinan di kedua-dua subatur cara pemanggil dan subatur cara dipanggil. Perubahan yang dilakukan di dalam subatur cara dipanggil akan mengubah nilai asal di dalam subatur cara pemanggil.

Latihan Formatif 1.6

- 1 Gunakan pemboleh ubah mudah untuk menyimpan semua data dalam jadual berikut:

Pelajar	Markah
Adam M. Saad	34
Alia Abdul Majid	56
Ang Yi Leng	78

- 2 Perhatikan jadual yang berikut:

No	Pelajar	Markah
1	Adam M. Saad	34
2	Alia Abdul Majid	56
3	Ang Yi Leng	78

- (a) Isytiharkan tatasusunan senaraiPelajar.
- (b) Isytiharkan tatasusunan senaraiMarkah
- (c) Umpukkan tatasusunan senaraiPelajar.
- (d) Umpukkan tatasusunan senaraiMarkah

- 3 Diberikan tatasusunan yang berikut:

```
int [] senaraiUmur = {17, 17, 18, 18, 18, 19};
```

Tentukan elemen yang dipulangkan oleh setiap arahan berikut:

- (a) `senaraiUmur[5]` akan memberikan nilai 19
- (b) `senaraiUmur[1]` akan memberikan nilai 17
- (c) `senaraiUmur[3]` akan memberikan nilai 18
- (d) `senaraiUmur[2]` akan memberikan nilai 18
- (e) `senaraiUmur[0]` akan memberikan nilai 17
- (f) `senaraiUmur[4]` akan memberikan nilai 18

- 4 Tulis subatur cara baharu `inputMarkahUjian()`. Subatur cara ini menerima input markah ujian untuk mata pelajaran Bahasa Melayu, Bahasa Inggeris dan Matematik. Output adalah seperti yang berikut:

Input markah Bahasa Melayu:

85

Input markah Bahasa Inggeris:

83



Input markah Matematik:

90

Markah Bahasa Melayu: 85

Markah Bahasa Inggeris: 83

Markah Matematik: 90

- 5 Tulis subatur cara baharu tentuGred(). tentuGred() menerima input markah murid dan menggunakan struktur kawalan pilihan untuk memilih gred yang bersesuaian. Seterusnya, gred dipaparkan pada skrin komputer. Gred markah ditentukan berdasarkan jadual berikut:

Markah	Gred
90-100	A+
80-89	A
75-79	A-
70-74	B+
65-69	B
60-64	C+
50-59	C
45-49	D
40-44	E
0-39	G

- 6 Dalam kod Java yang berikut:

```
1 public static void main(String [] args){  
2 paparHarga("telefon pintar",1500.0);  
3 }  
4 static void paparHarga(String item,  
double h){  
5 System.out.print("Harga " + item + "  
adalah " + h);  
6 }
```

Kenal pasti baris pernyataan di bawah dengan menulis nombor baris bagi atur cara di atas:

- (a) Subatur cara main()
- (b) Definisi paparHarga
- (c) Panggilan paparHarga
- (d) Subatur cara pemanggil
- (e) Subatur cara dipanggil

7 Diberikan definisi berikut:

```
static void kuasaDua(int nom){  
    double jawapan = nom * nom;  
    System.out.print(jawapan);  
}
```

- (a) Tulis pernyataan kod Java dalam subatur cara `main()` untuk definisi subatur cara yang diberikan bagi menentukan kuasa dua bagi nilai-nilai berikut:
(i) 2
(ii) 10
- (b) Bolehkah subatur cara yang diberi digunakan untuk memproses data "2" ?
Justifikasikan jawapan anda.
- (c) Bolehkah subatur cara yang diberi digunakan untuk memproses data "12.3" ?
Justifikasikan jawapan anda.
- 8** Tulis atur cara lengkap dalam kod Java untuk menerima input nama pengguna dan kemudian menghantar nama tersebut kepada parameter subatur cara `hello` untuk dipaparkan.

9 Diberikan dua set atur cara Java seperti yang berikut:

Atur cara A	Atur cara B
<pre>static void inkremen(int nom){ nom = nom + 1; } public static void main(String[] args) { System.out.print(inkremen(9)); }</pre>	<pre>static int inkremen(int nom){ nom = nom + 1; return nom; } public static void main(String[] args) { System.out.print(inkremen(9)); }</pre>

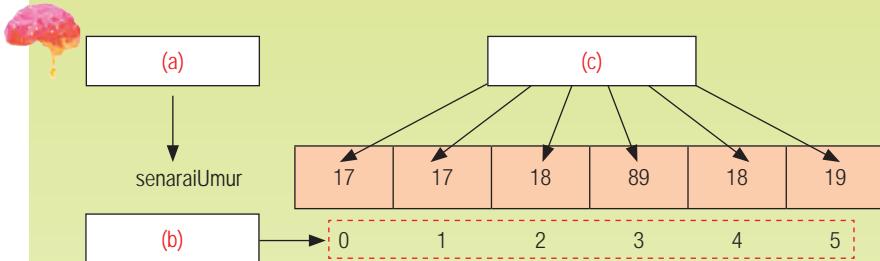
- (a) Tentukan output Atur cara A.
(b) Tentukan output Atur cara B:
(c) Tentukan yang mana satu menggunakan subatur cara fungsi.
Bagaimakah anda menentukannya?
(d) Tentukan yang mana satu menggunakan subatur cara prosedur.
Bagaimakah anda menentukannya?

10 Tulis kod Java yang mendefinisikan subatur cara `cariTerbesar()`. Subatur cara tersebut akan menyemak tatasusunan dalam terima untuk mencari dan memulangkan elemen terbesar.
Diberikan sebahagian kod dalam subatur cara `main()` seperti yang berikut:

```
double []senaraiNombor = {2.0, 3.142, 37.5};  
double nomborMaksimum = cariTerbesar(senaraiNombor);
```



11 Perhatikan sel memori untuk suatu tatasusunan berikut:



Kenal pasti nama komponen sel memori (a), (b) dan (c) untuk struktur tatasusunan.

12 Diberikan definisi berikut.

```
static void paparKuantiti(String item, int x){  
    System.out.print("Kuantiti " + item + " adalah " + x);  
}
```

Tentukan sama ada panggilan yang berikut padan ataupun tidak dengan parameter subatur cara. Berikan penjelasan anda jika padanannya tidak.

- (a) paparKuantiti("telefon pintar", 10.0);
- (b) paparKuantiti(1500.0, "telefon pintar");
- (c) paparKuantiti(30, "telefon pintar");
- (d) paparKuantiti("telefon pintar", 30);
- (e) paparKuantiti("telefon pintar",10, 1500.0);

13 Tulis kod Java bagi subatur cara fungsi yang mencari nilai hipotenusa bagi sebuah segi tiga dengan sisi x dan y. Formulanya ialah, hipotenusa = $\sqrt{x^2+y^2}$.

Nota: Gunakan fungsi *built-in* Java Math.sqrt(double n).

STANDARD KANDUNGAN

1.7

STANDARD PEMBELAJARAN

1.7.1 Menguraikan setiap fasa dalam Kitaran Hayat Pembangunan Sistem (SDLC):

- (i) Menganalisis masalah bagi menentukan keperluan
- (ii) Mereka bentuk penyelesaian
- (iii) Melaksanakan penyelesaian
- (iv) Menguji dan menyah ralat
- (v) Dokumentasi

1.7.2 Mencipta aplikasi atau aplet dengan antara muka bergrafik

Pembangunan Aplikasi

Kitaran Hayat Pembangunan Sistem (*Software development Life Cycle -SDLC*) juga dikenali sebagai kitaran hayat pembangunan aplikasi. Kitaran hayat pembangunan aplikasi ialah istilah yang digunakan dalam kejuruteraan sistem dan perisian, sistem maklumat, dan pembangunan aplikasi yang menjelaskan proses merancang, mereka bentuk, menguji dan mengimplementasi sesuatu aplikasi atau perisian. SDLC terdiri daripada satu kitaran fasa yang berjujukan dan menjadikannya sebagai pelan tindakan yang berkesan kepada pasukan projek. Selain itu, SDLC bantu mengesan status bagi penyempurnaan projek tersebut.

Metodologi umum yang terdapat bagi SDLC ialah model air terjun (*Waterfall model*), model RAD (*Rapid Application Development model*), model lelaran (*Iterative model*), model lingkaran (*Spiral model*) dan model tangkas (*Agile model*). Model hibrid pula ialah kombinasi yang terdiri daripada beberapa model. Dokumentasi adalah penting tanpa mengira model yang digunakan dan selalunya seiring dengan proses pembangunan. Misalnya, ada kaedah yang sesuai dengan jenis projek yang spesifik tetapi dalam analisis terakhir, faktor terpenting ialah kejayaan projek yang dibangunkan.

Model air terjun digunakan sebagai model pembangunan aplikasi memandangkan model ini mirip kepada proses-proses dalam SDLC. Model ini merupakan model terawal, mudah difahami dan mudah diuruskan. Model air terjun mengandungi lima fasa secara umumnya, di mana setiap fasa dalam model air terjun distrukturkan iaitu satu fasa perlu diselesaikan sebelum ke fasa berikutnya. Maklumat bagi setiap fasa diperlukan untuk fasa yang berikutnya dan tidak boleh berpatah balik.



1.7.1 Kitaran Hayat Pembangunan Sistem

1.7.1.1 Fasa Analisis Masalah

Fasa analisis masalah ialah proses mengenal pasti keperluan program dan mencari sebab sesuatu program dibina. Langkah-langkah sistematik harus dipatuhi untuk menyelesaikan masalah dan penting untuk kita memahami pernyataan masalah dengan jelas. Permulaannya, analisis masalah yang dihadapi dengan menggunakan analisis IPO iaitu analisis input, proses dan output.

Contoh yang berikut menunjukkan cara untuk membuat analisis masalah dengan menggunakan carta IPO.

CONTOH (65)

Katakan anda dikehendaki menyelesaikan masalah kiraan gaji staf mengikut jam bekerja.

Penyelesaian:

- Cara mengisi carta IPO adalah dengan menentukan penyelesaian yang dikehendaki iaitu gaji staf sebagai output.
- Formula untuk hasilkan output ‘gaji staf’; iaitu kadar sejam bekerja didarabkan dengan bilangan jam bekerja.
- Input untuk formula iaitu kadar sejam bekerja dan bilangan jam bekerja.

Analisis IPO bagi masalah di atas adalah seperti yang berikut:

Input	Proses	Output
Kadar sejam Bilangan jam bekerja	1 Dapatkan kadar sejam. 2 Dapatkan bilangan jam bekerja. 3 Kirakan gaji staf = kadar x bilangan sejam jam bekerja	Gaji staf



Beberapa frasa sering digunakan untuk menggambarkan input, output dan proses. Misalnya, frasa yang digunakan:

Input – pengguna masukkan,

Prompt ... for..., Input pengguna

Proses – ...masukkan input dalam algoritma..., ...guna formula...

Output – ...untuk paparkan..., untuk tentukan..., untuk kira...



Model Air Terjun
goo.gl/8BLz8D

1.7.1.2 Fasa Reka bentuk Penyelesaian

Fasa reka bentuk penyelesaian masalah dibuat setelah analisis IPO yang merupakan set keperluan untuk memulakan fasa mereka bentuk dilakukan. Walaupun projek yang kecil dan keperluan yang mudah, satu reka bentuk mental untuk pemahaman keperluan dengan penyelesaian perlu wujud. Fasa ini melihat kepada potensi penyelesaian yang wujud dan menentukan penyelesaian yang efektif dan efisien. Hal ini menjawab “bagaimana” membina penyelesaian terbaik.



International Olympiad in Informatics (IOI) merupakan salah satu pertandingan antarabangsa dalam bidang sains dan informatik. Pertandingan tahunan setanding sukan Olimpik ini bertujuan melahirkan minat murid sekolah menengah terhadap sains komputer dan pengaturcaraan. Setakat ini, Malaysia berjaya merangkul dua pingat perak dan tiga pingat gangsa sejak menyertai pertandingan ini dari tahun 2012.



Kerja Sains Komputer



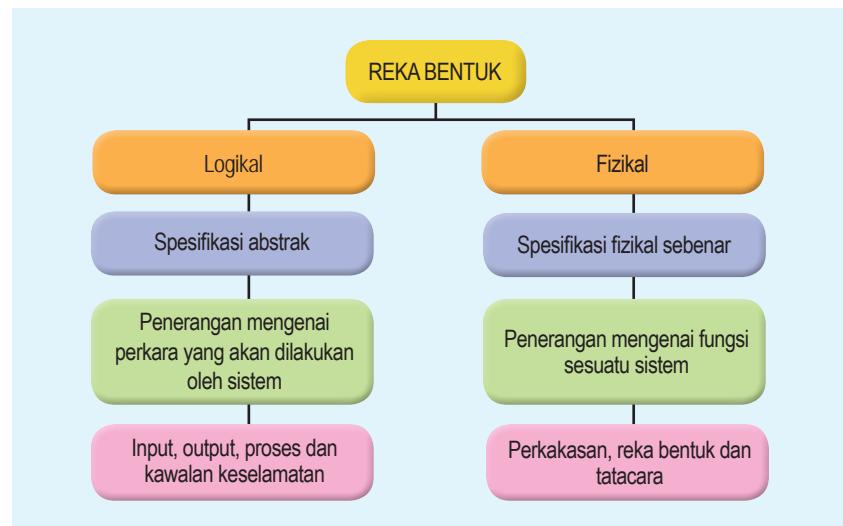
Rahsia Kejayaan Mark Zuckerberg
goo.gl/WJirLL



SDLC

Penggunaan algoritma ialah langkah awal bagi penyelesaian masalah dalam fasa reka bentuk penyelesaian. Pseudokod ialah aturan langkah yang ditulis dalam bahasa pertuturan manakala carta alir adalah satu perwakilan grafik yang menunjukkan langkah penyelesaian sesuatu masalah dan mempunyai hubung kait antara satu sama lain. Misalnya,

Algoritma untuk mengira gaji staf:
Dapatkan kadar sejam bekerja bagi staf.
Kemudian, darabkan bilangan jam staf tersebut bekerja. Hasilnya ialah gaji staf tersebut.



Rajah 1.71 Reka bentuk dari aspek logikal dan fizikal

Selepas membina algoritma, kita akhiri dengan menulis pseudokod dan carta alir. Hasilnya ialah seperti yang ditunjukkan di bawah.

Pseudokod	Carta alir
<pre> 1. Baca kadar sejam, kadarsj 2. Baca bilangan jam bekerja, bilJam 3. Hitung gaji staf = kadarsj * bilJam 4. Paparkan gaji staf </pre>	<pre> graph TD M([Mula]) --> I1[/Input kadar sejam
Input bilangan jam/] I1 --> H[Hitung Gaji staf] H --> P[/Paparkan "Gaji anda ialah RM"/] P --> T([Tamat]) </pre>

Tulis pseudokod dan lakar carta alir untuk situasi di bawah:

- Memaparkan teks "Salam 1Malaysia", "Tahniah anda hadir ke sekolah pada hari ini."
- Menyatakan komen kepada markah. Jika 86-100%, "Syabas, teruskan usaha anda!" 60-85%, "Baik, anda boleh maju lagi!" Kurang daripada 60%, "Jangan putus asa, cuba lagi sehingga berjaya!"



Aplikasi Harian

Kepentingan temu bual dan soal selidik untuk mendapatkan maklumat yang diperlukan dalam fasa analisis.



goo.gl/FTn1QW

1.7.1.3 Fasa Pelaksanaan Penyelesaian

Tujuan utama fasa perlaksanaan penyelesaian adalah untuk mengubah reka bentuk kepada program yang akan dipasang pada perkakasan dan bersedia untuk melaksanakan penyelesaian. Selain itu, fasa ini bertujuan untuk membina dan menghasilkan sistem yang dapat menyelesaikan masalah yang dihadapi.

Aktiviti pembangunan aplikasi melibatkan pengekodan. Pengekodan memerlukan bahasa pengaturcaraan seperti *Java*, *Javascript* dan sebagainya. Pengekodan juga mengubah spesifikasi program kepada kod sumber. Pengkompilan ialah proses menukar kod pengaturcaraan kepada kod boleh laksana (*executable*).

```
//Atur cara untuk memaparkan mesej Hello World
public class HelloWorld {
    public static void main(String[ ] args)
        // Paparkan teks
        System.out.println("Hello World!");
}
```

Rajah 1.72(a) Contoh kod atur cara

Rajah 1.72(b) Contoh kod boleh laksana

Setelah menyediakan pseudokod dan carta alir, kita boleh ke langkah seterusnya iaitu pelaksanaan penyelesaian untuk mencari gaji staf dengan menulis kod atur cara. Jenis data yang digunakan ialah *integer* untuk memboleh ubah bilangan jam bekerja iaitu 'bilJam', manakala *double* digunakan untuk memboleh ubah kadar sejam seseorang itu bekerja dan gaji staf. Nama memboleh ubah yang digunakan ialah 'gajistaf' dan 'kadarsj'. Penyelesaian aritmetik yang dilakukan ialah mendarab kadar sejam bekerja dengan bilangan jam bekerja. Hasil daripada aplikasi ini akan memaparkan gaji staf.



Uji Minda

Bina satu kod atur cara untuk gaji staf dengan ketetapan seperti yang ditunjukkan di bawah:

(a) Bilangan tahun bekerja 1-3 tahun, kadar sejam ialah RM12.50

(b) Bilangan tahun bekerja 4-6 tahun, kadar sejam ialah RM14.00

(c) Bilangan tahun bekerja 7 tahun ke atas, kadar sejam ialah RM16.00

```
public class Gajistaf {
    public static void main(String[] args) {
        int bilJam = 20;
        double gajistaf, kadarsj;
        kadarsj = 25.0;
        gajistaf = kadarsj*bilJam;
        System.out.println("Gaji anda ialah RM"+gajistaf);
    }
}
```



30

Fasa analisis, reka bentuk dan pelaksanaan penyelesaian

Kerja Individu

Isikan tempat kosong dengan aktiviti pada fasa yang betul.

Spesifikasi Reka Bentuk Sistem

Keperluan Sistem

Pelan Penyelenggaraan

Sistem Baharu

Arahan untuk Pengatur cara

Proposal Sistem



Analisis



Reka bentuk



Pelaksanaan penyelesaian

1.7.1.4 Fasa Uji dan Nyah ralat

Dalam fasa ini, kod atur cara yang siap dibina akan diuji. Fasa pengujian memainkan peranan penting dalam SDLC. Fasa ini mempunyai beberapa tujuan seperti yang berikut:

- a** Memastikan semua keperluan dipenuhi.
- b** Memastikan semua pengekodan berfungsi seperti yang dikehendaki.
- c** Memastikan semua modul boleh berfungsi bila digabungkan.
- d** Mendapatkan maklum balas daripada pengguna sistem untuk tujuan pembetulan dan penambahbaikan.
- e** Melibatkan pengguna sepenuhnya di peringkat pembangunan.
- f** Mengesan ralat yang tercicir.
- g** Membantu pasukan projek membuat dokumentasi dengan mengesan kesilapan oleh pengguna.
- h** Menyimpan keputusan ujian sebagai bukti penyempurnaan pembangunan sistem.

Semakan kod (*code review*) dilakukan untuk mengesan ralat. Pengatur cara akan merujuk log yang dipaparkan untuk membetulkan dan membuang ralat yang dikesan. Semakan boleh dibuat dalam tiga peringkat seperti Jadual 1.22.

Jadual 1.22 Jenis-jenis semakan

Jenis Semakan	Siapa?	Bila?
Sendiri	Pengarang	Semasa pengekodan
Rakan Sebaya	Rakan Sebaya	Selepas tamat modul
Selepas tamat modul	Pasukan projek yang diketuai oleh pakar bahasa pengaturcaraan	Selepas kedua-kedua peringkat di atas

Jenis-jenis ralat seperti ralat sintaks, ralat logik dan ralat masa larian dilakukan sehingga kod atur cara itu bebas dari semua ralat. Semua ralat yang ditemui, dibetulkan dan diuji semula.



Uji Minda

Apakah langkah-langkah penambahbaikan yang diperlukan untuk atur cara bebas ralat? Uji dan pamerkan hasil kerja anda. Sebagai latihan imbas semula, nyatakan apakah ralat sintaks logik dan ralat masa larian?



Phases of System
Development Life
goo.gl/ZFav0F



Inovasi Sains
Komputer

Ikuti pelbagai cerita yang menarik mengenai pengatur cara yang bersusah-payah melakukan inovasi dalam bidang pengkomputeran.



goo.gl/WqE12V

Jadual 1.22 Jenis pengujian dan perincian



Langkah-langkah menguji arut cara mengikut program yang dibina
goo.gl/vjCiLm

Jenis Pengujian	Perincian
Sendiri	Memastikan setiap unit dalam sistem yang dibina berfungsi.
Sistem	Menguji sistem secara keseluruhan selepas ujian atas unit individu.
Integrasi	Memastikan sistem dapat berfungsi dengan sistem sedia ada.
Penerimaan	Ujian ini dijalankan semasa ujian integrasi oleh pengguna sistem untuk memastikannya memenuhi kehendak pengguna.

Skrip ujian perlu ditulis untuk mengautomasikan proses pengujian modul sistem kerana terdapat ujian-ujian yang perlu dilakukan berulang kali. Tandakan dalam senarai semak di bawah untuk menguji dan menyah ralat pengiraan gaji staf.

Jadual 1.23 Senarai semak pengiraan gaji staf

Item	Aktiviti	Tandakan(✓)atau (x)
Ralat sintaks	Ejaan teks nama boleh ubah	
	Ejaan teks komen	
	Penggunaan objek atau aksara yang tidak dikenali	
	Pengisytiharan jenis data untuk bilangan jam bekerja	
	Pengisytiharan jenis data untuk gaji staf	
	Pengisytiharan jenis data kadar sejam	
Ralat masa larian	Input pengiraan untuk bilangan jam	
	Input pengiraan untuk kadar sejam	
Ralat logik	Semak output gaji staf	

Dalam pembangunan aplikasi, dokumentasi ialah ekosistem isi kandungan dengan tujuan mengajar pengguna berinteraksi dengan *Application Programming Interface (API)*. API ialah satu set rutin, protokol dan alat untuk membina aplikasi. Sesuatu API menentukan bagaimana komponen aplikasi harus berinteraksi. API yang baik memudahkan pembangunan aplikasi dengan menyediakan blok pembangunan, di mana pengatur cara komputer akan mencantumkan blok-blok tersebut.

Bayangkan dalam situasi ketika memesan makanan di kedai makan di mana pelayan itu ialah API. Dapur dan menu adalah sebagai sebahagian daripada sistem. Pelayan sebagai API akan mengambil pesanan anda dan memberitahu dapur iaitu sistem apa yang perlu dilaksanakan. Kemudian, pelayan akan membawa makanan kepada anda, iaitu respons balik kepada anda. Contoh API ialah seperti API *Google maps* dan API *Twitter*.

Nyatakan jenis pengujian yang harus dijalankan berdasarkan situasi di bawah.

Situasi	Jenis Pengujian
Sistem Analisis Peperiksaan Sekolah (SAPS)	
Aplikasi operasi tambah dalam sistem aritmetik (1-10)	
Sistem Daftar Murid yang akan digunakan dengan Sistem Peminjaman Buku di Pusat Sumber Sekolah (PSS)	
Sistem Pengurusan Hasil	
Sistem POS di Restoran CZF Sdn. Bhd.	

1.7.1.5 Fasa Dokumentasi

Dokumentasi merupakan suatu proses mengutip dan mengumpulkan data, mengumpulkan maklumat dan ringkasan seperti laporan pengujian yang dijalankan, carta alir, kod atur cara dan juga carta IPO. Dokumen-dokumen ini adalah sangat penting untuk rujukan pengguna sistem, pegawai IT dan juga kakitangan baharu di setiap fasa. Sebagai contoh, carta Gantt ialah dokumen penting kerana carta Gantt menetapkan tarikh akhir untuk setiap peringkat pembangunan.

Setiap fasa perlu disiapkan sebelum ke fasa seterusnya. Hasil bagi satu-satu fasa berfungsi sebagai input untuk urutan fasa yang seterusnya. Oleh itu, dokumentasi yang sepenuhnya bagi fasa projek dari awal pada setiap fasa akan dijadikan sebagai rujukan untuk fasa seterusnya.

Contoh dokumentasi dalam aplikasi mengira gaji staf adalah seperti yang ditunjukkan di bawah.

- a) komen untuk kod atur cara mengira gaji staf

```

int bilJam = 20; //Pengisytiharan boleh ubah
double gajistaf, kadarsj; //Pengisytiharan boleh ubah
kadarsj = 25.0; // Mendapatkan data kadar bayaran sejam
    
```

b Carta Gantt untuk pembangunan aplikasi bagi mengira gaji staf

Senarai Aktiviti	20 Mac	21 Mac	22 Mac	23 Mac	24 Mac	25 Mac	26 Mac	27 Mac	28 Mac
1. Menyediakan borang maklumat staf	█								
2. Mendapatkan maklumat bagi kerja yang dilakukan oleh staf		█							
3. Membina algoritma untuk mengira gaji staf			█						
4. Membina kod atur cara				█					
5. Menjalankan pengujian dan nyah ralat					█				
6. Menambahbaik dan membetulkan ralat						█			
7. Menggunakan atur cara – gaji staf							█		
8. Dokumentasi								█	

c Pengujian dan nyah ralat

Nama	Jenis	Penerangan	Catatan
bilJam	integer	Bilangan jam bekerja	Tidak boleh angka negatif
kadarsj	double	Kadar sejam	Dinyatakan dalam RM
gajistaf	double	Hasil darab bilangan jam bekerja dengan kadar sejam	Dinyatakan dalam RM

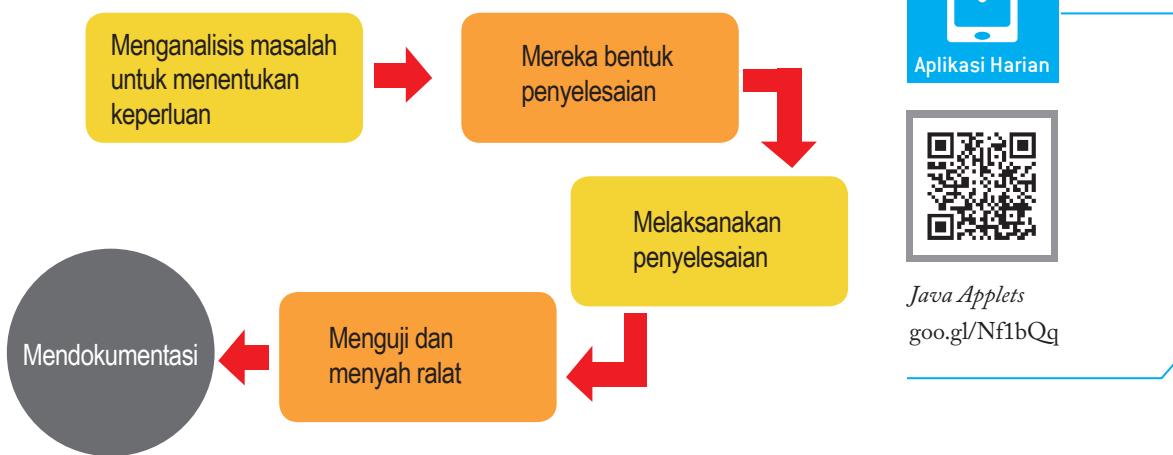
d API

Nama Staf	Input: Bilangan jam bekerja	Input: Kadar Sejam	Output: Gaji Staf	Catatan
A101	20	25.00	500.00	

1.7.2 Penciptaan Aplikasi atau Aplet

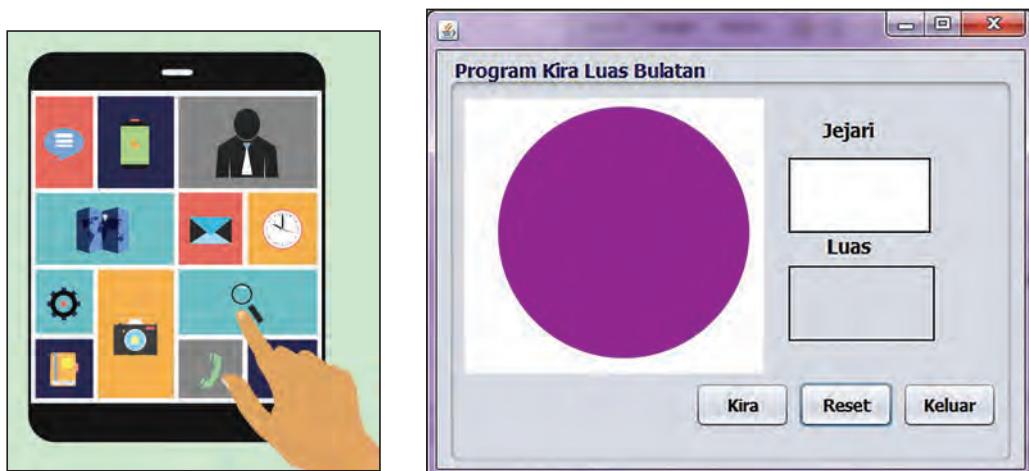
Dengan merujuk fasa-fasa dalam Model Air Terjun, kita dapat membina dan menghasilkan aplikasi penyelesaian masalah yang mengandungi

subatur cara atau tatasusunan dengan menggunakan pernyataan umpan dan struktur kawalan dalam menghasilkan output berdasarkan input yang diberikan seperti dalam rajah yang ditunjukkan di bawah.



Rajah 1.73 Langkah-langkah mencipta sebuah aplikasi atau aplet

Antara muka bergrafik pengguna (*Graphical user interface, GUI*) adalah antara muka pengguna yang membolehkan pengguna berinteraksi dengan sistem seperti komputer dan telefon pintar. Ciri utama dalam GUI ialah penunjuk dan ikon bersifat visual. Pengguna boleh melakukan tindakan dengan memanipulasikan atau mengolah ikon grafik tersebut.



Rajah 1.74 Contoh paparan antara muka bergrafik

Dalam kehidupan seharian, kita sering menghadapi situasi yang memerlukan penyelesaian. Namun begitu, banyak masalah dapat diselesaikan dengan pembangunan aplikasi yang mudah. Mari kita lihat masalah mencari luas objek yang selalunya diselesaikan dalam mata pelajaran Matematik. Kaedah alternatif penyelesaian boleh dibuat dengan membina aplikasi yang mudah.



Java Applets
goo.gl/Nf1bQq



Learn Javascript
goo.gl/otWkD1



Langkah 1: Fasa Analisis Masalah

Masalah yang dikenal pasti merupakan satu atur cara untuk mencari luas objek iaitu sama ada bulatan, segi tiga atau segi empat. Atur cara ini perlu menyelesaikan masalah tersebut dengan murid memilih jenis objek, memasukkan input dan output akan dihasilkan berdasarkan objek dan input yang dimasukkan.

Data yang perlu diketahui oleh murid ialah bentuk objek dan juga formula untuk mencari luas objek tersebut. Maklumat lain yang perlu diketahui ialah maksud istilah seperti jejari, tinggi, tapak dan lebar.

Jadual 1.24 Analisis IPO

INPUT	PROSES	OUTPUT
Pilihan objek	Dapatkan objek	
Lebar	Dapatkan lebar	
Tinggi	Semak output gaji staf Kirakan luas objek = $\frac{1}{2} \times \text{lebar} \times \text{tinggi}$	Luas objek



Applet vs Application Java
goo.gl/bf0FD6

Langkah 2: Mereka Bentuk Penyelesaian

Fasa ini boleh dimulakan dengan menulis pseudokod, diikuti dengan melakar carta alir untuk mereka bentuk penyelesaian untuk aplikasi mencari luas objek.

- Menulis pseudokod untuk aplikasi mencari luas objek
- Membina antara muka bergrafik untuk membuat pilihan untuk mencari luas objek, selepas itu taipkan kod atur cara untuk setiap pilihan

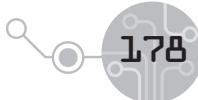
- Mula
- Paparkan "Sila buat pilihan anda" dan nama 3 objek
- Input pilihan
- Jika pilih segi tiga, borang mengira luas segi tiga akan dipaparkan.
- Input lebar dan tinggi
- $\text{Luas} = \frac{1}{2} \times \text{lebar} \times \text{tinggi}$
- Papar luas
- Tamat



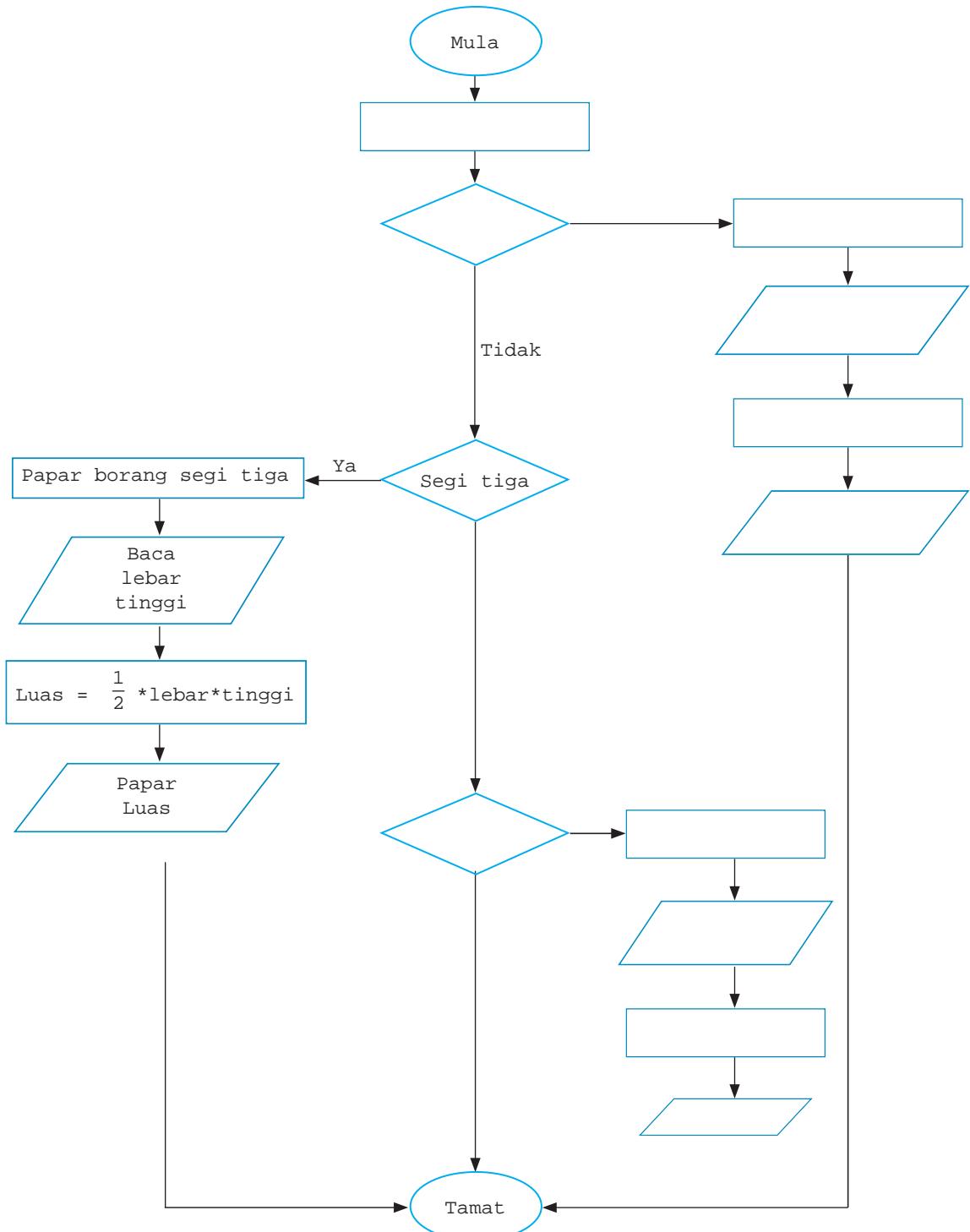
Antara aplikasi yang berjaya dikomersialkan oleh pembangun aplikasi tempatan di Apps Store ialah unXT Sports (permainan) oleh Just Mobile Sdn Bhd, Ganoo Jump (permainan) oleh

Commix Arts Sdn Bhd, E-Sentral (e-book) oleh Xentral Methods Sdn Bhd dan YaHanana Club (hiburan) oleh CouponXcess Sdn Bhd.

Sumber: <http://www.kkmm.gov.my/>



Lengkapkan carta alir di bawah untuk mencari luas objek yang dipilih, contoh untuk luas segi tiga yang diberikan. Selepas itu, uji algoritma mencari luas objek yang dipilih dan catatkan hasil ujian.





Java – Lesson using swing controls
goo.gl/5Ria3Z

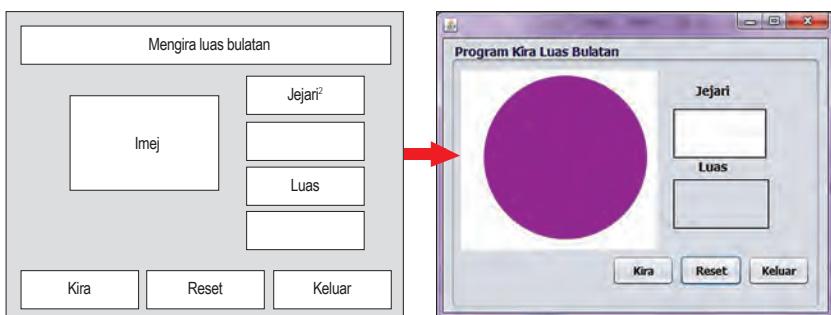


Java – How to use various components
goo.gl/5Ria3Z

c Sediakan antara muka bergrafik untuk aplikasi.

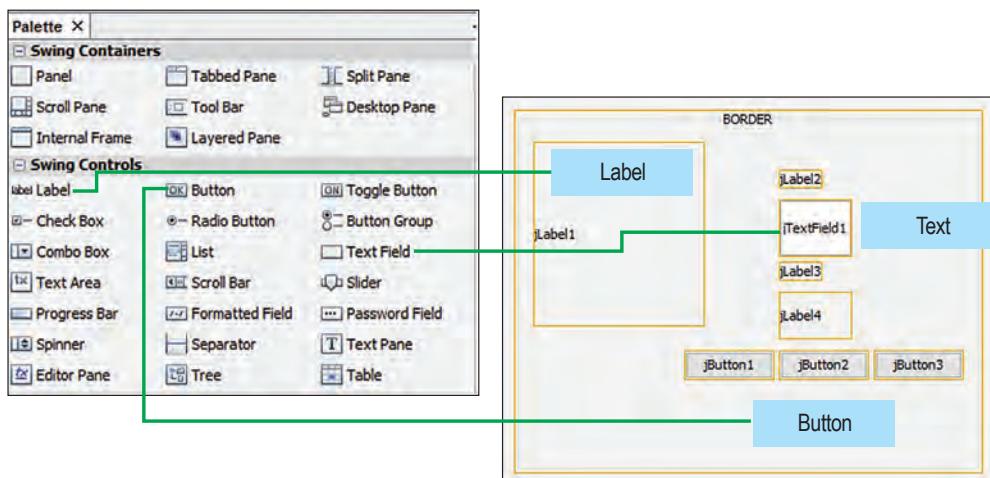
Dengan menggunakan *tools* dari *Swing Controls* yang disediakan, bina antara muka untuk paparan mengira luas objek. *Tools* yang perlu gunakan ialah *Label*, *Button* dan *TextField*. Fungsi *jButton* adalah untuk menyediakan tiga pilihan iaitu “Kira”, “Reset” dan “Keluar”. *Jlabel* digunakan untuk melabel dan memaparkan output dan *jTextField* untuk menginput data.

Setelah mengetahui cara melakar carta alir untuk mereka bentuk penyelesaian yang diperlukan bagi aplikasi menghitung luas objek, maka antara muka bergrafik bolehlah disediakan untuk aplikasi tersebut. Contoh yang ditunjukkan dalam Rajah 1.75(a) merupakan cadangan antara muka bergrafik bagi menghitung luas bulatan. Dengan menggunakan *tools* dari *Swing Controls* yang disediakan, bina antara muka untuk paparan menghitung luas objek. *Tools* yang perlu digunakan ialah *Label*, *Button* dan *TextField*. Fungsi *jButton* adalah untuk menyediakan tiga pilihan, iaitu Kira, Reset dan Keluar. *Jlabel* digunakan untuk melabel dan memaparkan output; dan *jTextField* untuk menginput data.



Rajah 1.75(a) Cadangan antara muka bergrafik

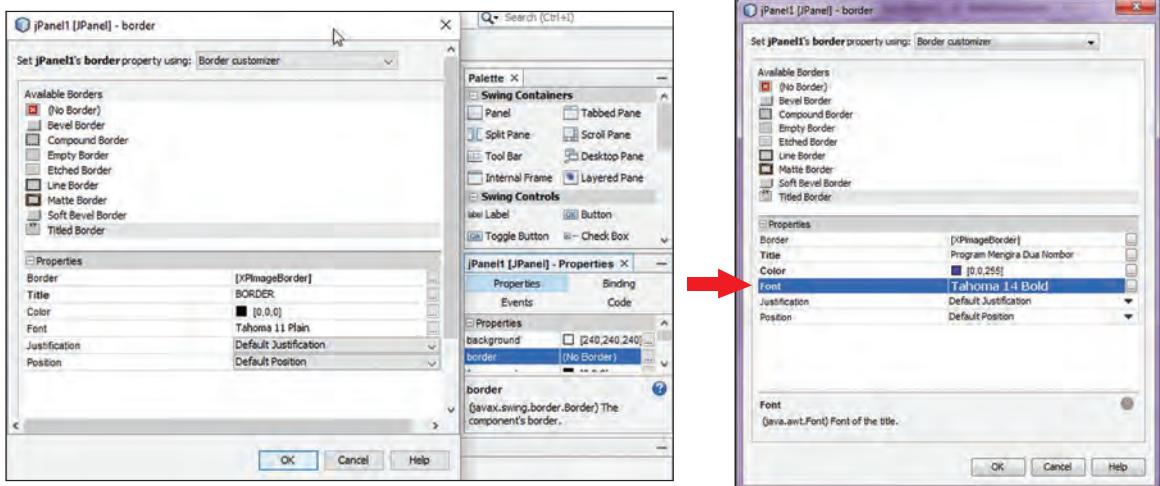
Rajah 1.75(b) menunjukkan ‘tools’ yang digunakan untuk membina antara muka bergrafik seperti yang ditunjukkan dalam Rajah 1.75(a).



Rajah 1.75(b) *Swing Controls* yang digunakan untuk membina antara muka bergrafik



Rajah 1.75(c) menunjukkan cara untuk menukar ‘border’ bagi “antara muka bergrafik” dalam Rajah 1.75(a).



Rajah 1.75(c) Cara menukar ‘border’

Aktiviti

32 Lakaran antara muka bergrafik

Kerja Individu

Dengan menggunakan *jframe* seperti yang ditunjukkan dalam rajah di bawah, lakarkan antara muka bergrafik untuk paparan ‘Menghitung Luas Objek’. Susun semula *Swing Controls* mengikut kreativiti anda. Anda boleh masukkan imej untuk latar belakang *jframe* apabila menghasilkan antara muka ini dalam perisian Java *Netbeans*.

The screenshot shows the Java NetBeans IDE interface. On the left, there is a toolbar with icons for file operations like New, Open, Save, and Print. Below the toolbar is a status bar with the message: 'The Tools>Palette>Swing/AWT Components menu item allows you to modify the content of the Palette.' The main workspace contains a blank JFrame window with an orange border. To the right of the workspace is the 'Swing Components' palette, which lists various Swing controls such as Panel, Split Pane, Internal Frame, Label, Button, etc. At the bottom of the palette is the 'JFrame - Properties' panel, showing properties like defaultCloseOperation (EXIT_ON_CLOSE), title, and background color (240,240,240).



JavaLang *Floatparse* *Float()*
Method
goo.gl/pOENQs

Langkah 3: Pelaksanaan Penyelesaian

Setelah membina reka bentuk yang dikehendaki, proses seterusnya ialah melaksanakan penyelesaian. Murid boleh membina ‘jframe’ di perisian *Java Netbeans* dan menulis kod atur cara yang betul. Berikut ialah cara membina antara muka mencari luas bulatan dan kod atur cara. Cuba uji atur cara yang telah anda bina.

- a Bina “antara muka bergrafik” untuk membuat pilihan untuk mencari luas objek, selepas itu taipkan kod atur cara untuk setiap pilihan.

The screenshot shows a Java application window titled "Sila Buat Pilihan Anda". It contains three radio buttons: "Luas Bulatan" (selected), "Luas Segi Empat Tepat", and "Luas Segitiga". Below the radio buttons are two buttons: "Pilih" and "Keluar". To the right of the window is a code editor displaying Java code:

```

//untuk membolehkan pengguna membuat HANYA satu pilihan
private void KumpulanButang() {
    ButtonGroup bg1 = new ButtonGroup();

    bg1.add(LuasBulatan);
    bg1.add(LuasSegiEmpatTepat);
    bg1.add(LuasSegitiga);

}

//Pengguna hanya boleh membuat HANYA satu pilihan
//memanggil fungsi KiraLuasPoligon()
public KiraLuasPoligon() {
    initComponents();
    KumpulanButang();
}

private void PilihanActionPerformed(java.awt.event.ActionEvent evt) {
    //Jika pengguna nemilih program mengira luas bulatan
    if (LuasBulatan.isSelected()){
        this.dispose(); //Fraparkan sile buat pilihan disembunyikan
        new my.bulatan().setVisible(true); //Program kira luas
        //bulatan dipaparkan
    }
    //Jika pengguna memilih program mengira luas segi empat tepat
    if (LuasSegiEmpatTepat.isSelected()){
        this.dispose();
        new KiraSegiempatTepat().setVisible(true);
    }
    //Jika pengguna memilih program mengira luas segitiga
    if (LuasSegitiga.isSelected()){
        this.dispose();
        new KiraSegi3().setVisible(true);
    }
}

```

Rajah 1.75(d) Antara muka untuk membuat pilihan objek untuk mencari luas dan kod atur cara setiap pilihan

The screenshot shows three windows of a Java application titled "Program Kira Luas Bulatan". The first window (Input) shows a circle icon, a "Jejari" text field containing "12.5", and a "Luas" text field. The second window (Calculation) shows the same interface with the "Luas" field now containing "490.63". The third window (Output) shows the same interface with the "Luas" field now containing "490.63". A green arrow labeled "Input" points from the first window to the second. A green arrow labeled "Output" points from the second window to the third.

Java code for the "KiraActionPerformed" method:

```

private void KiraActionPerformed(java.awt.event.ActionEvent evt) {

    double jejari, luas;
    double pi = 3.14;

    jejari = Float.parseFloat(Jejari.getText()); //Mendapatkan data input
    //dari pengguna

    luas = pi * jejari * jejari; //Sistem melaksanakan proses kira
    //luas bulatan

    //Operasi mengira luas bulatan akan menaparkan output di kotak jawapan
    //Menukarkan jawapan dari jenis data float kepada string
    LuasBulatan.setText(String.valueOf((double) Math.round(luas*100)/100));
}

```

Rajah 1.75(e) Antara muka mengira luas bulatan dan kod atur cara luas bulatan

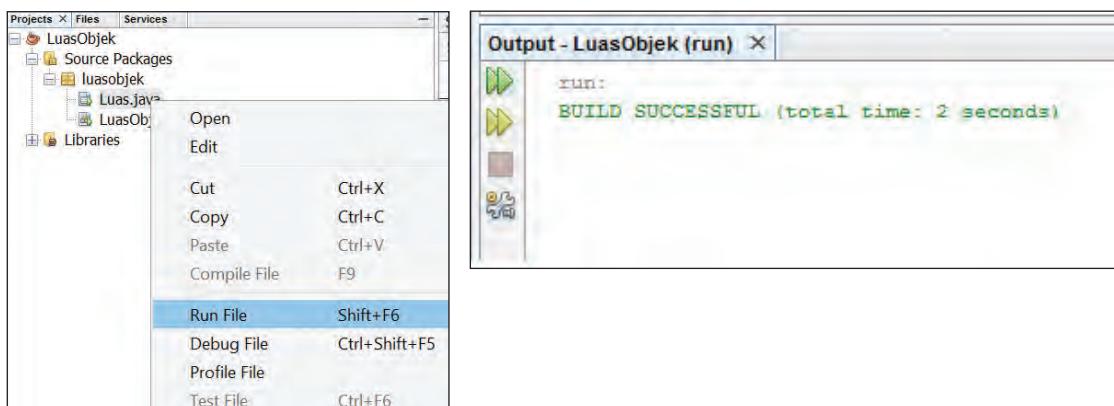
Langkah 4: Menguji dan Menyah ralat

Setelah berjaya membina antara muka dan kod arah cara untuk ketiga-tiga objek, uji setiap program yang telah dibangunkan. Untuk menguji dan menyah ralat yang wujud dalam suatu arah cara yang dibangunkan, langkah-langkah seperti yang berikut perlu dilakukan.

- Masukkan input yang diperlukan, kemudian semak output yang dihasilkan. Jika wujud ralat, kita perlu menghapus/menyah ralat dan membaikinya. Seterusnya, sistem perlu diuji semula untuk memastikan arah cara adalah bebas ralat.
- Uji kesemua unit yang telah dibina, paparkan output aplikasi yang telah dibina. Dapatkah aplikasi yang telah dibina menghasilkan output seperti yang dikehendaki?



How to debug in Java?
goo.gl/gKW2Ts



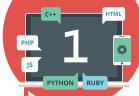
Rajah 1.75(f) Paparan menguji program dan output pengujian yang dijalankan

Langkah 5: Dokumentasi

Dokumentasikan setiap fasa proses mencipta aplikasi mencari luas objek yang dipilih. Pamerkan hasil kerja anda dalam bentuk persembahan dan bincang secara berkumpulan.

FASA	AKTIVITI
Analisis Masalah	
Mereka bentuk penyelesaian	
Melaksanakan Penyelesaian	
Pengujian dan menyah ralat	

PROJEK



Paparan Mesej untuk Papan Notis di Pintu Pagar Sekolah

Kerja Individu



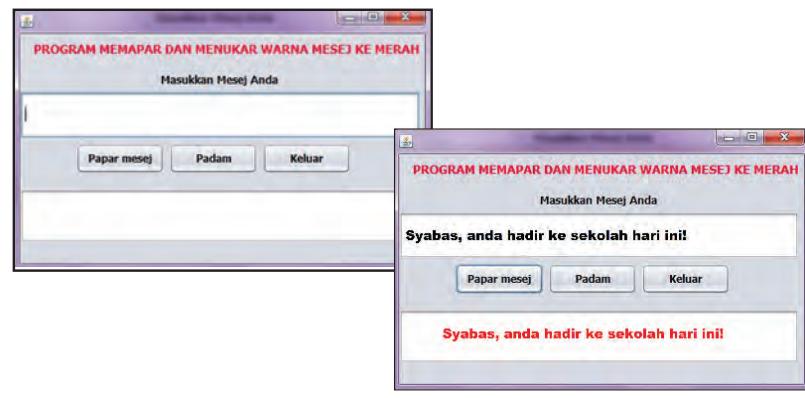
Kerjaya Sains Komputer



Penerangan pengaturcaraan dari segi:

- (a) Kerjaya dalam pengaturcaraan
 - (b) Kebaikan pengaturcaraan
 - (c) Masa depan kerjaya pengaturcaraan
- <https://techprep.fb.com/why/>

Rancang fasa-fasa kitaran hayat pembangunan sistem dengan lengkap berdasarkan rajah yang ditunjukkan di bawah. Kemudian, bina atur cara bebas ralat dan pamerkan hasil kerja anda. Rajah-rajab di bawah boleh digunakan sebagai panduan untuk menyelesaikan projek anda.



PROJEK



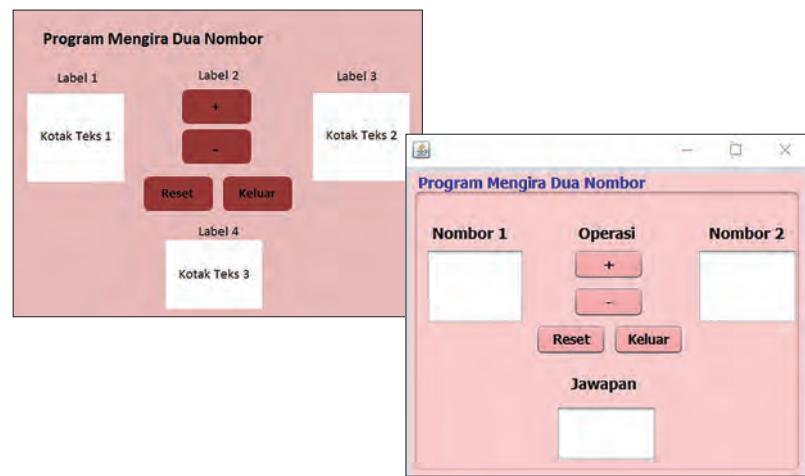
Mengira Dua Nombor untuk Operasi Campur dan Tolak

Kerja Berpasangan



Dapatkan NetBeans dari:
goo.gl/bWL9Cg

Rancang fasa-fasa kitaran hayat pembangunan sistem dengan lengkap berdasarkan rajah yang ditunjukkan di bawah. Kemudian, bina atur cara bebas ralat dan pamerkan hasil kerja anda. Rajah-rajab di bawah boleh digunakan sebagai panduan untuk menyelesaikan projek anda.



Latihan Formatif 1.7

- 1 Berdasarkan pseudokod di bawah, lakar carta alir untuk mereka bentuk penyelesaian kepada masalah di bawah.

```
OUTPUT Siapa nama anda?
INPUT pengguna masukkan nama
SIMPAN input pengguna sebagai pemboleh ubah nama
OUTPUT 'Selamat pagi' + nama
OUTPUT 'Berapa umur anda?'
INPUT pengguna masukkan umur
SIMPAN input pengguna sebagai pemboleh ubah umur
Jika umur >= 60 then
    OUTPUT "Syabas + nama, anda sudah penceren!"
ELSE
    OUTPUT "WAH+NAMA, ANDA MASIH MUDA."
```

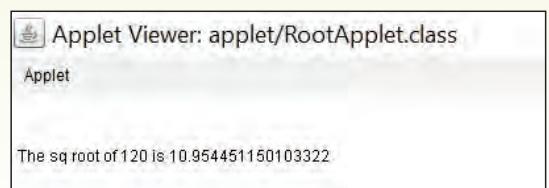
- 2 Rujuk rajah di bawah dan jawab soalan-soalan berikut:

```
public static void main(String[ ] args) {
    int nom1, nom2, jawapan;
    nom1 = 15; nom2 = 40; jawapan = nom1 + nom2;
    System.out.println("Hasil campur = " + jawapan );
```

- (a) Namakan semua pemboleh ubah yang digunakan.
- (b) Nyatakan jenis data yang diwakili oleh pemboleh ubah.
- (c) Apakah paparan yang dikeluarkan apabila program di atas diuji?
- (d) Taipkan kod atur cara di atas dalam Java Netbeans, paparkan hasil operasi nom1-nom2+nom3 jika nom1 = 30, nom2=7 dan nom3=10.

- 3 Tonton video tentang pembinaan aplet mencari punca kuasa dua sesuatu nombor ini di <https://www.youtube.com/watch?v=Hs3Ulbx1GPA>

- (a) Lakarkan algoritma dan jelaskan kod atur cara yang telah digunakan.
- (b) Bina aplikasi di atas dengan perisian Java Netbeans, uji dan nyah ralat yang dijumpai.
- (c) Cuba cari jawapan punca kuasa dua kepada nombor berikut : 120, 975 dan 1 500. Paparkan jawapan anda dalam bentuk seperti di sebelah.
- (d) Bundarkan jawapan anda kepada dua tempat titik perpuluhan dalam atur cara yang dibina.



-  4 Atur cara yang berikut adalah untuk menghasilkan satu aplikasi operasi matematik.

```
private void tolakActionPerformed(java.awt.event.ActionEvent evt) {  
    float nom1, nom2, tolak;//Isytihar pemboleh ubah operasi tolak  
  
    nom1 = Float.parseFloat(nombor1.getText()); //Pengguna memasukkan input  
                                              //nombor 1  
    nom2 = Float.parseFloat(nombor2.getText()); //pengguna memasukkan input  
                                              //nombor 2  
  
    //Lakukan proses untuk operasi tolak  
    tolak = nom1 - nom2;  
  
    //Operasi tolak akan menaparkan output di kotak jawapan  
    //Menukar jawapan dari jenis data float kepada string  
    jawapan.setText(String.valueOf(tolak));
```

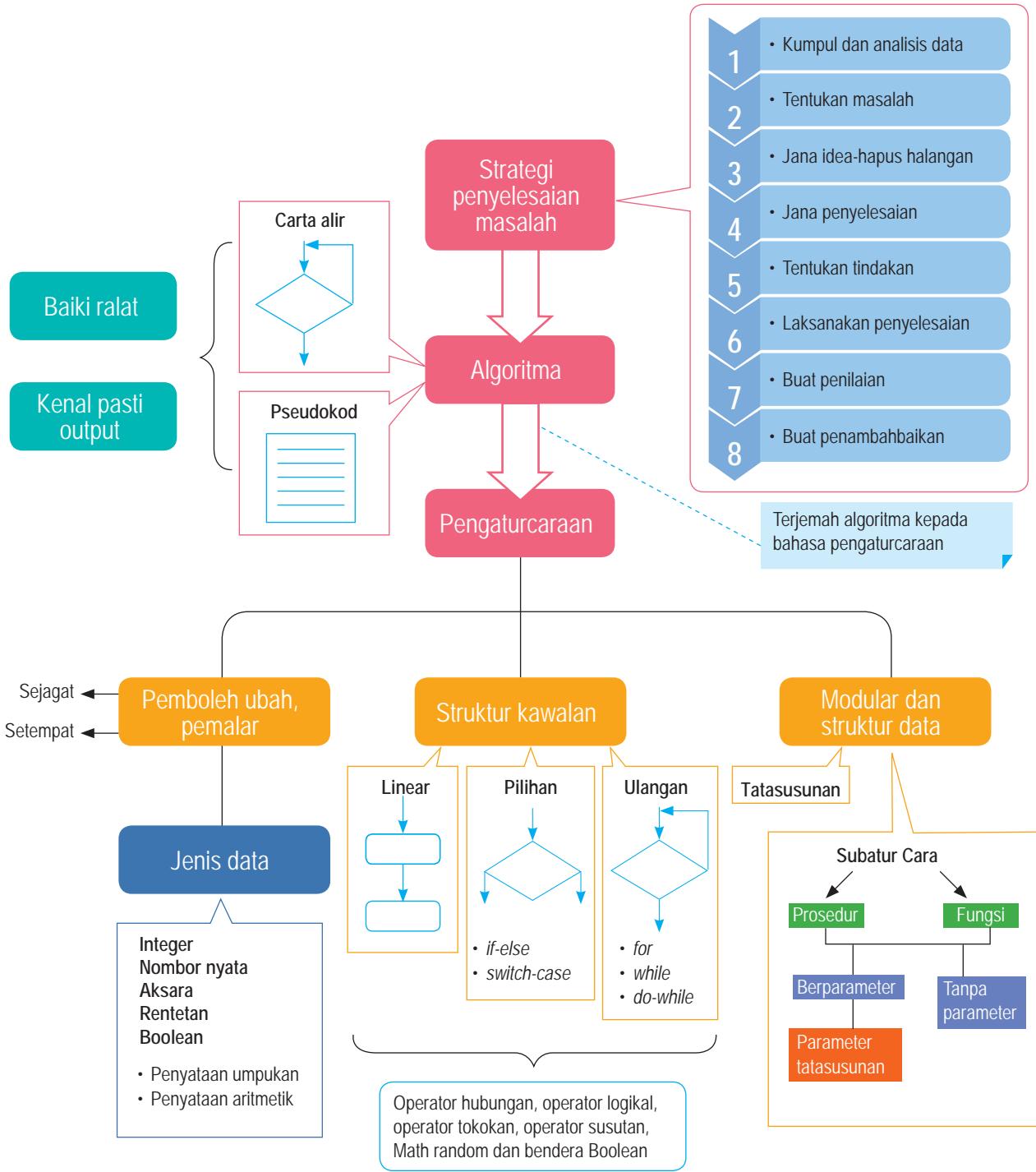
Berdasarkan atur cara tersebut:

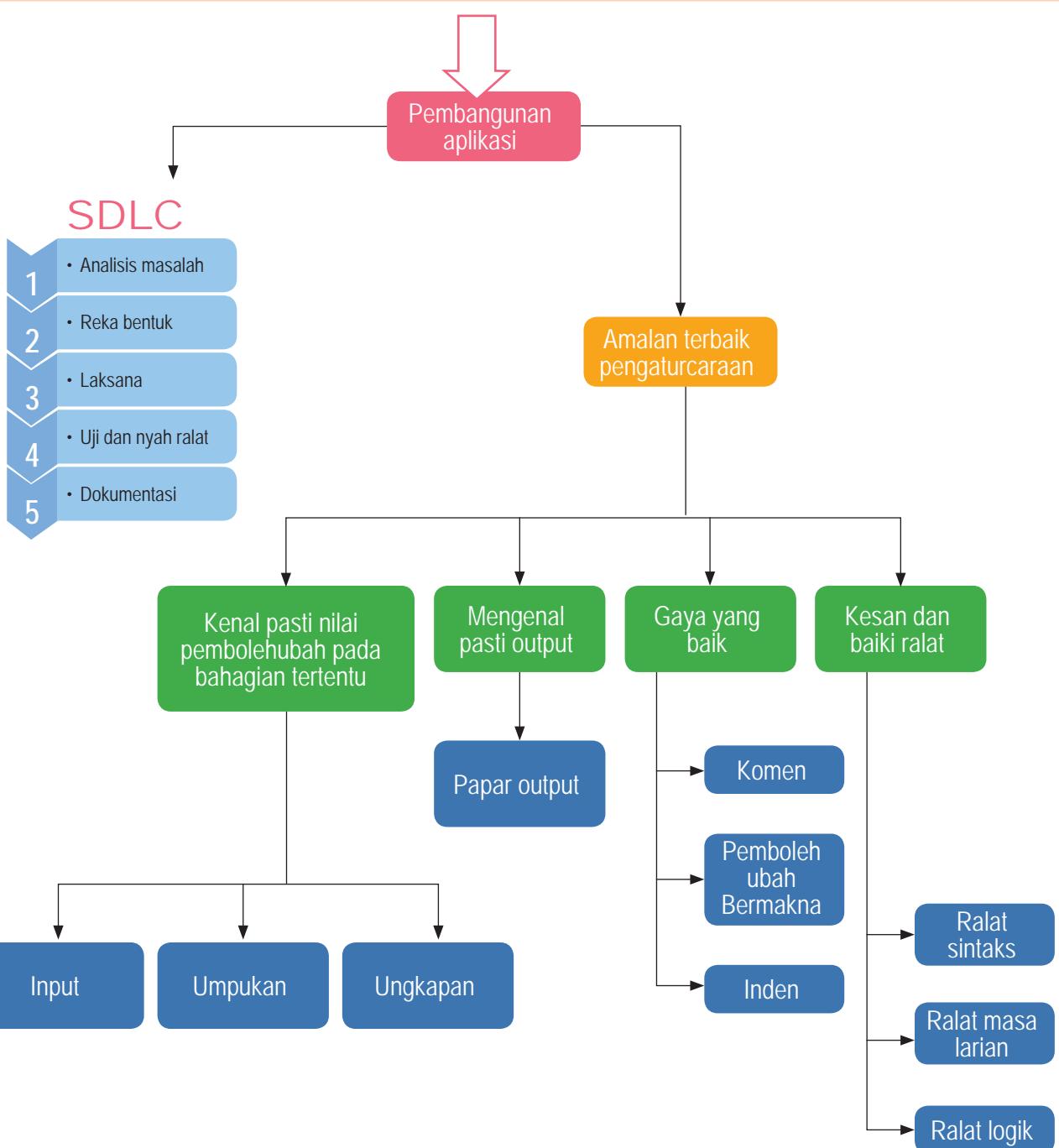
- (a) Namakan pemboleh ubah dalam atur cara ini.
- (b) Lakarkan antara muka bergrafik untuk aplikasi di atas.
- (c) Laksanakan penyelesaian di atas dengan menghasilkan aplikasi bebas ralat.

-  5 Hasilkan satu permainan ringkas yang melibatkan penggunaan nombor atau abjad. Tahap umur yang sesuai untuk aplikasi ini ialah murid-murid prasekolah.



IMBAS SEMULA





LATIHAN SUMATIF

1

Soalan Objektif

Jawab semua soalan.

- 1 Yang manakah pernyataan yang tidak tepat mengenai mengapa perlunya strategi dalam penyelesaian masalah?
 - A Membantu pengembangan sesuatu konsep
 - B Menggalakkan pembelajaran kendiri
 - C Meningkatkan kemahiran berfikir
 - D Mewujudkan komunikasi sehala
- 2 Yang manakah pernyataan yang tepat untuk menerangkan konsep penyelesaian masalah?
 - A Proses mendalami butiran sesuatu masalah untuk mendapatkan satu penyelesaian.
 - B Proses mengkaji butiran sesuatu masalah untuk mendapatkan satu penyelesaian.
 - C Proses mengulang kaji butiran sesuatu masalah untuk mendapatkan satu penyelesaian.
 - D Proses mereka bentuk butiran sesuatu masalah untuk mendapatkan satu penyelesaian.
- 3 Yang manakah pernyataan yang tepat mengenai teknik dalam penyelesaian masalah?
 - I Teknik Leraian
 - II Pengecaman corak
 - III Peniskalaan
 - IV Algoritma
 - A I dan II
 - B I, II dan III
 - C II, III dan IV
 - D Semua di atas
- 4 Apakah ciri-ciri yang tepat bagi algoritma?
 - I Arah terperinci
 - II Butiran jelas
 - III Mempunyai batasan
 - IV Boleh dilaksanakan
 - A I, II dan III
 - B II, III dan IV
 - C III dan IV
 - D Semua di atas

- 5 Apakah fungsi bagi struktur kawalan pilihan?
 - A memberikan perisian komputer keupayaan untuk membuat keputusan berasaskan syarat yang telah ditentukan pengatur cara.
 - B memberikan pengguna komputer keupayaan untuk membuat keputusan berasaskan syarat yang telah ditentukan pengatur cara.
 - C membolehkan perisian komputer membuat penilaian berasaskan syarat yang telah ditentukan oleh pengatur cara.
 - D membolehkan perisian komputer membuat rujukan berasaskan syarat yang telah ditentukan oleh pengatur cara.
- 6 Apakah yang dimaksudkan dengan amalan terbaik dalam pengaturcaraan?
 - A Pengatur cara dapat menjalankan atur cara tanpa sebarang ralat.
 - B Pengatur cara dapat menulis kod atur cara dengan baik.
 - C pengatur cara dapat mempraktikkan amalan-amalan yang biasa diikuti untuk menghasilkan atur cara yang baik.
 - D pengatur cara dapat menghasilkan atur cara yang bebas ralat yang senang untuk difahami.
- 7 Yang manakah antara kata kunci yang berikut digunakan untuk mengisyiharkan dan memperuntukkan ruang simpanan bagi satu atau lebih boleh ubah?
 - A int
 - B final
 - C Class
 - D Public
- 8 Maklumat di bawah merujuk nilai beberapa boleh ubah boolean.

A = False
B = True
C = True



Yang manakah antara berikut memberikan keputusan False?

- A C Or (A And Not C)
- B (B Or C) Or Not A
- C Not A Or Not(B Or Not C)
- D Not C Or (A And (B Or C))

9 Pilihan yang berikut adalah benar berkenaan Do... While loop dan Do... Until loop, KECUALI

- A Do... While juga dipanggil gelung praujian
- B Do... Until akan mengulang blok arahan dalam kekerapan tertentu
- C Do... Until akan memeriksa syarat setelah blok arahan diulang



D Do... While akan hanya memproses arahan dalam gelung apabila syarat yang diujui memberikan keputusan palsu

10 Fasa analisis pembangunan perisian melibatkan aktiviti ...

- A menulis perisian menggunakan program seperti Java.
- B mengumpulkan keperluan pengguna (user requirements) berkenaan fungsi program yang dikehendaki.
- C menghasilkan pelan rapi perihal bagaimana perisian dibangunkan dapat memenuhi keperluan pengguna.
- D semua di atas

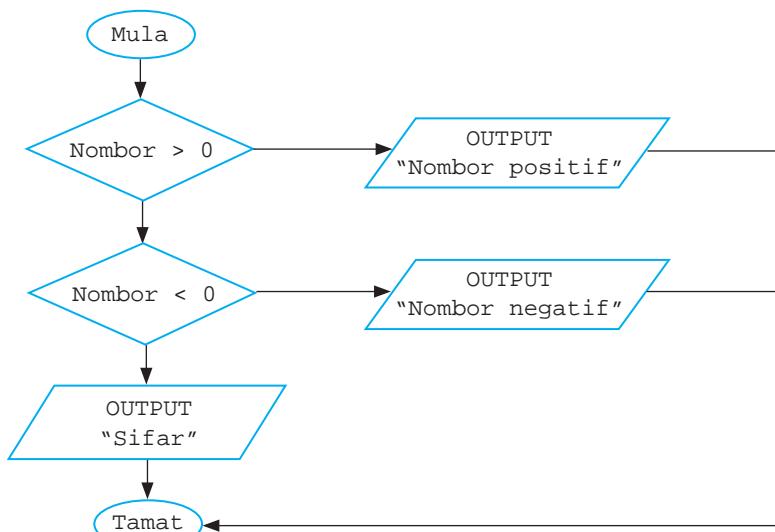


Soalan Struktur

Jawab **semua** soalan.

1 Tulis kod pengaturcaraan Java untuk melaksanakan algoritma yang berikut. Kod atur cara ini boleh menerima data input dari papan kekunci jenis double.

Carta alir



2 Tentukan output bagi atur cara yang berikut:

```
public class Tutorial14 {  
    public static void main(String[] args) {  
        String x = "22";  
        String y = "88";  
        System.out.println(x + y);  
    }  
}
```

3 Apakah output yang mungkin bagi atur cara di bawah?

```
public class Guna_While {  
    public static void main(String[] args) {  
        int i = 0;  
        while (i < 5){  
            System.out.println("Selamat datang Ke Pengaturcaraan  
JAVA");  
  
            i++;  
        }  
    }  
}
```

4 Apakah fungsi struktur kawalan ulangan yang melibatkan *Math.random()*?

Penyelesaian Masalah

Jawab **semua** soalan.

1 Tulis atur cara Java menggunakan struktur kawalan *switch-case* untuk menerima input nombor jenis integer. Kemudian, atur cara itu akan memaparkan output seperti di bawah.



Input	Output
1	Kesihatan anda baik
2	Kesihatan anda memuaskan
3	Anda tidak sihat
default	Anda tidak memilih 1, 2 atau 3!



2 Hasilkan satu atur cara bebas ralat untuk mengira BMI (*Body Mass Index*). Atur cara anda mestilah boleh menerima berat dan tinggi dalam *double* serta dapat memaparkan keputusan seperti di bawah.

Bacaan BMI	Keputusan
<= 18.5	Kurang berat
= 18.5 – 24.9	Normal
= 25 – 29.9	Lebih berat
>= 30	Obesiti

- 3** Berikan kod atur cara untuk melaksanakan arahan yang berikut:



Arahan	Kod Atur Cara
Keluar sistem	
Reset kotakTeks1	
Ulangan integer i dari 0 hingga 10	

- 4** Hasilkann sebuah program yang boleh memaparkan jadual pendaraban atau Jadual Sifir. Gunakan *JOptionPane* untuk memaparkan output seperti di bawah. Atur cara mestilah boleh memaparkan tajuk jadual “JADUAL SIFIR” dan dapat menghitung dari sifir 1 hingga sifir 9.



- 5** Al Rahmat ialah sebuah kedai runcit di Taman Iskandar. Untuk memudahkan pengiraan bagi barang-barang runcit, pemilik kedai tersebut, iaitu Encik Rahmat meminta anda membantunya untuk membina sebuah perisian bagi menghitung jumlah harga runcit. Antara muka sudah disiapkan dengan nama-nama kontrol seperti rajah yang berikut. Data bagi input kuantiti barang dan harga barang boleh berubah dan dapat dimasukkan dari papan kekunci.



- 6** Pelbagai jenis model kitaran hayat pembangunan sistem yang didapati untuk memastikan sesuatu projek dilaksanakan dengan berjaya. Dengan menggunakan peta i-Think, jawab soalan-soalan di bawah.
- Nyatakan dan jelaskan ciri fasa-fasa dalam SDLC dengan menggunakan peta buih.
 - Pilih **dua** model SDLC yang anda ketahui, jelaskan kebaikan dan kelemahan setiap satu dengan menggunakan peta pokok.
 - Jelaskan ciri-ciri penting dalam fasa-fasa model air terjun dengan menggunakan peta alir.

BAB 2

PANGKALAN DATA

Salah satu faktor kejayaan syarikat gergasi IT seperti *Google*, *Amazon* dan *Facebook* ialah kemampuan platform teknologi untuk menampung jumlah data yang sangat besar dan sentiasa bertambah. Cabaran utama bagi syarikat ini adalah untuk menampung kesemua data tersebut dan mengekalkan prestasi pencarian serta pengurusan data. Jika data diumpamakan sebagai satu perkataan, proses menyimpan dan mencari sesuatu perkataan dalam sebuah buku nota kecil tidaklah sukar. Sekarang, cuba anda bayangkan apabila buku tersebut disimpan di dalam perpustakaan sekolah anda. Bolehkah anda mencari perkataan tersebut dalam masa yang singkat?

Dalam bab ini, anda akan mempelajari:

- » Pangkalan data hubungan
- » Reka bentuk pangkalan data hubungan
- » Pembangunan pangkalan data hubungan
- » Pembangunan sistem pangkalan data

Fikirkan...

- Apakah contoh-contoh penggunaan pangkalan data dalam kehidupan harian kita?
- Bagaimanakah teknologi pangkalan data diuruskan supaya kualiti penggunaan sistem komputer memberikan kepuasan kepada pengguna?
- Berapa ramaikah pengguna yang boleh mengakses pangkalan data dalam satu-satu masa?
- Bagaimanakah pangkalan data direka bentuk?



STANDARD PEMBELAJARAN

2.1.1 Menjelaskan kepentingan integriti data, ketekalan data dan kelebihan data

2.1.2 Membezakan model-model pangkalan data: hierarki, rangkaian, hubungan (*relational*), berorientasi objek (*object oriented*)

2.1.3 Mengenal pasti entiti, atribut, set hubungan dan kekardinalan (satu ke satu, satu ke banyak dan banyak ke banyak) berdasarkan keperluan pengguna

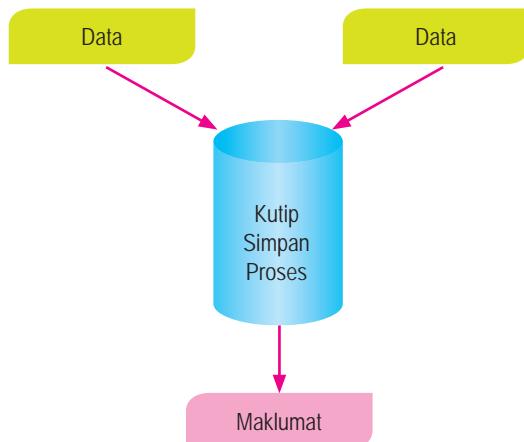
Pangkalan Data Hubungan

Data ialah sumber komputer untuk membuat perancangan dan keputusan. Jumlah data yang perlu diproses oleh komputer adalah besar. Tambahan pula, data ini perlu dikongsi dengan efisien dan selamat. Pengendalian data dalam jumlah yang besar amat mencabar kerana data yang disimpan mungkin mengandungi ralat, berulang ataupun tidak dikemas kini dengan lengkap.

Data yang diproses ini akan menjadi maklumat yang berguna kepada pengguna. Jadual 2.1 menerangkan definisi data dan maklumat manakala Rajah 2.1 menunjukkan konsep hubungan data dan maklumat.

Jadual 2.1 Pengertian data dan maklumat serta contoh

	Definisi	Contoh
Data	<ul style="list-style-type: none"> Data merupakan himpunan fakta mentah mengenai sesuatu benda, kejadian, orang atau entiti yang boleh terdiri daripada perkataan, angka atau gambar. Data boleh menjadi sesuatu fakta yang bermakna atau tidak bermakna sehingga fakta-fakta tersebut diproses. 	Markah ujian bulanan bagi mata pelajaran Sejarah untuk setiap murid di dalam kelas 4 Dinamik.
Maklumat	<ul style="list-style-type: none"> Maklumat merupakan hasil daripada pengumpulan, pemprosesan dan penganalisisan data yang boleh digunakan untuk membuat sesuatu keputusan. Maklumat yang dihasilkan ini akan menjadi lebih bermakna dan mudah untuk difahami oleh seseorang. 	Purata markah ujian bulanan bagi mata pelajaran Sejarah untuk kelas 4 Dinamik.



Rajah 2.1 Hubungan antara data dan maklumat



Konsep Pangkalan Data Hubungan
goo.gl/p1MMVM

Untuk memastikan integriti data yang disimpan, data perlu disimpan dalam komputer secara sistematik supaya mudah dicapai apabila diperlukan. Cuba bayangkan, tanpa simpanan digital, jumlah kertas dokumen yang diperlukan untuk menyimpan rekod mungkin memenuhi kabinet sehingga satu ruang bilik menjadi penuh.

Sebelum konsep pangkalan data diperkenalkan, data dalam sistem berkomputer disimpan dengan menggunakan sistem fail biasa. Sistem fail biasa mempunyai banyak kelemahan untuk menguruskan data dalam jumlah yang banyak dan kompleks. Untuk menjadikan pengurusan data lebih cekap dan efisien, teknologi komputer digunakan untuk memudahkan proses penyimpanan, pengeluaran dan pengemaskinian data. Hal ini mencetuskan evolusi sistem pemprosesan data, lihat Rajah 2.2.



Rajah 2.2 Evolusi pengurusan data

1 Pemprosesan Manual tidak melibatkan penggunaan komputer. Pada peringkat ini, tenaga manusia diperlukan untuk mengendalikan data. Data yang telah diproses dimasukkan ke dalam fail dan kabinet mengikut aturan tertentu. Kelemahan pemprosesan manual adalah banyak tenaga buruh diperlukan, ruang yang besar untuk pemprosesan dan simpanan dan pencarian yang agak lambat apabila data semakin bertambah.



Rajah 2.3 Sistem pemprosesan fail

2 Sistem Pemprosesan Fail menggunakan teknologi komputer yang menyimpan dokumen-dokumen dalam bentuk fail digital. Hal ini membolehkan aktiviti-aktiviti manual sebelum ini diproses secara automatik menggunakan sistem pengoperasian dan aplikasi komputer. Kelemahan utama Sistem Pemprosesan Fail adalah kesukaran

Tahukah Anda ?

Data yang dimasukkan ke dalam komputer disimpan dalam memori dan boleh hilang apabila bekalan elektrik diputuskan. Oleh itu, data perlu disimpan sebagai fail dalam cakera keras dan dapat dicapai apabila diperlukan. Simpanan menggunakan fail mempunyai banyak kekurangan seperti sukar mencapai data yang diperlukan dalam masa yang singkat, banyak salinan yang serupa, dan sukar untuk mengemaskinikan data.

DBMS-Database Management System
goo.gl/pMrvk5

mengesan salinan data yang sama dalam fail berlainan dan data lambat untuk dikemas kini.



Rajah 2.4 Sistem Pangkalan Data



Tutorial pembinaan gambar rajah perhubungan entiti (ERD)
goo.gl/J02rGW

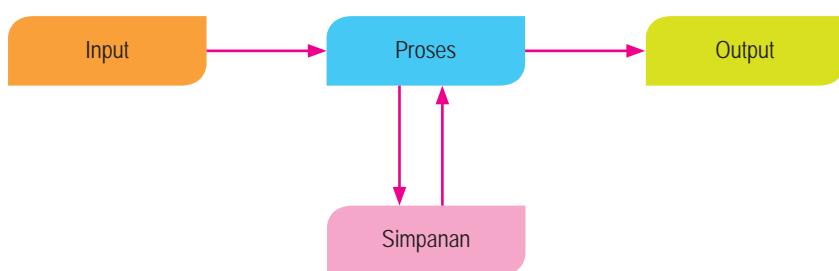
3 Sistem Pangkalan Data adalah lebih canggih daripada sistem pemprosesan fail kerana boleh menyimpan jumlah data yang banyak dalam ruang simpanan yang jauh lebih kecil. Sistem pangkalan data ialah sistem perisian yang menguruskan pangkalan data. Pangkalan data ialah koleksi data logikal yang berkaitan dan strurnya membolehkan kandungannya dikeluarkan, diuruskan dan dikemas kini dengan mudah. Terdapat beberapa jenis model pangkalan data yang biasa digunakan. Model pangkalan data ialah penyusunan secara konseptual suatu pangkalan data dan merupakan satu cara mentakrif dan menggunakan data dalam satu pangkalan data. Lihat Rajah 2.5.

Pangkalan Data (PD) menyimpan data dengan strurnya sekali. Ruang memori komputer dapat digunakan secara minimum meskipun data yang disimpan amat besar dan banyak.

Tahukah Anda ?

Sistem Pangkalan Data yang moden mengandungi ciri-ciri yang berikut:

- Mengandungi entiti dunia sebenar
- Jadual berdasarkan hubungan
- Pengasingan data dan aplikasi
- Kurang pertindihan
- Mempunyai ketekalan data
- Mempunyai bahasa pertanyaan
- Keselamatan
- Pelbagai paparan



Rajah 2.5 Gambar rajah blok sistem komputer

Oleh sebab itu, kebanyakan organisasi, sama ada kecil atau besar, menggunakan pangkalan data untuk menyimpan, memanipulasi dan mencapai data di kebanyakan organisasi. Sedar ataupun tidak, banyak maklumat individu dan aktiviti yang dilakukan disimpan dalam pangkalan data di syarikat-syarikat mahupun organisasi kerajaan.

- Contoh aspek kehidupan yang melibatkan simpanan pangkalan data:
- Maklumat setiap rakyat disimpan dalam pangkalan data kerajaan.
 - Universiti mempunyai pangkalan data untuk menyimpan data pensyarah, kursus dan pelajar.
 - Apabila membeli tiket penerbangan, maklumat penumpang dan penerbangan disimpan dalam PD.
 - Pekerja di pasar raya mengimbas setiap barang yang dijual untuk mencapai harga barang daripada pangkalan data.



*Top Ten Enterprise for
Database System*
goo.gl/VCJaV3

2.1.1 Kepentingan Integriti Data, Ketekalan Data dan Kelewanan Data

Pangkalan data bukan koleksi data semata-mata. Pangkalan data merupakan satu koleksi data yang disimpan dalam format piawaian (*standard*) dan direka khusus supaya mampu untuk berkongsi data dengan banyak pengguna secara efisien.

- 1 Format piawai membolehkan data disimpan dan dicapai kembali daripada mana-mana perkakasan dan sistem maklumat.
- 2 Data yang disimpan dalam pangkalan data boleh digunakan kembali oleh pengguna yang berbeza melalui sistem yang berbeza bagi tujuan yang berlainan. Sebagai contoh, data warganegara dalam pangkalan data Jabatan Pendaftaran boleh digunakan semula apabila warganegara memohon pasport ataupun memperbaharui lesen merandum.
- 3 Simpanan data dalam pangkalan data adalah efisien kerana format dan struktur data hubungan dikhatusukan untuk simpanan sistematik dan capaian pantas serta fleksibel. Faktor lain ialah perkakasan komputer pelayan yang menyokong operasi pangkalan data, mempunyai spesifikasi tinggi bagi capaian kepada pangkalan data tersebut.

Umumnya, pangkalan data dikhatusukan untuk menyimpan data manakala tafsiran data sebagai maklumat yang disempurnakan oleh sistem-sistem komputer yang dihubungkan kepada pangkalan tersebut. Data yang baik diperlukan untuk menghasilkan maklumat yang tepat dan bermakna. Ciri-ciri data yang disimpan dalam pangkalan data mesti mencapai kualiti yang berikut:

- Integriti data
- Ketekalan data
- Kelewanan data



Kementerian Pendidikan Malaysia juga membangunkan satu pangkalan data untuk memudahkan para guru memasukkan semua maklumat murid dalam pangkalan data tersebut. Kehadiran murid juga dimasukkan melalui sistem pangkalan data ini, dan memudahkan para guru untuk mengetahui para murid yang mengalami masalah tidak hadir ke sekolah. Malah, pelbagai borang yang berkaitan dengan guru dan murid juga boleh dimuat turun melalui sistem pangkalan data ini.

Integriti Data (*Data integrity*)

- Integriti data bermaksud kesempurnaan, ketepatan dan kesahan data (*validation*) serta merujuk ketepatan data yang sah melalui keseluruhan kitaran hayat data.
- Untuk memastikan integriti data, input bukan sahaja mesti tepat tetapi mematuhi peraturan yang telah ditetapkan.
 - 1 Integriti entiti: Jadual mempunyai sekurang-kurangnya satu atribut data yang unik bagi setiap rekod. Contoh: Nombor murid mesti unik.
 - 2 Integriti rujukan: Rekod-rekod dalam dua jadual yang berbeza boleh dihubungkan melalui perkongsian atribut data yang sama. Sekiranya tiada hubungan antara rekod, atribut tersebut dibiarkan kosong. Sekiranya terdapat hubungan antara dua rekod tersebut, atribut dalam kedua-dua jadual berkongsi nilai data yang sama.
 - 3 Integriti domain: Atribut-atribut data dalam jadual mestilah betul dengan berdasarkan suatu domain masalah.
- Kenapa perlunya integriti data?
 - 1 Data yang lemah integriti mungkin korup atau hilang. Data yang korup ialah data yang terubah (secara tidak sengaja) daripada nilai asalnya.
 - 2 Integriti data bertujuan untuk mengelakkan perubahan yang tidak sengaja kepada maklumat. Oleh itu, integriti data adalah penting untuk memastikan tiada ralat dalam kemasukan data.



Uji Minda

Berdasarkan Contoh 1, bina suatu jadual yang mempunyai nombor pendaftaran peserta Kem Kepimpinan Pengawas dan nama peserta. Semak jadual anda dan pastikan integriti data anda terjamin.

Contoh (1)

Jika seorang murid didaftarkan dengan nombor pendaftaran murid, 123/16, pangkalan data sepatutnya tidak membenarkan pendaftaran murid lain dengan nombor pendaftaran yang sama (123/16). Ini menjamin bahawa setiap rekod adalah unik dengan dengan mematuhi peraturan integriti.

Integriti data memastikan tiada nombor pendaftaran yang sama didaftarkan untuk murid yang berbeza

Nombor Pendaftaran	Nama
123/16	Adam bin Ali
124/16	Yap Ling Tzi
123/16	Fatihah bt Roslan

Contoh situasi integriti data yang tidak dipatuhi



Ketekalan Data (Data Consistency)

- Ketekalan merujuk konsistensi ataupun keseragaman data yang akan mempengaruhi kebolehpercayaan data.
- Untuk memastikan ketekalan data, proses kemas kini salinan data di semua lokasi simpanan perlu dilakukan dengan rapi.
- Kepentingan ketekalan data:
 - 1 Untuk memastikan data yang diperoleh boleh dipercayai dan tidak dipersoalkan.
 - 2 Untuk memastikan tiada isu ketekalan data timbul seandainya data yang sama berulang di beberapa lokasi simpanan.



Structuring Data for Strong Consistency
goo.gl/Pw1FzQ

Contoh (2)

Tan mencatat nombor-nombor telefon pelanggannya di dalam dua buah buku iaitu diari dan buku nota secara berasingan. Sekiranya nombor telefon pelanggannya bertukar, Tan perlu mengemaskinkan nombor tersebut pada kedua-dua buah buku yang digunakan. Apakah yang terjadi sekiranya Tan hanya mengemaskinkan nombor telefon pelanggan yang berubah hanya pada sebuah buku sahaja?

Penyelesaian:

Disebabkan Tan hanya mengemaskinkan nombor telefon pelanggannya pada sebuah buku sahaja, maka data nombor telefon tersebut hilang ketekalannya.

Kelewanhan Data (Data Redundancy)

- Kelewanhan data merujuk pertindihan data yang berpunca daripada salinan data yang berulang tetapi di lokasi yang berlainan.

Contoh:

Rajah 2.6 menunjukkan rekod murid dalam Sistem Maklumat Murid. Cuba lihat pada baris yang berwarna merah jambu. Kad pengenalan murid dan Nama murid adalah sama tetapi No. telefon yang berbeza wujud sebanyak tiga kali di dalam sistem fail. Keadaan inilah yang dipanggil kelewanhan data.



IC Murid	Nama Murid	No. Telefon
000405-11-5434	Alia bt Roslan	03-61402987
000213-03-5675	Ah Chong	03-77876789
000607-14-4343	Suraj A/L Ramu	03-43543456
000405-11-5434	Alia bt Roslan	03-61402997
001121-14-2312	Tan Mei Ling	03-69104356
000405-11-5434	Alia bt Roslan	03-61403997



Redundant
goo.gl/zchUUw

Rajah 2.6 Kelewanhan data dalam sistem fail



How to plan and manage data centre redundancy?

- Apakah yang terjadi sekiranya kelebihan data ini tidak diatasi?

 - 1** Kelebihan menggunakan lebihan sumber simpanan komputer dan oleh itu kelebihan data perlu diminimumkan.
 - 2** Kemas kini data yang lewah memakan banyak masa dan mungkin tidak menyeluruh.
 - 3** Menyebabkan data menjadi korup.
 - 4** Menjadikan pangkalan data tersebut kurang efisien.

Antara tujuan mengelakkan kelebihan data adalah untuk,

- 1 mengelakkan pertambahan saiz pangkalan data yang tidak perlu.
(Pertindihan data yang sama berulang-ulang).
 - 2 mengelakkan data yang tidak konsisten.
 - 3 mengelakkan data menjadi tidak efisien.
 - 4 mengelakkan data menjadi korup iaitu tidak boleh digunakan lagi.



33

Kerja Berkumpulan

Integriti data, Ketekalan data dan Kelewanhan data

Lakukan aktiviti ini bersama-sama di dalam kelas.

Alat yang diperlukan: Tiga keping kertas, pensel, pemadam.

Bahagian 1

Langkah-langkah:

- 1** Ambil sekeping kertas yang baharu dan lukis jadual berikut setiap satu:



- 2** Pastikan anda mempunyai bilangan baris kosong yang mencukupi.
- 3** Cari lima orang rakan di dalam kelas dan minta butiran untuk diisikan dalam jadual.
- 4** Semak jadual anda.
- 5** Bincangkan dari segi integriti, kelebihan dan ketekalan data.
 - (a) Adakah terdapat data yang berulang?
 - (b) Adakah terdapat kemungkinan dua orang murid mempunyai nombor pendaftaran murid yang sama?
 - (c) Adakah terdapat kemungkinan dua atau lebih murid mempunyai nama tingkatan dan nama guru tingkatan yang sama?
 - (d) Katakan guru tingkatan telah diganti. Berapa kaliakah kemas kini perlu dibuat? Adakah terdapat sebarang kemungkinan ketekalan data terganggu?
 - (e) Katakan nama tingkatan ditulis sebagai 4 Sains, Empat sains, ataupun 4 sains di baris-baris yang berbeza. Adakah ini masalah integriti, kelebihan atau ketekalan?

Bahagian 2

Ulang aktiviti di atas menggunakan reka bentuk jadual yang baharu.

- 1** Gunakan dua keping kertas yang baharu. Lukis jadual berikut:

Nombor pendaftaran murid	Nama murid	Nama tingkatan

Nama tingkatan	Nama guru tingkatan

- 2** Salin data-data ke dalam jadual-jadual yang baharu.
- 3** Bincangkan soalan berikut:
 - (a) Bagaimanakah data-data yang diagihkan kepada dua jadual dikaitkan semula? Adakah ini integriti, kelebihan ataupun ketekalan?
 - (b) Berbanding penggunaan reka bentuk jadual sebelumnya, adakah kelebihan sudah berkurang? Adakah kelebihan ini dapat dikurangkan lagi?
 - (c) Katakan nama guru tingkatan telah diganti. Apakah yang perlu anda lakukan untuk mengemaskinikan maklumat ini? Adakah reka bentuk jadual baharu membantu mengurangkan ketekalan data?

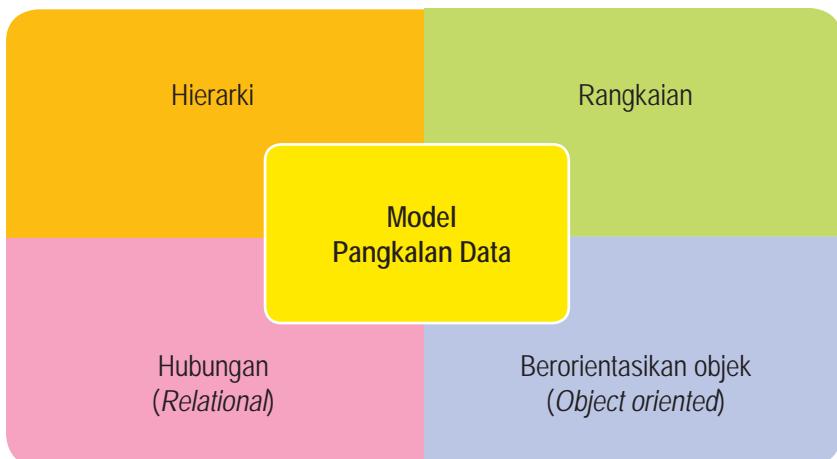
Tahukah Anda ?

Data dalam pangkalan data Jabatan Pendaftaran Negara juga boleh diakses dan dicapai oleh Jabatan Imigresen serta Jabatan Pengangkutan Jalan. Hal ini bermaksud data tentang kad pengenalan kita boleh digunakan semula semasa kita ingin memohon pasport atau lesen memandu!



2.1.2 Model-model Pangkalan Data

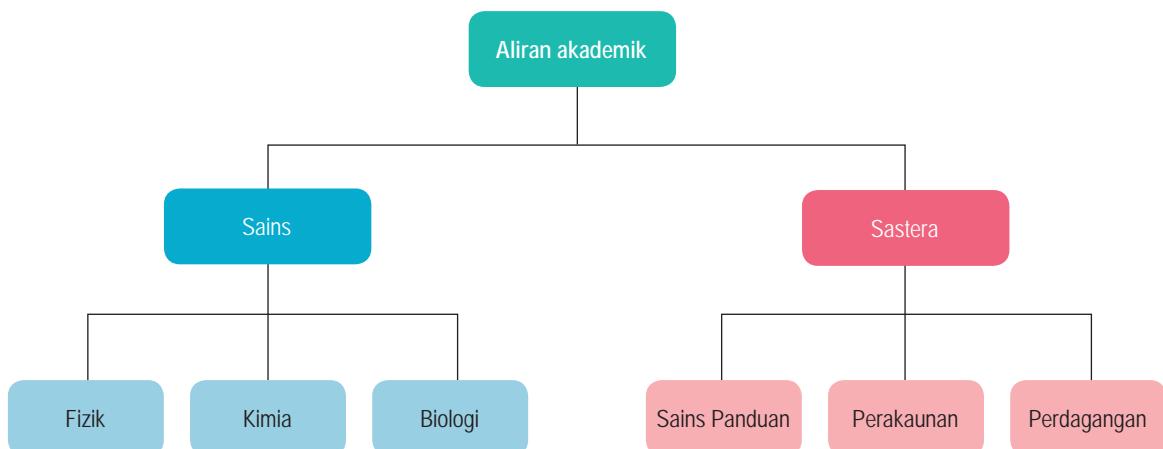
Sistem pangkalan data menggunakan model pangkalan data tertentu sebagai strukturnya. Terdapat beberapa jenis model pangkalan data yang biasa digunakan. Model pangkalan data ialah penyusunan secara konseptual suatu pangkalan data dan merupakan satu cara mentakrif dan menggunakan data dalam satu pangkalan data. Lihat Rajah 2.7 di bawah.



Rajah 2.7 Model-model Pangkalan Data

Terdapat empat model pangkalan data dengan kelebihan masing-masing iaitu hierarki, rangkaian, hubungan dan berorientasikan objek. Model pangkalan data hubungan adalah model yang paling popular untuk dirujuk sebagai contoh pangkalan data.

Model Pangkalan Data Hierarki



Rajah 2.8 Contoh model pangkalan data hierarki

Model pangkalan data hierarki ialah model terawal semenjak tahun 1950 dan sering digunakan dalam sistem pengurusan maklumat keluaran syarikat IBM. Dalam model pangkalan data hierarki, data disusun dalam struktur pokok seperti dalam Rajah 2.8. Model ini sesuai digunakan sekiranya suatu entiti mempunyai hubungan satu **induk (parent)** dengan satu atau lebih entiti **anak (child)**. Model ini tidak menyokong hubungan banyak entiti induk kepada banyak entiti anak. Mungkin disebabkan itu, model ini tidak banyak digunakan.

Contoh:

Katakan sebuah pangkalan data berdasarkan model pangkalan data hierarki diperlukan untuk menyimpan rekod data entiti-entiti seperti Sekolah, Kelas, Murid, Guru dan Subjek. Pertama sekali, takrifkan hubungan antara entiti-entiti tersebut melalui model hierarki, lihat Rajah 2.9.

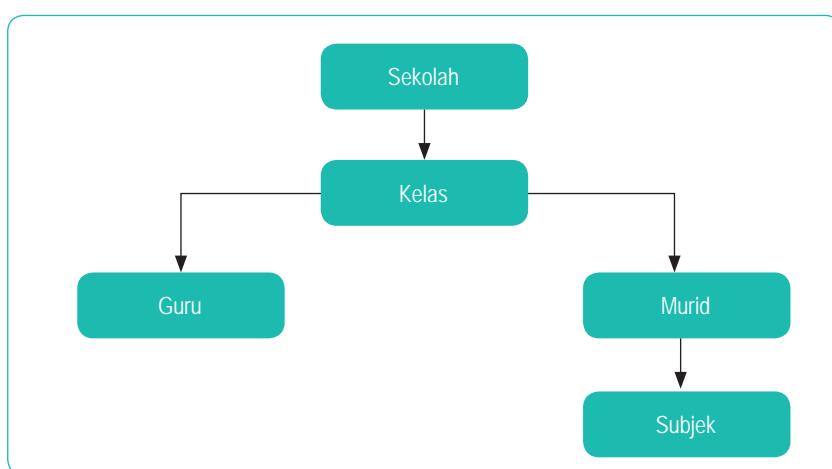
Model hierarki memberikan struktur untuk mengekod rekod data untuk setiap entiti. Perhatikan bahawa entiti sekolah ialah **induk** kepada entiti kelas manakala entiti kelas ialah **anak** kepada entiti sekolah. Sebaliknya, entiti kelas ialah **induk** kepada entiti guru dan entiti murid masing-masing. Entiti murid ialah **induk** kepada entiti subjek. Entiti subjek ialah **anak** kepada entiti murid dan tidak menjadi induk kepada mana-mana entiti.



Sistem Pangkalan data telah mengalami evolusi semenjak dari awal kemunculannya. Anda boleh baca artikel di laman sesawang ini untuk mengetahuinya dengan lebih lanjut.



goo.gl/F0c97d



Rajah 2.9 Model Pangkalan Data Hierarki

Rekod data untuk setiap entiti (Sekolah, Kelas, Murid, Guru dan Subjek) dikodkan dalam organisasi hierarki, lihat Rajah 2.10. Setiap entiti diwakili oleh grid dengan dua bahagian di baris atas dan di baris bawah. Bahagian atas menunjukkan nama atribut manakala bahagian bawah menunjukkan data untuk atribut tersebut.



Rajah 2.10 Entiti sebuah model pangkalan data hierarki



Uji Minda

Berdasarkan Rajah 2.12, fikirkan sejenak. Bukankah Izati boleh mengambil satu atau lebih subjek seperti Matematik Moden, Matematik Tambahan, Biologi, Kimia, Sejarah? Ini bukanlah satu masalah kerana model hubungan tersebut boleh dihasilkan dengan mudah dalam modal hierarki. Cuba anda lukiskan.

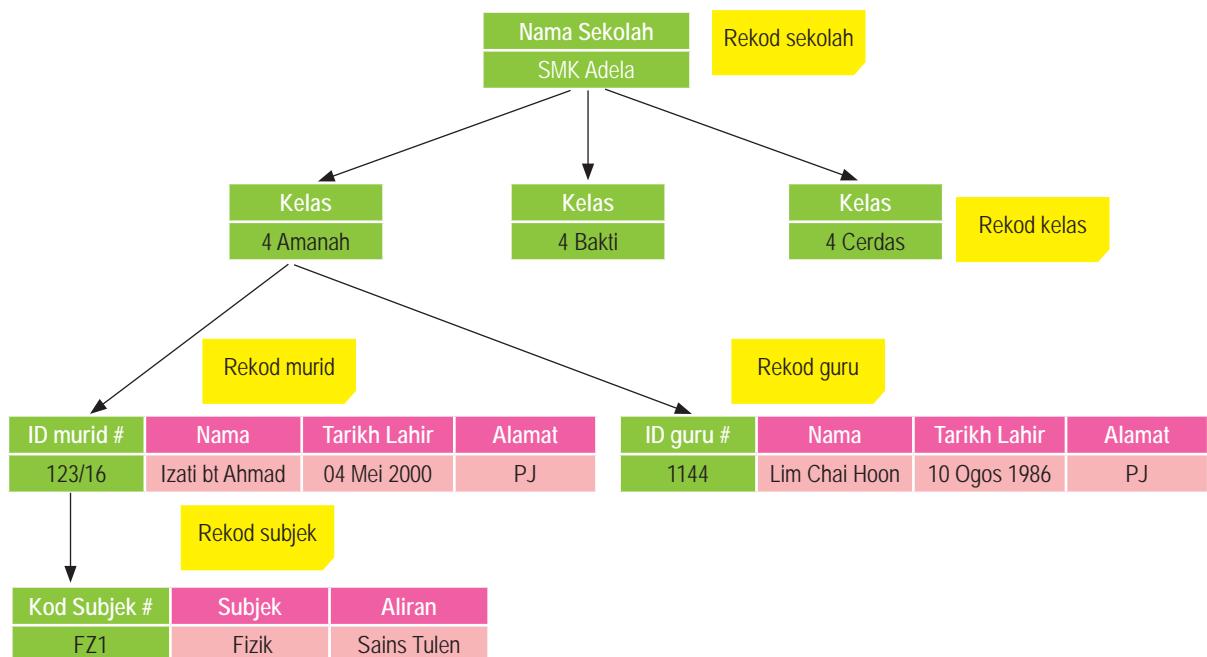
Sebuah entiti boleh mempunyai satu atau lebih atribut. Dalam contoh yang diberikan, entiti sekolah cuma mempunyai satu atribut, nama sekolah. Bagi entiti lain seperti murid, terdapat beberapa atribut lain seperti ID murid, nama, tarikh lahir dan alamat.

ID murid #	Nama	Tarikh Lahir	Alamat
123/16	Izati bt Ahmad	04 Mei 2000	PJ

Rajah 2.11 Entiti dengan beberapa atribut

Dalam entiti dengan beberapa atribut, pilih atribut unik sebagai rujukan (dipanggil kunci primer). Sebagai contoh, ID murid dipilih sebagai kunci primer untuk entiti murid. Gunakan simbol *hash* # sebagai penanda kunci primer.

Gunakan anak panah untuk menghubungkan rekod data mengikut susunan entiti seperti struktur dalam Rajah 2.11. Anak panah dilukis daripada satu rekod data kepada satu rekod data yang lain. Anak panah ditujukan kepada atribut kunci primer dalam setiap rekod data (atribut kunci primer ditandakan #). Lihat Rajah 2.12.



Rajah 2.12 Rekod data dalam model hierarki

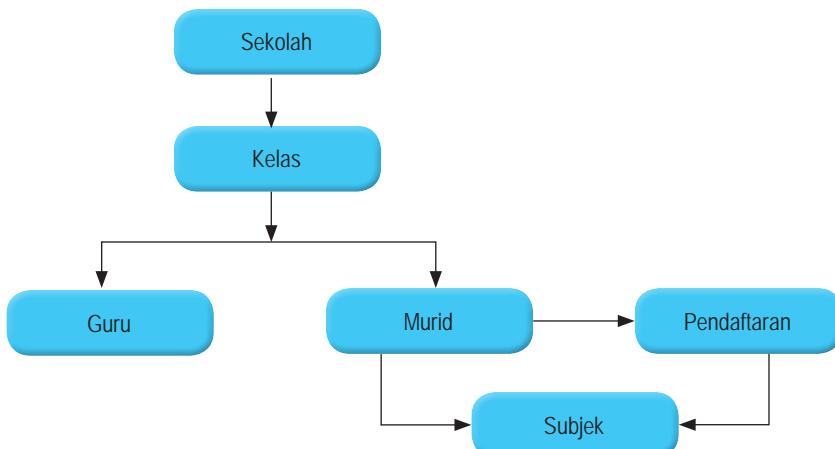
Dalam rajah 2.12, entiti sekolah SMK Adela mempunyai tiga rekod entiti kelas iaitu 4 Amanah, 4 Bakti dan 4 Cerdas. Kelas 4 Amanah mempunyai rekod data murid dengan ID 123/16 dan juga rekod guru dengan ID 1144. Murid 123/16 mengambil entiti subjek dengan kod subjek FZ1.

Model Pangkalan Data Rangkaian

Model pangkalan data rangkaian terdiri daripada beberapa jenis rekod dan dihubungkan melalui penunjuk. Model ini dapat mengatasi beberapa ketidakbolehtentuan dalam model hierarki. Kembali kepada contoh daripada model hierarki. Bolehkah suatu subjek seperti Fizik diambil oleh murid-murid yang lain selain daripada Izati? Sudah tentu ya. Akan tetapi, bolehkah hubungan tersebut ditunjukkan dalam model hierarki? Kelemahan ini diatasi melalui model rangkaian.



Database Models
goo.gl/nquXWs



Rajah 2.13 Model pangkalan data rangkaian

Contoh:

Entiti Subjek boleh mempunyai dua ‘induk’ iaitu entiti Kelas dan Murid. Pencarian semua Subjek dalam Kelas tertentu boleh dibuat secara terus berbanding dengan model hierarki. Hubungan Murid dan Subjek adalah hubungan **banyak-ke-banyak**, iaitu setiap murid mungkin mendaftar untuk banyak Subjek dan setiap Subjek mempunyai banyak Murid.



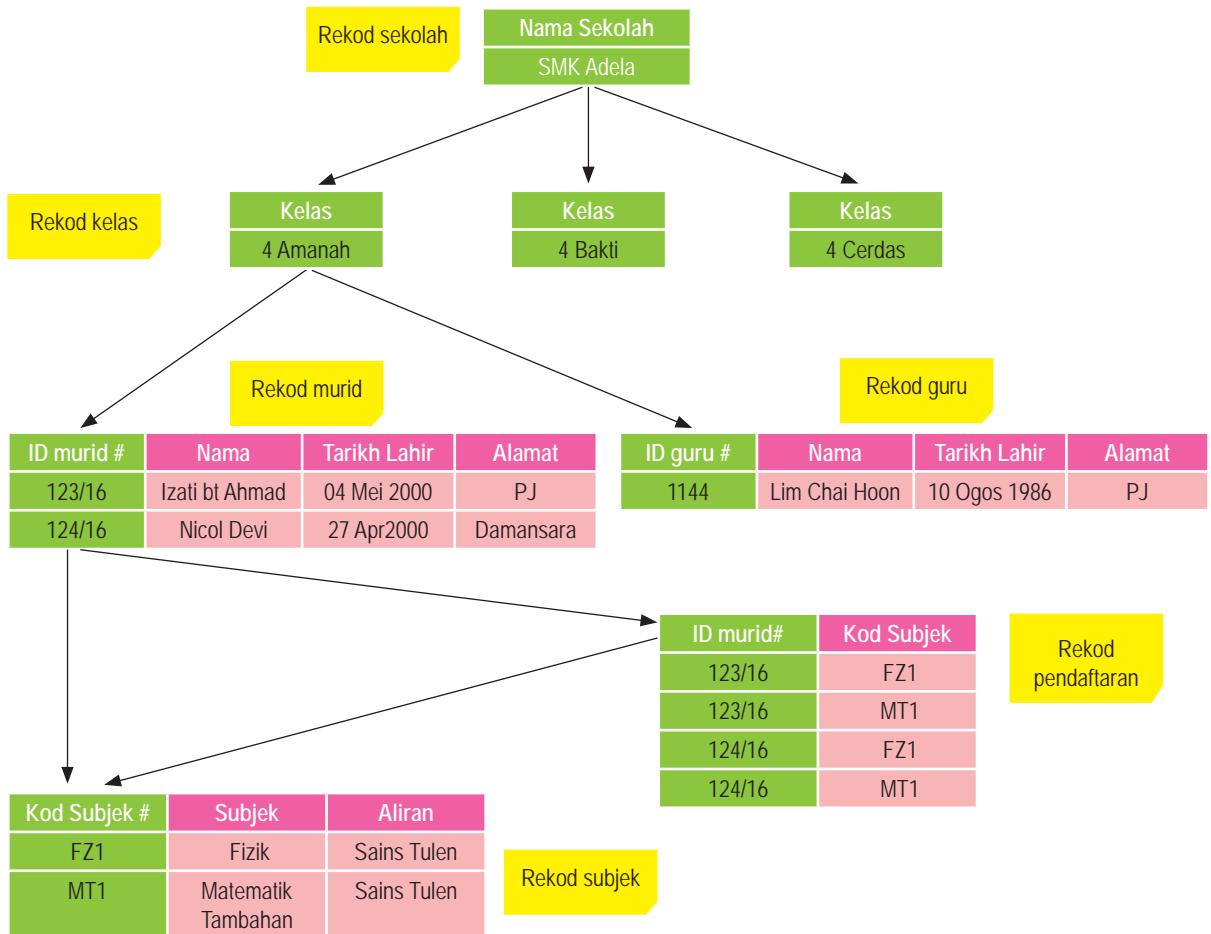
Uji Minda

Anda diberikan tugas untuk membina model pangkalan data rangkaian sama seperti di dalam contoh yang ditunjukkan di sebelah tetapi menukar Entiti Murid dengan mempunyai dua induk iaitu entiti jantina dan subjek.

Tahukah Anda?

Salah seorang tokoh yang pernah memenangi *Turing Award* bagi ciptaan-ciptaan teknologi ialah Edgard Frank Codd. Sewaktu bekerja dengan International Business Machine (IBM) New York, beliau telah mengemukakan idea pangkalan data hubungan pada zaman 1960an. Model tersebut menjadi model yang sangat popular dalam reka bentuk pangkalan data moden. Seorang usahawan membina teknologi pangkalan data berdasarkan idea Codd. Usahawan tersebut, Larry Ellison, telah berjaya mencipta pangkalan data yang popular, *Oracle* dengan kos permulaan USD2000.





Rajah 2.14 Rekod data model pangkalan data rangkaian



Relational database
goo.gl/UVIRnO

Model Hubungan (Relational)

Model ini sering digunakan kerana mudah dibina, digunakan dan diuruskan dalam dunia yang sebenar. Datanya disusun dalam jadual yang terdiri daripada lajur dan baris. Kelebihan model pangkalan data hubungan ialah struktur pangkalan data tidak perlu dinyatakan terlebih dahulu. Perhatikan Rajah 2.15 untuk gambaran yang lebih jelas.

Contoh:

Model ini digambarkan dengan menggunakan tiga entiti (jadual) iaitu MURID, PENDAFTARAN dan SUBJEK. Atribut ID murid dari entiti MURID dan Kod subjek dari entiti SUBJEK dijadikan kunci primer. Kunci primer ini akan dihubungkan kepada kunci asing atribut ID murid dan Kod Subjek di dalam entiti PENDAFTARAN. Selepas dihubungkan melalui kunci primer dan kunci asing ini, data berkenaan akan digabungkan.

Murid			PENDAFTARAN		SUBJEK		
Kelas	Nama	ID murid	ID murid	Kod Subjek	Kod Subjek	Subjek	Aliran
5A	Izati bt Ahmad	123/16	123/16	FZ1	FZ1	Fizik	Sains Tulen
5B	Chong Wei	124/16	128/16	KIM	SC1	Sains	Sastera

Rajah 2.15 Contoh model pangkalan data hubungan

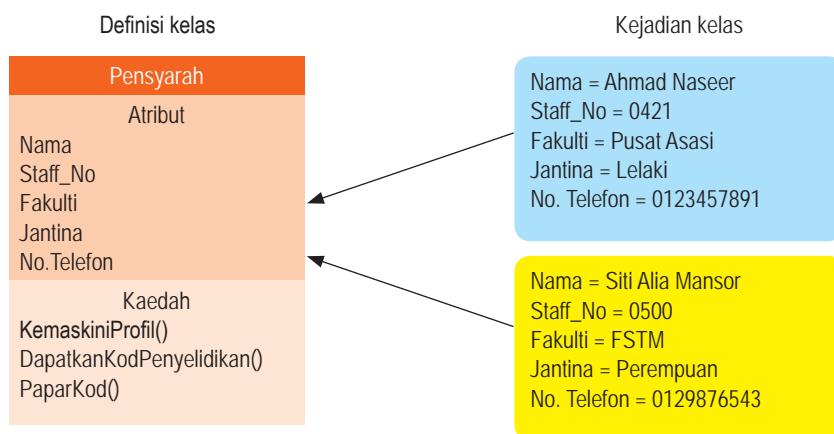
Model Pangkalan Data Berorientasikan Objek (Object-oriented)

Model berorientasikan objek merupakan kaedah yang baharu dalam pengurusan data. Model ini menyimpan takrifan objek-objek yang boleh digunakan semula oleh perisian. Model ini menyimpan objek-objek dan hubungan. Model ini mempunyai ciri yang sama dengan model rangkaian iaitu dapat mewakilkan data untuk hubungan **banyak-ke-banyak**.

Contoh:

Rajah 2.16 menggambarkan konsep orientasi objek bagi kelas Pensyarah yang mempunyai atribut dan kaedah (*method*) tersendiri.

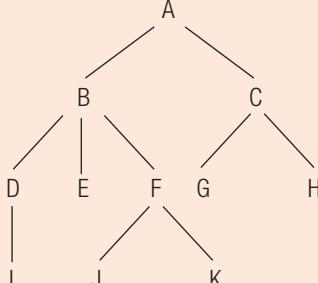
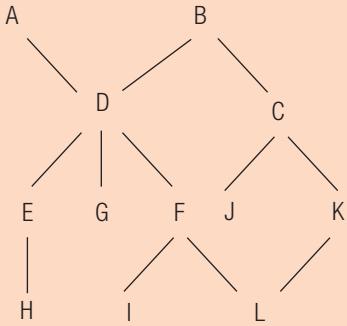
- (a) Kelas (*class*) – menggambarkan keadaan entiti objek sebenar
- (b) Atribut (*attributes*) – data yang mewakili sifat-sifat objek tersebut
- (c) Kaedah (*method*) – menjelaskan kelakuan bagi objek dan juga dikenal sebagai prosedur atau fungsi.



Rajah 2.16 Perwakilan kelas model pangkalan data berorientasikan objek

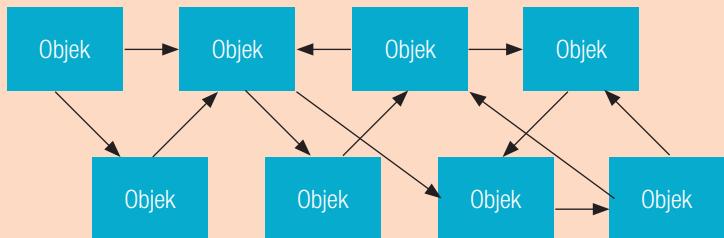
Objek di dalam kelas dikenal sebagai kejadian kelas (*class instances*). Setiap kejadian kelas mempunyai nilai data yang tersendiri bagi setiap atribut tetapi masih boleh berkongsi nama atribut dan kaedah yang sama dengan kejadian kelas yang lain.

Jadual 2.2 Perbandingan antara model-model pangkalan data

Model	Deskripsi																								
Hierarki	<ul style="list-style-type: none"> • Model terawal. • Konsep mudah menggunakan struktur pepohon untuk menyusun rekod. • Untuk hubungan satu induk ke banyak anak. • Tidak sesuai untuk hubungan banyak induk ke banyak anak.  <pre> graph TD A --- B A --- C B --- D B --- E B --- F C --- G C --- H D --- I F --- J F --- K </pre>																								
Pangkalan Data Rangkaian	<ul style="list-style-type: none"> • Untuk mengatasi kelemahan model pangkalan data hierarki. • Untuk hubungan banyak induk ke banyak anak. • Sulit untuk menambahkan hubungan baharu.  <pre> graph TD A --- B A --- C B --- D B --- C D --- E D --- G C --- J C --- K E --- H F --- I F --- L </pre>																								
Pangkalan Data Hubungan <i>(Relational)</i>	<ul style="list-style-type: none"> • Data disusun di dalam jadual terdiri daripada lajur dan baris. • Mudah dibina, digunakan dan diuruskan. • Model ini adalah paling selalu digunakan antara model-model yang lain. <table border="1"> <thead> <tr> <th>ID</th> <th>Alamat</th> </tr> </thead> <tbody> <tr> <td>10359</td> <td>134 Blok Palma-2</td> </tr> <tr> <td>10337</td> <td>10-2 Block C</td> </tr> <tr> <td>10555</td> <td>227 Prima Condo</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ID</th> <th>Nama</th> </tr> </thead> <tbody> <tr> <td>10359</td> <td>Adam</td> </tr> <tr> <td>10337</td> <td>Alan</td> </tr> <tr> <td>10555</td> <td>Atan</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th>ID</th> <th>Gred</th> </tr> </thead> <tbody> <tr> <td>10359</td> <td>A</td> </tr> <tr> <td>10337</td> <td>B+</td> </tr> <tr> <td>10555</td> <td>B</td> </tr> </tbody> </table> <p>Red lines connect the ID column from the first table to the ID column in the middle table, and from the middle table to the ID column in the third table.</p>	ID	Alamat	10359	134 Blok Palma-2	10337	10-2 Block C	10555	227 Prima Condo	ID	Nama	10359	Adam	10337	Alan	10555	Atan	ID	Gred	10359	A	10337	B+	10555	B
ID	Alamat																								
10359	134 Blok Palma-2																								
10337	10-2 Block C																								
10555	227 Prima Condo																								
ID	Nama																								
10359	Adam																								
10337	Alan																								
10555	Atan																								
ID	Gred																								
10359	A																								
10337	B+																								
10555	B																								

Pangkalan Data
Berorientasikan Objek
(*Object-oriented*)

- Kaedah baharu dalam pengurusan data.
- Model ini menyimpan takrifan kelas objek dan hubungan.
- Untuk pangkalan data yang memerlukan hubungan kompleks di antara objek-objek.
- Mempunyai prestasi pencarian yang terbaik antara model-model yang lain.



2.1.3 Entiti, Atribut, Set Hubungan dan Kekardinalan dalam Pangkalan Data

Pangkalan data dibangunkan untuk sistem maklumat yang berada dalam pelbagai persekitaran. Lihat Jadual 2.3. **Persekutaran** ialah organisasi di mana pangkalan data direka bentuk, dibangunkan dan digunakan. Pangkalan data diperlukan untuk menyimpan data-data dari persekitaran untuk memproses sistem maklumat. Untuk mewakili persekitaran tersebut, pangkalan data mengandungi komponen-komponen yang terdiri daripada entiti, atribut, set hubungan dan kekardinalan dalam reka bentuknya.

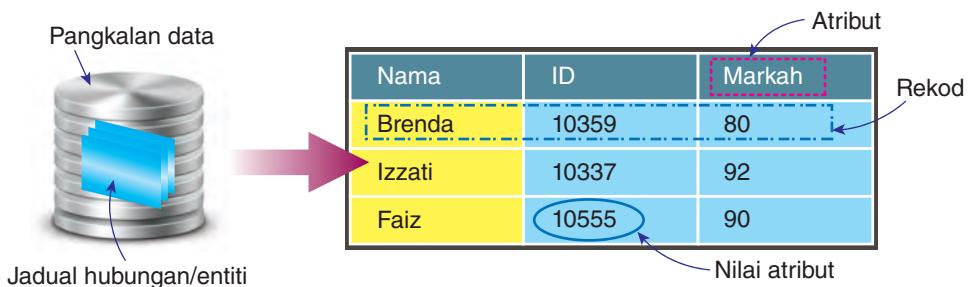


Kelas entiti mestilah memenuhi syarat-syarat yang berikut:

- (a) Kelas tersebut mestilah dilindungi.
- (b) Kelas ini tidak boleh diisyiharkan akhir.
- (c) Entiti boleh melanjutkan kedua-dua entiti.

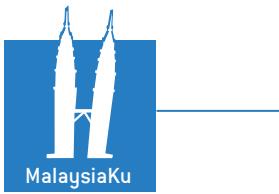
Jadual 2.3 Contoh persekitaran yang biasa menggunakan pangkalan data

Persekutaran	Sistem pangkalan data
Universiti dan kolej	Sistem pangkalan data maklumat pelajar Sistem pangkalan pendaftaran kursus oleh pelajar
Perpustakaan awam	Sistem pangkalan data buku dan koleksi bacaan Sistem pangkalan data pinjaman
Pasar raya	Sistem pangkalan data jualan Sistem pangkalan data produk

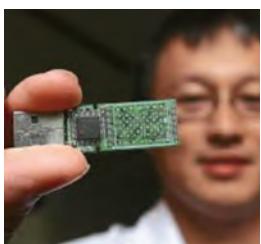


Rajah 2.17 Komponen dalam pangkalan data.

Reka bentuk pangkalan data diperlukan untuk menghasilkan satu atau lebih banyak jadual hubungan. Setiap jadual hubungan mewakili satu jenis entiti. Setiap jadual hubungan mengandungi satu atau lebih rekod. Setiap rekod ialah satu kejadian entiti dan mengandungi sekumpulan nilai atribut. Perhatikan bahawa setiap rekod dalam jadual yang sama mempunyai jenis-jenis atribut yang sama. Kandungan yang seterusnya akan menerangkan dengan lebih lanjut mengenai komponen-komponen pangkalan data ini.



Pada Jun 2001, *PenDrive*, pemacu kilat yang pertama telah diperkenalkan kepada dunia. Pua Khein Seing yang berasal dari Sekinchan, Selangor merupakan salah seorang daripada lima sekawan yang mencipta pemacu kilat pertama di dunia. Kini, beliau merupakan presiden Phison Electronics Corp, syarikat yang telah diasaskan di Taiwan.



Entiti

Entiti merujuk suatu objek unik dan boleh dikenal pasti dalam sesuatu persekitaran seperti orang, tempat atau benda. Entiti mempunyai data-data deskriptif yang boleh dikutip dan disimpan. Bukan semua objek sesuai dijadikan entiti. Jika tiada data dapat dikaitkan atau tiada keperluan untuk menyimpan data berkaitan sesuatu objek, maka objek tersebut tidak sesuai dijadikan sebagai entiti. Dalam pangkalan data, setiap jadual mewakili satu entiti manakala setiap rekod mewakili kejadian satu entiti.

Reka bentuk pangkalan data menggunakan gambar rajah terhubung. Apabila mereka bentuk entiti di atas kertas, gunakan simbol segi empat tepat dan tulis nama entiti. Nama entiti seharusnya kata nama umum yang bersesuaian. Kata nama khusus tidak sepatutnya digunakan.

<Nama entiti>

Rajah 2.18 Simbol entiti

Simbol ini akan dipelajari secara mendalam dalam subtopik 2.2.

CONTOH (3)

Senaraikan entiti untuk pangkalan data dalam persekitaran sekolah.

Penyelesaian:

Entiti-entiti yang boleh digunakan termasuk guru, murid dan mata pelajaran.

CONTOH (4)

Diberikan nama murid seperti Arumugam, Boh Leng, Che Siti dan Iking. Lukiskan simbol entiti yang bersesuaian.

Penyelesaian:

Murid

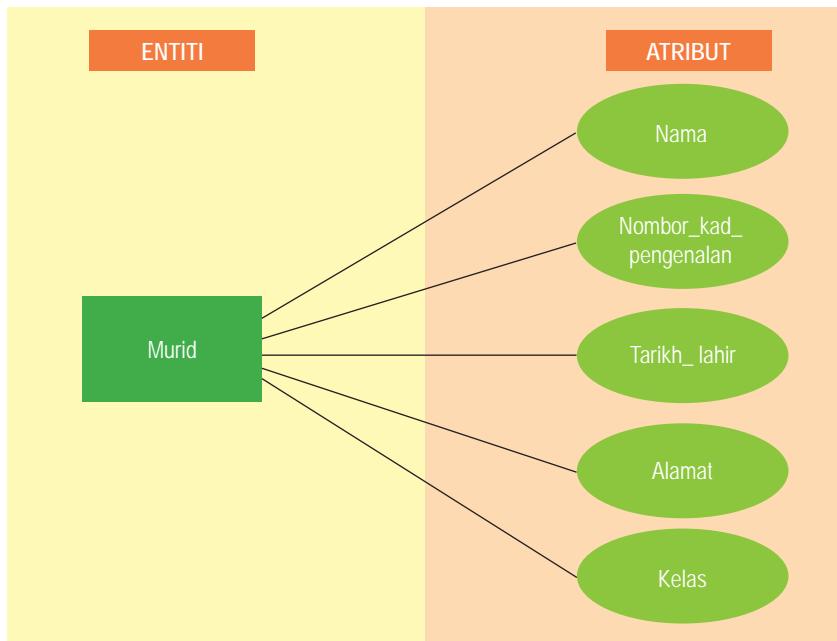
Atribut

Entiti mempunyai data-data deskriptif. Data-data deskriptif ini dipanggil atribut dalam pangkalan data. Atribut penting kerana menerangkan ciri-ciri sesuatu entiti. Dalam pangkalan data, atribut ialah lajur-lajur dalam jadual dengan nama lajur sebagai nama atribut. Bagi setiap rekod, nilai atribut boleh diisi di bawah setiap lajur.

Apabila mereka bentuk atribut di atas kertas, tulis nama atribut dalam simbol bulatan bujur. Kemudian, lukis garis lurus menghubungkan simbol atribut dengan simbol entiti berkaitan.

<nama atribut>

Rajah 2.19 Simbol atribut



Rajah 2.20 Hubungan antara entiti dan atribut

Tahukah Anda ?

Atribut terdiri daripada beberapa jenis. Antaranya ialah,

Atribut mudah

Atribut yang mempunyai nilai yang kecil sahaja dan tidak boleh dibahagikan lagi. Misalnya, nombor kad pengenalan murid mempunyai 12 digit dan tidak boleh diubah lagi.

Atribut komposit

Atribut komposit ialah atribut yang dihasilkan gabungan daripada atribut mudah. Misalnya, nama seorang murid yang terdiri daripada nama_awal dan nama_akhir.

Atribut terbitan

Atribut terbitan ialah atribut yang tidak wujud dalam pangkalan data secara fizikal tetapi nilainya diterbitkan oleh atribut yang lain di dalam pangkalan data. Misalnya, umur yang boleh diterbitkan melalui tarikh lahir.

Atribut nilai tunggal

Atribut yang memiliki nilai yang tunggal. Misalnya, nombor_kad_pengenalan.

Atribut nilai gandaan

Atribut yang memiliki beberapa nilai. Misalnya, seseorang boleh memiliki lebih daripada satu nombor telefon.



Set Hubungan

Hubungan ialah perkaitan antara entiti. Hubungan merujuk perbuatan dan diwakili oleh kata kerja. Bandingkan pengertian hubungan dengan entiti. Entiti merujuk objek dan diwakili oleh kata nama umum.

Hubungan pula melibatkan dua entiti. Entiti pertama ialah entiti subjek, iaitu pelaku yang melaksanakan hubungan tersebut ke atas entiti kedua. Entiti kedua menerima kesan daripada hubungan itu, dan oleh itu dipanggil entiti objek.



Uji Minda

Berdasarkan Contoh 5, jawab soalan yang berikut. Diberikan hubungan-hubungan berikut.

- Samy suka bermain bola sepak.
 - Hassan suka bermain sepak takraw.
 - Jeremy suka bermain hoki.
- (a) Kenal pasti entiti dalam hubungan-hubungan di atas.
(b) Nyatakan nama hubungan yang sesuai dan lukis gambar rajah hubungan yang bersesuaian.

Set hubungan ialah koleksi sepasang entiti subjek-objek yang mempunyai hubungan yang sama. Set hubungan menggunakan sintaks yang berikut.

<entiti subjek> <hubungan> <entiti objek>

Sintaks gambar rajah terhubung adalah seperti berikut:



Rajah 2.21 Set hubungan yang menunjukkan hubungan di antara dua entiti

Untuk memahami konsep ini dengan lebih mendalam, kaji contoh-contoh yang berikut.

Contoh 5

Diberikan hubungan-hubungan berikut.

- Adam belajar Sains Komputer.
- Ai Ling belajar Geografi.
- Anastasia belajar Sejarah.

- (a) Kenal pasti entiti dalam hubungan-hubungan di atas.
(b) Nyatakan nama hubungan yang sesuai.
(c) Lukis gambar rajah hubungan yang bersesuaian.

Penyelesaian:

- (a) Adam, Ai Ling dan Anastasia ialah kejadian ataupun objek bagi entiti Murid. Sains Komputer, Geografi dan Sejarah ialah kejadian ataupun objek bagi entiti Mata pelajaran.
(b) Belajar
(c)



CONTOH (6)

Diberikan hubungan-hubungan berikut.

- Adam deposit A0109.
- Ali deposit A0150.
- Barbara deposit A1011.

(a) Kenal pasti entiti dalam hubungan-hubungan di atas.

(b) Lukis gambar rajah hubungan yang bersesuaian.

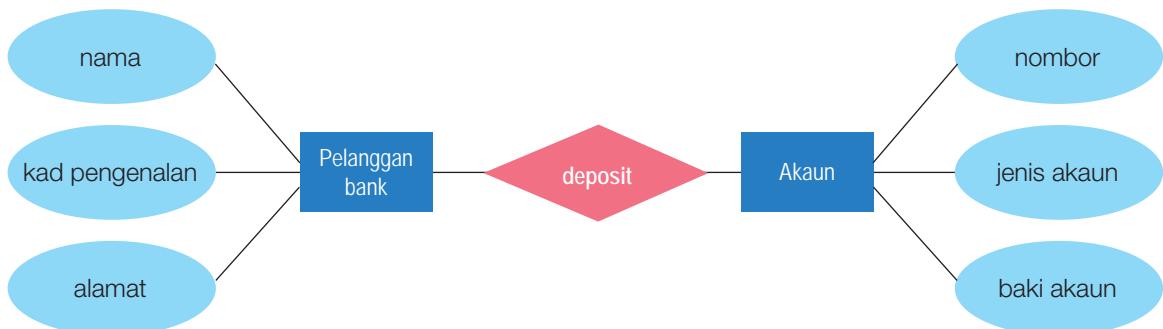
Penyelesaian:

- (a) Dalam contoh di atas, entiti Adam, Ali dan Barbara masing-masing ialah entiti subjek jenis Pelanggan Bank. Entiti A0109, A0150 dan A1011 pula masing-masing ialah entiti objek jenis Akaun. Semua pasangan entiti, iaitu Adam-A0109, Ali-A0150 dan Barbara-A1011 mempunyai hubungan yang sama iaitu deposit.

(b)



Contoh di atas menggunakan gambar rajah terhubung yang ringkas. Gambar rajah yang lengkap seharusnya menunjukkan atribut-atribut bagi entiti. Lihat Rajah 2.22.



Rajah 2.22 Gambar rajah lengkap yang menunjukkan atribut-atribut bagi entiti

Kekardinalan (Cardinality)

Kekardinalan merujuk perhubungan antara entiti. Kekardinalan menyatakan bilangan entiti yang boleh dihubungkan dengan entiti yang lain melalui set hubungan.

Dalam reka bentuk set hubungan, beberapa jenis kekardinalan akan digunakan. Antaranya ialah kekardinalan satu-ke-satu (1:1), satu-ke-banyak (1:M) atau banyak-ke-banyak (M:N). Kekardinalan ialah berasaskan keperluan pengguna. Dalam gambar rajah terhubung, kekardinalan ditulis di atas kedua-dua garis yang menghubungkan entiti kepada hubungan dan hubungan kepada entiti. Lihat Jadual 2.4.

Jadual 2.4 Kekardinalan

Kekardinalan	Gambar rajah terhubung	Penerangan
1:1 (satu-ke-satu)	<pre> graph LR Warganegara[Warganegara] -- 1 --> punya{punya} punya -- 1 --> KadKad[Kad pengenalan] </pre>	<p>Hubungan satu entiti dengan satu entiti yang lain.</p> <p>Contoh : Seorang warganegara mempunyai satu kad pengenalan sahaja.</p>
1:M (satu-ke- banyak)	<pre> graph LR Murid[Murid] -- 1 --> sertai{sertai} sertai -- M --> Kelab[Kelab] </pre>	<p>Hubungan satu entiti dengan lebih daripada satu entiti yang lain.</p> <p>Contoh : Seorang murid boleh menyertai lebih daripada satu kelab di sekolah.</p>
M:N (banyak-ke- banyak)	<pre> graph LR Pelanggan[Pelanggan] -- M --> membeli{membeli} membeli -- N --> barang[barang] </pre>	<p>Hubungan antara banyak entiti dengan banyak entiti yang lain.</p> <p>Contoh : Ramai pelanggan bagi sebuah pasar raya membeli pelbagai jenis barang.</p>

Dalam pangkalan data, kekardinalan merujuk hubungan di antara rekod-rekod dalam satu jadual (katakan jadual A) kepada rekod-rekod dalam jadual lain (katakan jadual B). Untuk kekardinalan 1:1, satu baris rekod dalam jadual A cuma boleh dihubungkan kepada satu baris rekod dalam jadual B. Sekiranya kekardinalan adalah 1:M, satu baris rekod dalam jadual A boleh dihubungkan kepada satu atau lebih baris rekod dalam jadual B; tetapi setiap rekod dalam jadual B cuma boleh mempunyai hubungan dengan satu baris rekod dalam jadual A. Bagaimanakah pula hubungan rekod-rekod antara jadual A dengan jadual B sekiranya kekardinalan adalah M:N?

Jadual A

Atribut 1	Atribut 2
Nilai atribut 1 untuk Rekod1	Nilai atribut 2 untuk Rekod1
Nilai atribut 1 untuk Rekod2	Nilai atribut 2 untuk Rekod2

Jadual B

Atribut 1	Atribut 2
Nilai atribut 1 untuk Rekod1	Nilai atribut 2 untuk Rekod1
Nilai atribut 1 untuk Rekod2	Nilai atribut 2 untuk Rekod2

Rajah 2.23 Bilangan hubungan antara rekod-rekod dalam jadual A dan jadual B mengikut kekardinalan M:N

Bincang dalam kumpulan dan persembahkan kepada rakan sekelas mengenai persoalan di bawah. Terangkan kepada rakan sekelas mengapa anda memilih jawapan tersebut.

- 1** Tentukan entiti yang sesuai bagi setiap kumpulan data yang berikut:
 - (a) Sejarah, Sains, Bahasa Inggeris, Matematik
 - (b) Perlis, Kedah, Pulau Pinang, Perak, Selangor, Negeri Sembilan, Melaka, Johor
 - (c) Malaysia Airlines, AirAsia, Garuda, Singapore Airlines, Thai Airways, VietJet Air
- 2** Selain daripada contoh yang telah diberikan, berikan contoh-contoh harian untuk kekardinalan berikut. Bincang dalam kumpulan anda.
 - (a) 1:1
 - (b) 1:M
 - (c) M:N



Kerja Sains Komputer

Teknologi pangkalan data telah mewujudkan empat jenis profesion seperti yang berikut:

- Pentadbir data (DA)
- Pentadbir pangkalan data (DBA)
- Perekabentuk pangkalan data
- Pembangun aplikasi

Latihan Formatif 2.1

- 1** Nyatakan sama ada pernyataan yang berikut benar atau palsu.
Ketekalan data merujuk proses mengekalkan dan memastikan ketepatan data yang sah melalui keseluruhan kitaran hayat data.
- 2** Data yang dimasukkan ke dalam komputer dan disimpan dalam memori boleh hilang apabila bekalan elektrik diputuskan. Apakah yang perlu anda lakukan untuk memastikan data itu kekal tersimpan dalam memori komputer?
- 3** Apakah yang anda faham dengan ketekalan data? Justifikasikan jawapan anda berserta contoh.
- 4** Berdasarkan rajah di bawah, tentukan entiti dan atribut dengan betul.

PENSYARAH

ID pensyarah	Nama	Fakulti	Kursus
AMIN011	Aminah bt Darus	Bahasa	Bahasa Melayu
CHONG012	Chong Wei Hou	Fakulti Sains	JAVA

- 5** Kenal pasti kekardinalan bagi set hubungan yang ditunjukkan di bawah. Berikan justifikasi bagi jawapan anda.



- 6** Apakah perbezaan antara model-model pangkalan data di bawah? Isi ruang kosong dalam jadual yang disediakan. Bincang dengan rakan anda. Anda juga boleh menggunakan enjin carian untuk mendapatkan maklumat dan melengkapkan jawapan anda.

Hierarki	Rangkaian	Hubungan	Berorientasikan objek

- 7** Aaron anak Jili, Lee dan Muthusamy sedang menyiapkan tugas mengenai pangkalan data Maklumat Murid Tingkatan 5 Cerdas yang diberikan oleh guru. Seramai 25 orang murid kelas 5 Cerdas yang terlibat. Semasa menyemak data yang diperoleh daripada murid 5 Cerdas, Lee mendapat terdapat lebihan sebanyak dua data yang tidak diketahui puncanya menjadikan jumlah data tersebut kepada 27. Apakah yang berlaku? Berikan justifikasi bagi jawapan anda.

IC Murid	Nama Murid	Tarikh Lahir	Kelas
990405-11-5434	Aliza bt Wan Ahmad	05 Mei 1999	5 Cerdas
990213-03-5675	Lim Kit Wu	13 Februari 1999	5 Cerdas
990607-14-4343	Suraj A/L Ramu	07 Jun 1999	5 Cerdas
990213-03-5675	Lim Kit Wu	13 Februari 1999	5 Cerdas
990405-11-5434	Aliza bt Wan Ahmad	05 Mei 1999	5 Cerdas

- 8** Dengan menggunakan enjin carian, cari dan kaji apakah kelebihan dan kelemahan antara model pangkalan data rangkaian dan model pangkalan data hubungan. Berikan justifikasi bagi jawapan anda.

	Model rangkaian	Model hubungan
Kelebihan		
Kekurangan		

- 9** Tentukan atribut-atribut yang bersesuaian dengan entiti-entiti berikut dengan melukis gambar rajah terhubung untuk setiap entiti.

- (a) Guru
(b) Mata pelajaran

- 10** Diberikan hubungan-hubungan yang berikut. Lukis gambar rajah terhubung ringkas yang bersesuaian setelah mengenal pasti nama entiti-entiti dan nama hubungan yang bersesuaian :

- (a)
- Tze-Ming mengambil peperiksaan Matematik Tambahan.
 - Rahmat mengambil peperiksaan Kimia.
 - Ganesh mengambil peperiksaan Perakaunan.
 - Terry Embong mengambil peperiksaan Perakaunan.

- (b) • Cikgu Saad mengajar Matematik Moden.
• Cikgu Dinesh mengajar Matematik Tambahan.
• Cikgu Chong mengajar Bahasa Melayu.
• Cikgu Sarah Nyelong mengajar Sejarah.
- (c) • Pengeluar kereta Proton mengeluarkan model kereta Iriz.
• Pengeluar kereta Proton mengeluarkan model kereta Perdana.
• Pengeluar kereta Perodua mengeluarkan model kereta Axia.
• Pengeluar kereta Perodua mengeluarkan model kereta Bezza.
• Pengeluar kereta Honda mengeluarkan model kereta Civic.
• Pengeluar kereta Toyota mengeluarkan model kereta Camry.

11 Lukis gambar rajah terhubung lengkap yang bersesuaian dengan maklumat-maklumat berikut.

- Seorang murid mengambil satu atau lebih mata pelajaran di sekolah.
- Sesuatu mata pelajaran boleh diambil oleh satu atau lebih murid.
- Seorang murid mempunyai data seperti nama, kad pengenalan dan tingkatan.
- Sesebuah subjek mempunyai data seperti nama dan nama guru yang mengajar.



STANDARD PEMBELAJARAN

- 2.2.1 Menentukan kunci primer dan/atau kunci asing yang sesuai bagi setiap entiti
- 2.2.2 Menghasilkan gambar rajah perhubungan entiti (ERD) untuk memodelkan data hubungan
- 2.2.3 Mereka bentuk pangkalan data hubungan dengan menukar ERD ke skema hubungan
- 2.2.4 Menjelaskan konsep kebergantungan fungsi sepenuh, kebergantungan fungsi separa dan kebergantungan fungsi transitif
- 2.2.5 Melaksanakan proses penormalan data sehingga 3NF terhadap skema hubungan

Reka Bentuk Pangkalan Data Hubungan

Pangkalan data dengan model hubungan boleh digunakan untuk mengatasi masalah kelebihan dan ketidaktekalan data secara berkesan.

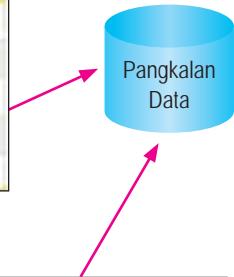
Lihat jadual dalam Rajah 2.24. Jadual mengandungi senarai peminjam buku dan tarikh pulang setiap buku yang dipinjam. Jadual ini lebar kerana mengandungi banyak lajur atribut. Jadual yang lebar cenderung mengandungi banyak baris rekod yang mengulang nilai-nilai atribut yang sama.

Jadual asal

Kod Buku	ID Murid	Nama	No Telefon	Tarikh Pinjam	Tarikh Hantar
IPB124044	125007	ALI BIN AHMAD	053689090	22-Aug-16	12-Sep-16
IPB257868	125007	ALI BIN AHMAD	053689090	22-Aug-16	12-Sep-16
IPB192254	125007	ALI BIN AHMAD	053689090	29-Aug-16	12-Sep-16
IPB051375	125096	FRANCIS JR	036039999	23-Aug-16	13-Sep-16
IPB061045	125888	LIM S W	075555768	23-Aug-16	13-Sep-16

Jadual PEMINJAM

ID Murid	Nama	No Telefon
125007	ALI BIN AHMAD	053689090
125007	ALI BIN AHMAD	053689090
125007	ALI BIN AHMAD	053689090
125096	FRANCIS JR	036039999
125888	LIM S W	075555768



Jadual PINJAMAN

Kod Buku	ID Murid	Tarikh Hantar	Tarikh Pulang
IPB124044	125007	22-Aug-16	12-Sep-16
IPB257868	125007	22-Aug-16	12-Sep-16
IPB192254	125007	29-Aug-16	12-Sep-16
IPB051375	125096	23-Aug-16	13-Sep-16
IPB061045	125888	23-Aug-16	13-Sep-16

Rajah 2.24 Penormalan jadual untuk menghasilkan jadual-jadual hubungan

Oleh itu, jadual yang lebar boleh dipecahkan kepada beberapa jadual hubungan yang lebih kecil. Jadual-jadual inilah yang akan disimpan dalam pangkalan data. Kaedah ini dipanggil **penormalan** dan digunakan untuk mencegah kelebihan dan kehilangan ketekalan data.



Persoalannya, bagaimanakah sesuatu jadual yang lebar dapat dipecahkan kepada jadual-jadual hubungan secara berkesan? Adakah carian dapat dilaksanakan merentasi hubungan-hubungan ini, seolah-olah telah disambung semula untuk dijadikan jadual asal apabila perlu?

Dalam model pangkalan data hubungan, lajur khusus dalam hubungan-hubungan digunakan sebagai penghubung logikal. Penghubung ini membolehkan atribut-atribut dihubungkan semula sewaktu carian atau kemas kini. Inilah yang dikatakan sebagai integriti data hubungan. Tanpa penghubung, atribut dalam hubungan-hubungan sukar dikesan dan ini boleh menjasaskan ketekalan dan kebolehcapaian data. Oleh hal yang demikian, pangkalan data dikatakan gagal mengekalkan integriti data hubungan.

Tahukah Anda ?

Kebolehcapaian merujuk perkongsian data untuk pengguna yang berbeza

2.2.1 Kunci Primer dan Kunci Asing

Ingatkah lagi dengan set hubungan? Dalam set hubungan, suatu entiti boleh dikaitkan dengan suatu entiti yang lain. Entiti-entiti diwakilkan sebagai jadual-jadual dalam pangkalan data. Kunci hubungan menghubungkan jadual-jadual yang mewakili entiti-entiti dalam set hubungan.

Kunci hubungan ialah nama atribut khusus dalam jadual yang digunakan untuk mengekalkan integriti data hubungan. Terdapat dua kunci hubungan iaitu kunci primer dan kunci asing.

Kunci primer ialah atribut (atau gabungan beberapa atribut) dengan nilai yang unik untuk mewakili setiap baris rekod dalam sesebuah jadual. Kunci primer tidak boleh dibiarkan kosong (*null*) dan tidak boleh mengandungi nilai yang berulang. Secara analogi, kunci primer adalah seperti nombor kad pengenalan yang unik dan sesuai untuk mengenal pasti identiti setiap murid yang merupakan warganegara Malaysia. Oleh hal yang demikian, penggunaan nama untuk suatu entiti kurang sesuai kerana nama kadang-kadang boleh berulang untuk kejadian entiti yang lain. Dalam entiti Peminjam (Lihat jadual PEMINJAM dalam Rajah 2.24), atribut ID Murid sesuai dijadikan kunci primer kerana tidak mempunyai nilai atribut berulang. Jadual 2.4 menunjukkan ciri-ciri dan kepentingan kunci primer.

Jadual 2.4 Ciri-ciri dan kepentingan kunci primer

Kunci primer	
Ciri-ciri	<ul style="list-style-type: none">Mempunyai nilai yang unik iaitu hanya satu-satunya sahaja.Tidak boleh dibiarkan kosong (<i>null</i>) iaitu tanpa mengisi data dalam barisnya.Tidak boleh mempunyai nilai yang sama secara berulang.Nilai yang tetap dan tidak berubah.



Uji Minda

Bayangkan anda diberikan satu jadual yang mempunyai rekod setiap murid tingkatan 4 di sekolah anda. Di dalam jadual tersebut mengandungi nama murid, tarikh lahir dan alamat rumah. Pada pendapat anda, yang manakah paling sesuai dijadikan sebagai kunci primer?

<p>Kepentingan</p>	<ul style="list-style-type: none"> Memastikan setiap rekod lama yang tersimpan tidak bertindih dengan rekod baharu. Memberikan identiti yang unik bagi setiap rekod yang disimpan di dalam pangkalan data. Menjadikan data yang disimpan adalah utuh kerana setiap rekodnya mempunyai satu nilai yang unik sebagai pengenalan diri. Mengelakkan berlakunya pertindihan data daripada rekod yang sama berulang-ulang disimpan di dalam pangkalan data. Seterusnya, dapat menjimatkan ruang stor komputer. Memudahkan proses carian dan capaian atas rekod. Rekod yang hendak dicari boleh dicapai dengan membuat carian menggunakan kunci primer.
--------------------	---

Kunci asing ialah atribut yang mengandungi nilai kunci primer daripada jadual sasaran. Dengan kata lain, untuk menghasilkan hubungan di antara dua jadual, jadual yang ingin mewujudkan hubungan akan menambahkan atribut kunci primer daripada jadual sasaran sebagai kunci asing. Lihat Rajah 2.25 di bawah.



Uji Minda

Setelah mempelajari mengenai kunci primer dan kunci asing, nyatakan perbezaan antara kunci primer dan kunci asing.

Jadual PINJAMAN

Kod Buku	ID Murid	Tarikh Hantar	Tarikh Pulang
IPB124044	125007	22-Aug-16	12-Sep-16
IPB257868	125007	22-Aug-16	12-Sep-16
IPB192254	125007	29-Aug-16	12-Sep-16
IPB051375	125096	23-Aug-16	13-Sep-16
IPB061045	125888	23-Aug-16	13-Sep-16

Kunci primer Kunci asing

Jadual PEMINJAM

ID Murid	Nama	No Telefon
125007	ALI BIN AHMAD	053689090
125096	FRANCIS JR	036039999
125888	LIM S W	075555768

Kunci primer



Primary key vs foreign key
goo.gl/0oah1m

Rajah 2.25 Hubungan antara kunci primer dan kunci asing yang menggunakan contoh jadual hubungan PINJAMAN dan PEMINJAM

Jadual 2.5 menerangkan penggunaan kunci hubungan dalam jadual hubungan PINJAMAN dan jadual hubungan PEMINJAM.

Jadual 2.5 Ciri-ciri dan kepentingan kunci primer

Jadual PINJAMAN	Jadual PEMINJAM
(a) Dalam jadual hubungan Pinjaman, terdapat empat atribut iaitu Kod Buku, ID Murid, TarikhPinjam dan TarikhPulang.	(a) Dalam Jadual PEMINJAM, kunci primer ialah ID_Murid. Perhatikan bahawa atribut yang sama muncul dalam Jadual PINJAMAN sebagai kunci asing.
(b) Lajur untuk atribut Kod Buku tidak boleh dibenarkan kosong, dan setiap satu nilai atribut adalah unik (tiada ulangan). Keunikan nilai menjadikan atribut kod buku sebagai kunci primer.	(b) Adakah Jadual PEMINJAM mempunyai kunci asing? Jadual PEMINJAM tidak mempunyai kunci asing kerana tiada sebarang hubungan kepada kunci primer.
(c) Selain atribut Kod Buku, satu lagi atribut yang menarik ialah ID Murid. Perhatikan bahawa atribut ID Murid ialah kunci primer dalam Jadual PEMINJAM. Oleh yang demikian, ID_Murid dalam Jadual PINJAMAN menjadi penghubung kepada Jadual PEMINJAM. ID_Murid ialah kunci asing dalam Jadual PINJAMAN.	



Terdapat beberapa perbezaan bagi kunci primer dan kunci asing. Antaranya ialah,

- (a) Kunci primer tidak menerima nilai kosong (*null*) manakala kunci asing boleh menerima beberapa nilai kosong (*null*).
- (b) Hanya satu kunci primer yang boleh didapati dalam satu jadual berbeza dengan kunci asing yang didapati banyak dalam satu jadual.



35

Kerja Berkumpulan
Penyimpanan Rekod Pinjaman Perpustakaan Sekolah

Adakan satu lawatan ke perpustakaan sekolah. Salah satu kegunaan perpustakaan adalah sebagai tempat meminjam buku. Temu ramah pustakawan untuk mengenal pasti perkara-perkara berikut:

- 1 Apakah data-data yang perlu direkodkan bagi setiap pinjaman?
 - (a) Mengapakah data yang direkodkan itu penting?
 - (b) Bagaimanakah data-data tersebut disimpan?
- 2 Bagaimanakah data-data tersebut dapat digunakan?
 - (a) Bagaimanakah perpustakaan mencari data-data daripada jumlah rekod yang sebeginu banyak?
 - (b) Pernahkah rekod-rekod lama perlu dibetulkan semula kerana kesilapan atau pertukaran data?
- 3 Lukis jadual untuk menyimpan rekod-rekod pinjaman.
 - (a) Masukkan sampel-sampel data untuk mewujudkan rekod-rekod yang baharu. Dicadangkan 20 baris rekod bagi tiga orang peminjam. Gunakan pensel.
 - (b) Gunakan pen *highlighter* untuk mewarnai kumpulan data yang berulang. Jika kumpulan adalah sama, gunakan warna yang sama.
 - (c) Ubah salah satu data, misalnya nombor kad pengenalan salah seorang murid. Adakah perubahan ini juga melibatkan rekod-rekod yang lain?



Uji Minda

Selain daripada menggunakan contoh kod buku dan ID murid, cuba tunjukkan hubungan antara kunci primer dan kunci asing melalui markah murid bagi subjek Sains Komputer Tingkatan 4 dan ID murid. Selamat Mencuba!





Entity Relationship
Diagram (ERD)
goo.gl/mXfT3q

2.2.2 Gambar Rajah Perhubungan Entiti (ERD)

Gambar Rajah Perhubungan Entiti (*Entity Relationship Diagram - ERD*) ialah teknik grafik untuk memodelkan data. Model yang dihasilkan adalah pada tahap konsep dan tidak terikat kepada mana-mana Sistem Pengurusan Pangkalan Data (*Database Management System - DBMS*). ERD mewakili persekitaran dalam struktur entiti, hubungan entiti dan atribut entiti.

Komponen-komponen asas ERD ialah:

Entiti

Entiti ialah sesuatu yang mempunyai data untuk disimpan. Entiti biasanya dilabelkan dengan kata nama. Simbol entiti ialah segi empat dengan name entiti di tengah-tengahnya.

Nama Entiti

Rajah 2.26 Simbol entiti dalam ERD



ER Diagram Representation
goo.gl/4PB9wD

Entiti boleh terdiri daripada elemen-elemen persekitaran (orang, objek, tempat, konsep dan kejadian). Entiti orang, objek dan tempat berbentuk fizikal. Konsep dan kejadian pula berbentuk abstrak. Contoh setiap satu diberikan dalam Jadual 2.6.

Jadual 2.6 Contoh entiti lazim

Bil	Jenis Entiti	Contoh
1	Orang	Guru, murid, doktor
2	Tempat	Negara, negeri, daerah, bandar, desa
3	Objek	Produk, kenderaan, peralatan, bangunan
4	Peristiwa	Pendaftaran, permohonan, rayuan, pertanyaan, transaksi
5	Konsep	Akaun, kursus

Hubungan

Hubungan ialah perkaitan yang wujud antara dua entiti. Hubungan biasanya dilabelkan menggunakan kata kerja. Simbol hubungan ialah rombus.



Rajah 2.27 Simbol hubungan

Contoh: mendaftar, mempunyai, mengisi, meminjam

Rajah 2.28 menunjukkan hubungan "mendaftar" antara entiti subjek Murid dan entiti objek Kursus. Rajah berikut boleh dibaca sebagai "Murid Mendaftar Kursus."

Contoh penggunaan:



Rajah 2.28 Hubungan antara entiti

Atribut

Atribut ialah ciri atau sifat entiti. Setiap set entiti terdiri daripada beberapa atribut. Atribut diwakili oleh simbol elips.

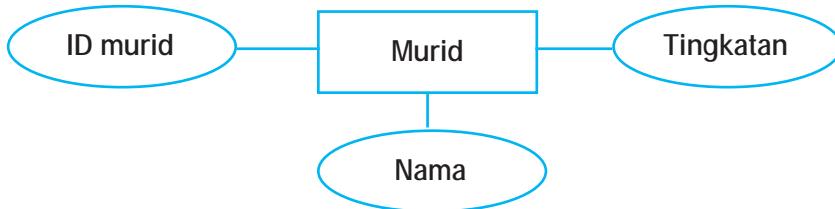


Rajah 2.29 Simbol atribut

Contoh atribut untuk entiti murid: ID murid, nama, tingkatan, jantina

Contoh penggunaan:

Rajah 2.30 menunjukkan entiti Murid mempunyai atribut seperti ID Murid, Nama dan Tingkatan.



Rajah 2.30 Atribut-atribut entiti murid

Langkah-langkah melukis ERD adalah seperti berikut:

- 1 Kenal pasti data yang diperlukan oleh sistem dari persekitaran pangkalan data.
- 2 Kenal pasti kumpulan untuk data yang berkait secara logikal. Kumpulan ini dinamakan sebagai entiti.
- 3 Kenal pasti perkaitan antara dua entiti untuk semua entiti.

Contoh 1

Berikut ialah langkah-langkah untuk melukis ERD untuk kenyataan "Murid mengambil Peperiksaan".

- 1 Kenal pasti entiti: Murid, Peperiksaan
- 2 Kenal pasti hubungan: mengambil
- 3 Lukiskan simbol entiti dan hubungan yang mengaitkan kedua-dua entiti.



Terdapat beberapa jenis ERD, iaitu:

- (a) ERD konseptual
- (b) ERD logikal
- (c) ERD fizikal

Sumber:

www.lucidchart.com



Jasmin Kaur minat membaca. Dia sering pergi ke pameran buku untuk membeli pelbagai jenis buku yang terdiri daripada pelbagai genre seperti buku fiksyen, bukan fiksyen, ensiklopedia, novel dan bibliografi. Bina satu gambar rajah perhubungan entiti berdasarkan situasi Jasmin Kaur.





Ingatkah lagi konsep kekardinalan dalam pangkalan data hubungan? Kekardinalan menunjukkan kuantiti entiti-entiti dalam perhubungan. Lihat Jadual 2.7.

Jadual 2.7 Jenis-jenis kekardinalan

Kekardinalan	Penerangan
1:1 (satu-ke-satu)	Hubungan: "Seorang warganegara mempunyai satu kad pengenalan sahaja."
1:M (satu-ke-banyak)	Hubungan: "Seorang murid boleh menyertai lebih daripada satu kelab di sekolah."
M:N (banyak-ke-banyak)	Hubungan: "Banyak pelanggan pasar raya membeli pelbagai jenis barang."



36

Kerja Berkumpulan

- A** Dalam kelas anda, wujudkan satu senarai butiran murid. Butiran ini diperlukan untuk mencatatkan sumbangan setiap murid kepada tabung kelas setiap bulan (12 bulan). Tentukan entiti dan maklumat-maklumat yang diperlukan bagi setiap murid. Lukis Gambar Rajah Perhubungan Entiti untuk tujuan tersebut. Berikan justifikasi kenapa maklumat-maklumat tersebut diperlukan.
- B** Setiap murid mungkin menyertai satu atau lebih kelab di sekolah. Seorang murid dibenarkan menyertai maksimum tiga kelab sahaja. Tentukan entiti-entiti dan maklumat-maklumat yang diperlukan untuk kelab. Lukis Gambar Rajah Perhubungan Entiti untuk tujuan tersebut. Berikan justifikasi mengapakah maklumat-maklumat tersebut diperlukan.



ERD
goo.gl/4PB9wD



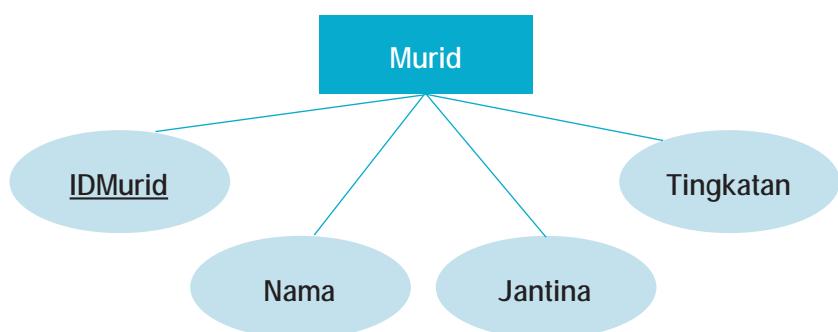
2.2.3 Mereka Bentuk Pangkalan Data Hubungan

Bagaimanakah suatu pangkalan data yang sesuai direka bentuk untuk kegunaan sistem maklumat?



Rajah 2.31 Pertimbangan dalam reka bentuk pangkalan data

Reka bentuk jadual hubungan perlu memenuhi pertimbangan seperti yang disenaraikan di atas. Perkara ini dilakukan secara sistematik. Setelah mengenal pasti entiti-entiti, atribut-atribut, hubungan antara entiti dan kardinaliti melalui ERD, langkah yang seterusnya adalah menghasilkan set hubungan. ERD bagi entiti Murid diberikan seperti dalam rajah yang berikut.



Rajah 2.32 ERD bagi entiti Murid

Set hubungan ialah set atribut dalam jadual hubungan. Set hubungan disediakan semasa proses merancang dan mereka bentuk pangkalan data yang menggambarkan entiti-entiti yang hendak disimpan, atribut yang diperlukan dan hubungan antara entiti tersebut. Set hubungan dihasilkan daripada ERD dan boleh dinyatakan dalam dua format, iaitu:



Tahukah Anda
Entiti terdiri daripada tiga jenis iaitu:

Entiti teguh
Entiti ini wujud dengan tidak bergantung pada entiti yang lain.

Entiti lemah
Entiti ini bergantung pada entiti sama jenis yang lain.

Entiti bersama
Entiti yang terjadi bersama dengan satu atau banyak jenis entiti yang lain.
Sumber: ecomputernotes.com



Uji Minda

Apakah peranan keduadua teknik dalam reka bentuk pangkalan data? Antara ERD dan skema hubungan, yang manakah lebih mudah disunting?



ERD sesuai digunakan untuk melakar entiti, atribut dan hubungan. Set hubungan lebih sesuai digunakan untuk analisis yang melibatkan struktur entiti yang diubah berulang kali.

(a) Penyataan teks

Format: NAMA ENTITI(Atribut 1<KP>, Atribut 2,...Atribut N)

Contoh 1: Murid(IDMurid<KP>, Nama, Jantina, Tingkatan)

Penyataan teks mudah untuk dihasilkan. Nama entiti ditulis dan kemudian diikuti oleh kurungan. Dalam kurungan, atribut-atribut entiti tersebut ditulis. Atribut yang berfungsi sebagai kunci primer ditandakan dengan menambah "<KP>" selepas nama atribut. Atribut yang berfungsi sebagai kunci asing pula ditandakan dengan menambah "<KA>". Contoh yang diberi menunjukkan entiti MURID mempunyai atribut-atribut IDMurid, Nama, Jantina dan Tingkatan. IDMurid dipilih sebagai kunci primer kerana IDMurid mewakili setiap baris rekod secara unik.

(b) Perwakilan grafik

Format:

NAMA ENTITI

Atribut 1<KP>	Atribut 2	Atribut...	Atribut-N
---------------	-----------	------------	-----------

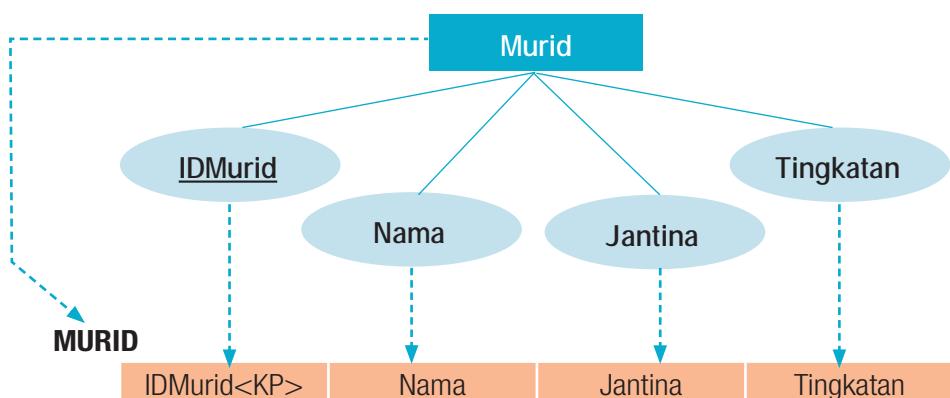
Contoh:

MURID

IDMurid<KP>	Nama	Jantina	Tingkatan
-------------	------	---------	-----------

Perwakilan grafik menggunakan jadual satu-baris. Nama entiti ditulis di atas baris jadual tersebut. Bilangan lajur yang diperlukan bergantung kepada bilangan atribut. Kunci primer sama seperti perwakilan teks, iaitu kunci primer ditanda dengan "<KP>" manakala kunci asing ditanda dengan "<KA>".

Anda boleh perhatikan bahawa set hubungan mengandungi butiran yang sama dengan ERD. Rajah yang berikut menunjukkan bagaimana ERD dapat diterjemahkan kepada set hubungan dengan mudahnya.



Rajah 2.33 Menukar ERD entiti murid kepada set hubungan



Apakah peranan kedua-dua teknik dalam reka bentuk pangkalan data? Antara ERD dan set hubungan, yang manakah lebih mudah disunting?

ERD sesuai digunakan untuk melakar entiti, atribut dan hubungan. Set hubungan lebih sesuai digunakan untuk analisis yang melibatkan struktur entiti yang disunting berulang kali.



37

Menukar ERD kepada Set Hubungan

SMK Pahlawan telah mengadakan satu pertandingan “Mereka cipta Replika Robot” di peringkat sekolah. Terdapat 50 penyertaan daripada murid Tingkatan 3 hingga Tingkatan 5. Setiap murid perlu mengisi borang penyertaan yang mempunyai maklumat seperti nama, nombor kad pengenalan, umur, jantina, tingkatan dan kelas.

Berdasarkan situasi di atas, anda dikehendaki menghasilkan ERD. Kemudian, tukar ERD kepada bentuk set hubungan.



Hubungan menunjukkan kaitan antara entiti. Hubungan menyediakan maklumat yang berguna untuk menjelaskan jenis sesuatu entiti.

Hubungan yang lemah menunjukkan kaitan yang lemah antara sesuatu entiti.

2.2.4 Kebergantungan fungsi sepenuh, separa dan transitif

Setiap atribut dalam jadual bergantung kepada atribut kunci primer. Sebagai contoh, dalam jadual murid, atribut-atribut seperti Nama, Jantina dan Tingkatan bergantung kepada atribut kunci primer, IDMurid. Menganalisis kebergantungan membolehkan sesebuah jadual dipecahkan kepada beberapa jadual hubungan yang lebih kecil untuk mengurangkan kelebihan data tanpa menjaskan integriti rujukan. Proses ini disebut penormalan dan memahami konsep kebergantungan amat penting untuk proses ini. Jadual 2.8 menunjukkan jenis-jenis kebergantungan.

Jadual 2.8 Jenis-jenis kebergantungan

Jenis Kebergantungan	Penerangan
Kebergantungan fungsi sepenuh	Berlaku apabila atribut-atribut bergantung sepenuhnya kepada kesemua atribut kunci dalam jadual.
Kebergantungan fungsi separa	Berlaku apabila atribut-atribut bergantung kepada salah satu daripada atribut kunci dalam jadual.
Kebergantungan fungsi transitif	Berlaku apabila atribut-atribut bergantung kepada atribut biasa yang lain dalam jadual.

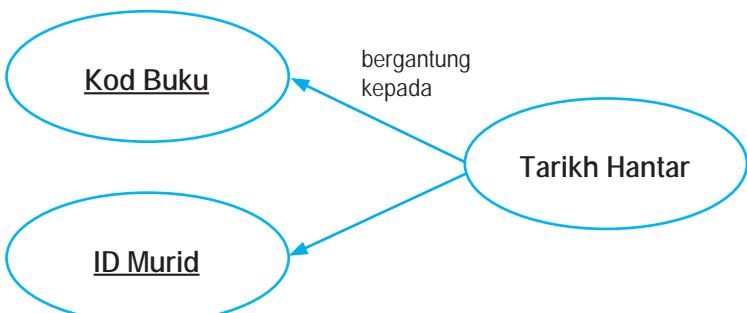
Untuk memahami dengan lebih lanjut, pertimbangkan contoh dalam Rajah 2.34. Jadual ini adalah jadual buku pinjaman untuk merekod butiran murid yang meminjam dan tarikh hantar buku tersebut. Dalam jadual ini, atribut-atribut Kod Buku dan ID Murid sahaja merupakan kunci primer.

Kod Buku	Nama Buku	Pengarang	ID Murid	Nama Murid	No Telefon Bimbit	Tarikh Pinjam	Tarikh Hantar
IPB124044	Java Programming	Maruyama et al	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB257868	C Programming	M. A. Bakar	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB192254	Computer Graphics	Shirley Jr	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB051375	Software Engineering	Roger Estain	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016
IPB051325	Information Systems, an Introduction	Farah et al	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016

Rajah 2.34 Jadual BUKU PINJAMAN

Kebergantungan Fungsi Sepenuh

Perhatikan bahawa atribut Tarikh Hantar bergantung kepada dua kunci primer iaitu Kod Buku dan ID Murid. Sekiranya tiada salah satu kunci primer, Tarikh Hantar buku tidak dapat ditentukan. Jadi, Tarikh Hantar mempunyai kebergantungan fungsi sepenuh kepada kedua-dua kunci primer.

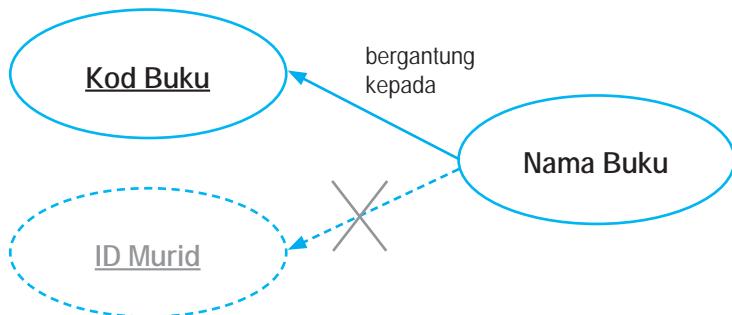


Rajah 2.35 Contoh kebergantungan fungsi sepenuh

Satu lagi kebergantungan fungsi sepenuh adalah antara TarikhPinjam dengan kunci primer NoBuku dan IDPeminjam.

Kebergantungan Fungsi Separa

Perhatikan juga bahawa dalam Rajah 2.36, atribut Nama Buku bergantung kepada Kod Buku sahaja dan bukan kedua-dua Kod Buku dan ID Murid. Jadi, Nama Buku mempunyai kebergantungan fungsi separa kepada kod Buku.



Rajah 2.36 Contoh kebergantungan fungsi separa



Selain daripada tiga jenis kebergantungan fungsi yang telah anda pelajari, masih terdapat dua jenis kebergantungan fungsi iaitu, kebergantungan fungsi tunggal dan kebergantungan fungsi trivial.

Kebergantungan Fungsi Transitif

Kebergantungan fungsi transitif ialah kebergantungan yang tidak melibatkan mana-mana atribut kunci primer dalam jadual Rajah 2.37. Jadi, kebergantungan di antara atribut Nama Murid dengan atribut No Telefon Bimbit adalah jenis kebergantungan fungsi transitif. Dalam contoh yang diberikan, memang diketahui secara umum bahawa No Telefon Bimbit adalah unik kepada pemiliknya. Oleh itu, adalah munasabah jika kebergantungan wujud di antara Nama Murid dan No Telefon Bimbit.



Rajah 2.37 Contoh kebergantungan fungsi transitif



38

Kebergantungan Fungsi

Individu

Anda ditugaskan untuk membina jadual peserta yang menyertai bagi satu Kempen Membaca. Dalam jadual tersebut perlu memiliki nama peserta, nombor kad pengenalan peserta, nombor telefon bimbit, tingkatan dan kelas. Daripada jadual yang telah anda bina, kenal pasti kunci primer dan kunci asing. Kemudian, berdasarkan Rajah 2.35, Rajah 2.36 dan Rajah 2.37, bina rajah-rajabagi kebergantungan fungsi sepenuh, kebergantungan fungsi separa dan kebergantungan fungsi transitif.

2.2.5 Proses Penormalan Data Sehingga 3NF Terhadap

Skema Hubungan

Jadual baharu biasanya berada dalam bentuk tidak ternormal, ataupun ONF. Jadual ONF boleh berfungsi tetapi akan menyebabkan penyimpanan data lewah yang banyak dan boleh menjelaskan integriti data apabila kecuaian berlaku sewaktu kemas kini. Oleh itu, jadual ONF tidak sesuai dilaksanakan dalam sistem pangkalan data.

Contoh jadual ONF adalah seperti dalam Rajah 2.38. Banyak duplikasi nilai atribut seperti ID Murid, Nama Murid dan No Telefon Bimbit untuk setiap pinjaman buku yang dapat dilihat. Perhatikan juga bahawa lajur-lajur dalam jadual ONF adalah tidak atomik kerana terdapat gabungan atribut-atribut untuk membentuk satu lajur. Lajur Buku mengandungi gabungan lajur-lajur Kod Buku, Nama Buku dan Pengarang.

Jadual ONF mempunyai banyak duplikasi nilai atribut

Buku			Murid			Tarikh Pinjam	Tarikh Hantar
Kod Buku	Nama Buku	Pengarang	ID Murid	Nama Murid	No Telefon Bimbit		
IPB124044	Java Programming	Maruyama et al	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB257868	C Programming	M. A. Bakar	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB192254	Computer Graphics	Shirley Jr	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB051375	Software Engineering	Roger Estain	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016
IPB051325	Information Systems, an Introduction	Farah et al	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016

Rajah 2.38 Jadual PINJAMAN BUKU dalam bentuk ONF

Oleh sebab itu, sesebuah jadual baharu perlu disemak terlebih dahulu sebelum dilaksanakan dalam sistem pangkalan data. Semak sama ada jadual tersebut sudah ternormal ataupun tidak. Jika belum ternormal, ambil langkah-langkah yang sepatutnya untuk mengubah jadual tersebut kepada bentuk ternormal.

Penormalan ialah satu kaedah menganalisis jadual-jadual berdasarkan atribut kunci dan kebergantungan fungsi dengan tujuan mengurangkan duplikasi data dalam pangkalan data. Penormalan, pada kebiasaannya, akan memecahkan jadual ONF kepada dua atau lebih **jadual-jadual hubungan** yang sudah ternormal, lihat Rajah 2.39.

Jadual asal (ONF): BUKU PINJAMAN

Buku			Murid			Tarikh Pinjam	Tarikh Hantar
Kod Buku	Nama Buku	Pengarang	ID Murid	Nama Murid	No Telefon Bimbit		
IPB124044	Java Programming	Maruyama et al	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB257868	C Programming	M. A. Bakar	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB192254	Computer Graphics	Shirley Jr	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB051375	Software Engineering	Roger Estain	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016
IPB051325	Information Systems, an Introduction	Farah et al	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016

Penormalan

Jadual PINJAMAN

Kod Buku	ID Murid	Tarikh Pinjam	Tarikh Hantar
IPB124044	125007	27-Sep-2016	10-Okt-2016
IPB257868	125007	27-Sep-2016	10-Okt-2016
IPB192254	125007	27-Sep-2016	10-Okt-2016
IPB051375	125096	28-Sep-2016	11-Okt-2016
IPB051325	125096	28-Sep-2016	11-Okt-2016

Jadual MURID

ID Murid	Nama Murid
125007	Harris bin Aman
125096	Francis Embong

Jadual BUKU

Kod Buku	Nama Buku	Pengarang
IPB124044	Java Programming	Maruyama et al
IPB257868	C Programming	M. A. Bakar
IPB192254	Computer Graphics	Shirley Jr
IPB051375	Software Engineering	Roger Estain
IPB051325	Information Systems, an Introduction	Farah et al

Jadual TELEFON

Nama Murid	No Telefon Bimbit
Harris bin Aman	0176677889
Francis Embong	0121122335

Rajah 2.39 Normalisasi jadual ONF BUKU PINJAMAN kepada jadual-jadual hubungan 3NF PINJAMAN, BUKU, MURID dan TELEFON

Penormalan dilakukan secara sistematik dan berperingkat. Umumnya, terdapat tiga bentuk penormalan iaitu 1NF, 2NF dan 3NF. Penormalan dibuat sehingga jadual mencapai peringkat 2NF ataupun 3NF. Ikuti langkah-langkah seperti yang berikut:

1 Tukarkan skema perhubungan ONF kepada 1NF

Objektif penukaran adalah untuk memastikan lajur jadual adalah atomik dan mempunyai kunci primer. Langkah pertama ialah memastikan keatomikan data-data dengan menggunakan satu lajur untuk setiap satu atribut. Lihat Rajah 2.40.

Kunci primer

Kod Buku	Nama Buku	Pengarang	ID Murid	Nama Murid	No Telefon Bimbit	Tarikh Pinjam	Tarikh Hantar
IPB124044	Java Programming	Maruyama et al	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB257868	C Programming	M. A. Bakar	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB192254	Computer Graphics	Shirley Jr	125007	Harris bin Aman	0176677889	27-Sep-2016	10-Okt-2016
IPB051375	Software Engineering	Roger Estain	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016
IPB051325	Information Systems, an Introduction	Farah et al	125096	Francis Embong	0121122335	28-Sep-2016	11-Okt-2016

Rajah 2.40 Jadual PINJAMAN BUKU dalam bentuk 1NF

Berdasarkan jadual 1NF, hasilkan skema hubungan. Skema hubungan menunjukkan struktur jadual dalam pangkalan data. Gunakan nama jadual (entiti) dalam huruf besar diikuti oleh senarai atribut dalam kurungan. Sintaks skema hubungan adalah seperti yang berikut:

```
NAMA ENTITI (atribut 1 kunci primer <KP>, atribut 2 kunci asing <KA>,  
atribut 3, atribut 4, ...)
```

Skema hubungan 1NF mengandungi semua atribut dalam jadual asal. Kenal pasti kunci primer dengan mengkaji hubungan kebergantungan fungsi sepenuh antara atribut-atribut dalam skema ataupun jadual. Tandakan atribut yang menjadi kunci primer menggunakan akhiran "<KP>". Lihat contoh berikut.

Contoh:

```
BUKU PINJAMAN (Kod Buku <KP>, Nama Buku, Pengarang, ID Murid <KP>, Nama  
Murid, No Telefon Bimbit, Tarikh Pinjam, Tarikh Hantar)
```

2 Tukarkan skema-skema hubungan 1NF kepada 2NF

Objektif penukaran bentuk 1NF kepada 2NF adalah untuk menghapuskan kebergantungan fungsi separa. Jadual 1NF mempunyai kebergantungan fungsi separa. Oleh itu, kenal pasti kumpulan-kumpulan data berulang dan pecahkan kepada jadual-jadual berasingan yang dipanggil jadual hubungan.

Kaji skema 1NF dengan mencari kebergantungan antara atribut bukan-kunci dengan atribut kunci primer. Kebergantungan fungsi separa berlaku apabila atribut biasa bergantung kepada salah satu atribut kunci primer sahaja. Kenal pasti kumpulan atribut tersebut dan asingkan sebagai skema hubungan yang baharu.

Contoh:

Mulakan dengan menganalisis jadual 1NF BUKU PINJAMAN. Dalam skema yang asal, didapati daripada data-data dalam jadual bahawa atribut-atribut yang mempunyai kumpulan data berulang ialah ID Murid, Nama Murid dan No Telefon Bimbit. Gunakan kurungan untuk menandakan kumpulan data yang berulang.

BUKU PINJAMAN (Kod Buku <KP>, Nama Buku, Pengarang, (ID Murid <KP>, Nama Murid, No Telefon Bimbit), Tarikh Pinjam, Tarikh Hantar)

Asingkan atribut-atribut bagi kumpulan data yang berulang dan berikan nama entiti Murid.

BUKU PINJAMAN (Kod Buku <KP>, Nama Buku, Pengarang, (), Tarikh Pinjam, Tarikh Hantar)

MURID (ID Murid <KP>, Nama Murid, No Telefon Bimbit)

Tukar nama jadual BUKU PINJAMAN kepada PINJAMAN. Bagi entiti Murid, jadikan ID Murid sebagai kunci primer kerana ID Murid membantu pengenalan nilai-nilai atribut lain secara unik. Tambah akhiran <KA> kepada ID Murid. Atribut ID Murid yang sama dikekalkan dalam jadual PINJAMAN tetapi dijadikan juga sebagai atribut kunci asing, tambah akhiran <KA>.

PINJAMAN (Kod Buku <KP>, Nama Buku, Pengarang, ID Murid <KP> <KA>, Tarikh Pinjam, Tarikh Hantar)

menghubungkan

MURID (ID Murid <KP>, Nama Murid, No Telefon Bimbit)

Nota:

- Semak jadual Pinjaman. Pastikan atribut Kod Buku dan ID Murid kekal sebagai kunci primer. Kekalkan akhiran <KP> bagi kedua-dua atribut tersebut.
- Pastikan jadual Murid mempunyai atribut ID Murid sebagai kunci primer.
- Pastikan jadual-jadual dihubungkan melalui pasangan atribut kunci primer dan kunci asing yang betul, iaitu ID Murid <KA> dalam jadual PINJAMAN kepada ID Murid <KP> dalam jadual MURID.

Kedua-dua jadual PINJAMAN dan MURID yang dihasilkan masih berada dalam 1NF selagi mengandungi kebergantungan fungsi separa. Marilah menganalisis kedua-dua jadual tersebut untuk sebarang kemungkinan kewujudan saki-baki kebergantungan fungsi separa.

Adakah jadual 1NF PINJAMAN mempunyai kebergantungan fungsi separa?

bergantung fungsi separa kepada



PINJAMAN (Kod Buku <KP>, Nama Buku, Pengarang, ID Murid <KP> <KA>, Tarikh Pinjam, Tarikh Hantar)

- Dalam jadual PINJAMAN, Tarikh Pinjam dan Tarikh Hantar bergantung penuh kepada kedua-dua kunci primer Kod Buku dan ID Murid. Ini adalah kebergantungan fungsi sepenuh.
- Atribut-atribut Nama Buku dan Pengarang bergantung kepada kunci primer Kod Buku sahaja walaupun ID Murid juga adalah kunci primer. Oleh itu, jadual PINJAMAN mempunyai kebergantungan fungsi separa di antara Nama Buku dan Pengarang dengan Kod Buku.

Oleh itu, kumpulan atribut data dengan kebergantungan fungsi separa diasingkan sebagai skema hubungan baharu. Berikan nama entiti Buku.

PINJAMAN (Kod Buku <KP> <KA>, ID Murid <KP> <KA>, Tarikh Pinjam, Tarikh Hantar)

menghubungkan

BUKU (Kod Buku <KP>, Nama Buku, Pengarang)

Nota:

- Semak jadual BUKU. Atribut Kod Buku dijadikan kunci primer.
- Semak jadual PINJAMAN. Atribut Kod Buku kekal sebagai salah satu kunci primer. Pada masa yang sama, Kod Buku juga menjadi kunci asing untuk menghubungkan jadual PINJAMAN kepada jadual BUKU. Tambah akhiran <KA> kepada Kod Buku jadual PINJAMAN.
- Semak kedua-dua jadual untuk kewujudan lain-lain kebergantungan fungsi separa. Jika tidak ada, maka jadual PINJAMAN dan jadual BUKU sudah menjadi jadual 2NF.

Gunakan analisis yang sama ke atas jadual 1NF MURID. Jadual MURID tiada kebergantungan fungsi separa. Oleh itu, jadual MURID sudah berada dalam bentuk 2NF.

MURID (ID Murid <KP>, Nama Murid, No Telefon Bimbit)

Hasil pernormalan menghasilkan skema perhubungan jadual-jadual 2NF berikut.

PINJAMAN (Kod Buku <KP> <KA>, ID Murid <KP> <KA>, Tarikh Pinjam, Tarikh Hantar)

BUKU (Kod Buku <KP>, Nama Buku, Pengarang)

MURID (ID Murid <KP>, Nama Murid, No Telefon Bimbit)



3 Tukarkan skema-skema perhubungan 2NF kepada 3NF

Objektif penukaran adalah untuk menghapuskan kebergantungan fungsi transitif. Pada kebiasaannya, penormalan sehingga tahap 2NF sudah memadai. Penormalan ke tahap 3NF cuma perlu dalam situasi di mana terdapat kebergantungan fungsi transitif di antara atribut dalam sesetengah jadual. Kebergantungan ini tersembunyi kerana wujud di antara atribut-atribut biasa, iaitu tidak melibatkan atribut kunci primer. Biasanya, ini dapat dikenal pasti daripada pengalaman penggunaan data-data.

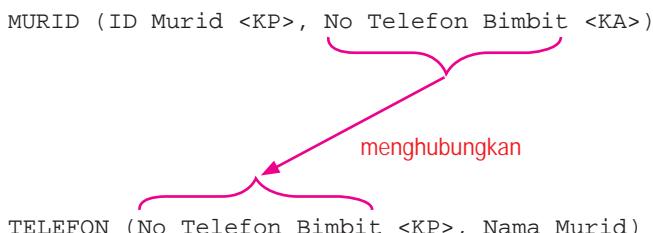
Contoh:

Kaji skema hubungan jadual MURID untuk mencari kebergantungan fungsi transitif:



Jika dilihat dengan teliti, skema MURID mempunyai tiga atribut ID Murid, Nama Murid dan No Telefon Bimbit. ID Murid ialah kunci primer dan oleh itu kedua-dua atribut Nama Murid dan No Telefon Bimbit bergantung kepada ID Murid. Tetapi dalam masa yang sama, atribut Nama Murid turut bergantung kepada No Telefon Bimbit walaupun No Telefon Bimbit bukan kunci primer. Ini adalah kebergantungan fungsi transitif.

Oleh itu, asingkan fungsi transitif tersebut dengan menghasilkan satu skema baharu, TELEFON. Skema hubungan TELEFON mengandungi atribut Nama Murid dan No Telefon Bimbit. Jadikan Nama Murid sebagai kunci primer. Dalam skema jadual MURID, atribut No Telefon Bimbit dijadikan kunci asing.



Bagaimanakah dengan jadual-jadual 2NF yang lain seperti PINJAMAN dan BUKU? Oleh sebab analisis ke atas kedua-dua skema jadual PINJAMAN dan BUKU tidak meghasilkan kebergantungan baharu, maka jadual-jadual tersebut tidak mempunyai bentuk 3NF.

Tahukah
Anda ?

Terdapat bentuk penormalan yang lebih tinggi daripada 3NF. Contohnya ialah Bentuk Penormalan Boyce-Codd (BCNF).



goo.gl/Pv1IaY



BAB 2 PANGKALAN DATA



Sebagai kesimpulan, bentuk-bentuk normal (*normal forms*) ialah peringkat-peringkat penukaran sesuatu skema hubungan tak-ternormal kepada skema-skema hubungan yang memenuhi kriteria bentuk normal. Kriteria-kriteria bentuk normal (NF) dirumuskan seperti yang berikut:

Jadual 2.9 Jenis-jenis bentuk normal (Normal Form, NF)

Bentuk normal	Penjelasan
1NF	<p>1. Objektif:</p> <ul style="list-style-type: none"> • Pastikan lajur adalah atomik. • Kenal pasti kunci primer dalam jadual. <p>2. Isu: Jadual mempunyai kedua-dua kebergantungan fungsi sepenuh dan kebergantungan fungsi separa.</p> <p>Contoh:</p> <p>1NF:</p> <p>BUKU PINJAMAN (Kod Buku <KP>, Nama Buku, Pengarang, (ID Murid <KP>, Nama Murid, No Telefon Bimbit), Tarikh Pinjam, Tarikh Hantar)</p>
2NF	<p>1. Objektif:</p> <ul style="list-style-type: none"> • Jadual tiada kebergantungan fungsi separa. • Semua jadual hubungan mempunyai kebergantungan fungsi sepenuh sahaja. • Setiap satu jadual hubungan mempunyai atribut kunci primer. <p>2. Isu: Namun, mungkin terdapat kebergantungan fungsi transitif antara atribut-atribut yang belum dikenal pasti.</p> <p>Contoh:</p> <p>2NF:</p> <p>PINJAMAN (Kod Buku <KP> <KA>, ID Murid <KP> <KA>, Tarikh Pinjam, Tarikh Hantar)</p> <p>BUKU (Kod Buku <KP>, Nama Buku, Pengarang)</p> <p>MURID (ID Murid <KP>, Nama Murid, No Telefon Bimbit)</p>
3NF	<p>1. Objektif: Hapuskan kebergantungan fungsi transitif dalam jadual jika ada.</p> <p>2. Isu: Tiada isu. Akhirnya, semua jadual hubungan tidak mempunyai kebergantungan fungsi separa mahupun transitif.</p> <p>Contoh:</p> <p>3NF:</p> <p>PINJAMAN (Kod Buku <KP> <KA>, ID Murid <KP> <KA>, Tarikh Pinjam, Tarikh Hantar)</p> <p>BUKU (Kod Buku <KP>, Nama Buku, Pengarang)</p> <p>MURID (ID Murid <KP>, No Telefon Bimbit <KA>)</p> <p>TELEFON (No Telefon Bimbit <KP>, Nama Murid)</p>

Ciri-ciri bentuk penormalan perlu diingat secara teliti kerana boleh membantu analisis jenis bentuk normal jadual dan menukar bentuk normal tersebut kepada peringkat yang lebih tinggi, iaitu 2NF dan kadang-kadang 3NF.

Diberikan sampel jadual PEPERIKSAAN seperti berikut:

Kertas		Murid			Tarikh Pinjam	Tarikh Hantar	Gred
Kod Ujian	Subjek	ID Murid	Nama Murid	Tingkatan			
S0123	Bahasa Melayu	125007	Harris bin Samad	4 Aman	27-Sep-2016	91	A+
S0123	Bahasa Melayu	125096	Francis Embong	4 Aman	27-Sep-2016	82	A
S0133	Bahasa Inggeris	125007	Harris bin Samad	4 Aman	28-Sep-2016	72	A-
S0211	Matematik Moden	125096	Francis Embong	4 Aman	29-Sep-2016	80	A
S0211	Matematik Moden	125128	Dewi Surinam	4 Aman	29-Sep-2016	75	A-

Dalam satu pasangan, hasilkan skema hubungan yang lengkap dan lukis semula jadual di atas bagi setiap bentuk normal berikut:

- (a) 1NF
- (b) 2NF
- (c) 3NF

Latihan Formatif 2.2

1 Senaraikan kepentingan kunci primer.

2 Jadual Vendor

KodVendor	NamaVendor	TelefonVendor	PostkadVendor
V00201	YK Dynamic Supply	05-3560123	31650
V00205	Delight Baker	05-3680020	31750
V00392	Suria Kitchen	03-5040235	60000

Jadual Produk

KodProduk	Produk	HargaSeunit	KodVendor
bk00001234	Tepung naik sendiri (tin)	12.00	V00205
bk00000689	Tepung gandum (guni 5kg)	60.00	V00205
bk00000023	Air Tebu (botol)	2.50	V00392

Rujuk jadual hubungan yang diberikan dan jawab soalan 2 dan 3.

- (a) Kenal pasti kunci primer dalam entiti Vendor.
- (b) Kenal pasti kunci primer dalam entiti Produk.
- (c) Adakah terdapat kunci asing dalam jadual-jadual di atas? Jelaskan jawapan anda.

3 Tandakan betul (✓) atau salah (✗).

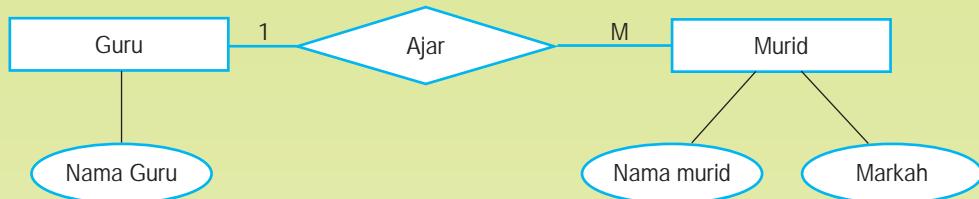
- (a) Atribut KodVendor boleh tidak menerima sebarang nilai. ()
- (b) Atribut TelefonVendor boleh tidak menerima sebarang nilai. ()
- (c) Dalam entiti Produk, kunci primer ialah KodProduk manakala kunci asing ialah Produk. ()
- (d) Setiap entiti mesti mengandungi satu kunci primer. ()
- (e) Setiap entiti mesti mengandungi satu kunci asing. ()

4 Lukiskan ERD untuk kenyataan "Pelanggan membeli Produk"

5 Lukisan ERD untuk persekitaran berikut:

- Akaun bank mempunyai butiran nombor akaun dan baki akaun.
- Seorang pelanggan bank mesti memiliki sekurang-kurangnya satu akaun.
- Pelanggan mempunyai butiran nama dan nombor kad pengenalan.

6 Berikan tafsiran anda untuk model dalam ERD berikut:



7 Sekiranya berlaku kesilapan dalam mereka bentuk pangkalan data, apakah yang anda perlu lakukan? Senaraikan tindakan yang mungkin akan anda ambil untuk menyelesaikan masalah tersebut.



8 Tentukan "benar" atau "palsu" untuk pernyataan-pernyataan berikut:

- (a) Penormalan menggabungkan beberapa jadual menjadi jadual tunggal. ()
- (b) Penormalan bertujuan menghasilkan pencarian yang cepat dengan menghasilkan duplikasi data di jadual-jadual 2NF. ()
- (c) Penormalan mengurangkan kelebihan dan meninggikan integriti pangkalan data. ()
- (d) Atribut kunci primer membolehkan setiap rekod dikenal pasti secara unik. ()
- (e) Kunci asing adalah kunci primer dalam jadual sasaran. ()
- (f) Data lewah dan integriti data adalah dua masalah pangkalan data yang tidak berkaitan. ()
- (g) Masalah integeriti data akan menjadi ketara apabila data perlu dikemas kini. ()
- (h) Jadual-jadual dalam 3NF tidak mempunyai isu integriti data kerana strukturnya tidak membenarkan duplikasi data. ()

9 Lihat rajah yang merumuskan langkah-langkah penormalan jadual berikut:



- (a) Kenal pasti bentuk normal (i) dan (ii).
- (b) Apakah objektif (i)?
- (c) Apakah objektif (ii)?

10 Berdasarkan jadual REKOD PELANGGAN di bawah, hasilkan penormalan ONF ke 3NF



REKOD PELANGGAN								
ID Pelanggan	Nama Pelanggan	Alamat	Tarikh Hantar	Tarikh Ambil	Kod Servis	Jenis Servis	Kod Harga	Harga Servis
A0001	Albert anak Thomas	Bintulu	01-Sep-16	03-Sep-16	K001	Tukar hard disk	KK001	250.00
A0200	Hanif	Muar	01-Sep-16	02-Sep-16	K010	Tukar Motherboard	KK010	1200.00
A8000	Tan Kok Seng	Kuala Lumpur	02-Aug-16		B001	Beli papan kekunci	BB001	12.00
B1000	Nurul	Kuala Krai	15-Aug-16		B001	Beli papan kekunci	BB001	12.00
B1100	Thilakamalar	Setia Alam	14-Aug-16		B002	Beli tetikus	BB002	8.00
A0001	Albert anak Thomas	Bintulu	14-Aug-16		B001	Beli papan kekunci	BB001	12.00



STANDARD PEMBELAJARAN

2.3.1 Membina jadual berpaduan skema hubungan menggunakan perisian pangkalan data hubungan

2.3.2 Mencipta borang yang berkaitan dengan kandungan jadual

2.3.3 Memasukkan data dalam jadual melalui borang

2.3.4 Menghasilkan *query* untuk mendapatkan maklumat yang diperlukan semula daripada:

- (i) Satu jadual menggunakan kriteria lebih daripada satu
- (ii) Dua jadual menggunakan satu kriteria

2.3.5 Menjana laporan berdasarkan hasil *query*

2.3.6 Menghasilkan satu sistem maklumat mudah melalui makro menggunakan menu (*switchboard*) yang mengandungi:

- (i) Borang
- (ii) Laporan
- (iii) Keluar dari sistem

2.3.7 Mendokumentasikan hasil kerja

Pembangunan Pangkalan Data Hubungan

Pangkalan data ialah koleksi bersepadan bagi rekod data, fail ataupun objek-objek pangkalan data. Sebagai perumpamaan, pangkalan data adalah seperti sebuah kabinet fail yang digunakan untuk menyimpan, menyusun, mengemas kini dan mencari rekod data dengan pantas, tepat, kemas dan mudah.



Rajah 2.41 Penggunaan pangkalan data sebagai ganti kepada penyimpanan dan pencarian kabinet fail

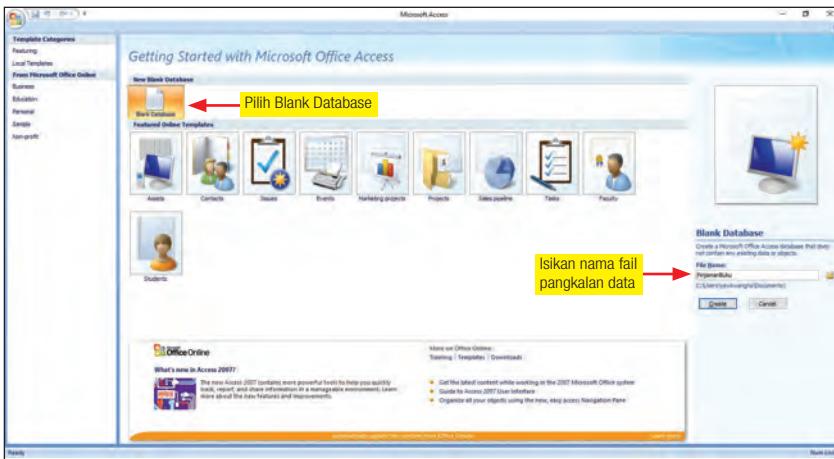
Penghasilan pangkalan data

Pangkalan data dibangunkan dengan menggunakan perisian sistem pengurusan pangkalan data atau *database management system* (DBMS). Sistem pengurusan pangkalan data (DBMS) merupakan pakej perisian dengan atur cara komputer bagi tujuan membangunkan, menggunakan dan menyelenggarakan pangkalan data. Contohnya ialah **Microsoft Access**, **Microsoft SQL**, **MySQL** dan **Oracle SQL**.

Microsoft Access ialah salah satu aplikasi dalam *Microsoft Office*. Kelebihan *Microsoft Access* dalam mempelajari konsep pangkalan data ialah penggunaan “Antara muka Pengguna Grafikal” ataupun *Graphical User Interface* (GUI). Untuk membina pangkalan data, lancarkan *Microsoft Access* dan pilih pangkalan data yang kosong daripada tetingkap dialog. Masukkan nama dan lokasi simpanan pangkalan data. Lihat Rajah 2.42.



Rajah 2.42 *Microsoft Access* merupakan salah satu aplikasi dalam *Microsoft Office*



Rajah 2.43 Mencipta pangkalan data

Pembangunan pangkalan data hubungan melibatkan beberapa aktiviti mengikut susunan seperti yang ditunjukkan dalam Rajah 2.43.



Rajah 2.44 Aktiviti-aktiviti pembangunan pangkalan data hubungan

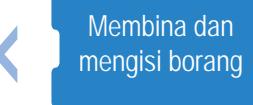
2.3.1 Pembinaan Jadual Berpandukan Skema Hubungan Menggunakan Perisian Pangkalan Data Hubungan

Pangkalan data mengandungi jadual (*table*). **Jadual hubungan** menghuraikan struktur logikal data sebagai kombinasi atribut yang mengenal pasti setiap baris rekod secara unik. Jadual hubungan mewakili

Tahukah Anda?

Database Management System (DBMS) berfungsi untuk menyelesaikan beberapa tugas. Misalnya:

- Membina rangkaian perniagaan melalui internet dan menyebarkan sistem maklumat.
- Sebagai storan dan menghasilkan imej grafik, video dan audio.
- Membina pejabat secara maya
- Memproses data yang kompleks
- Membina laman sesawang



Uji Minda

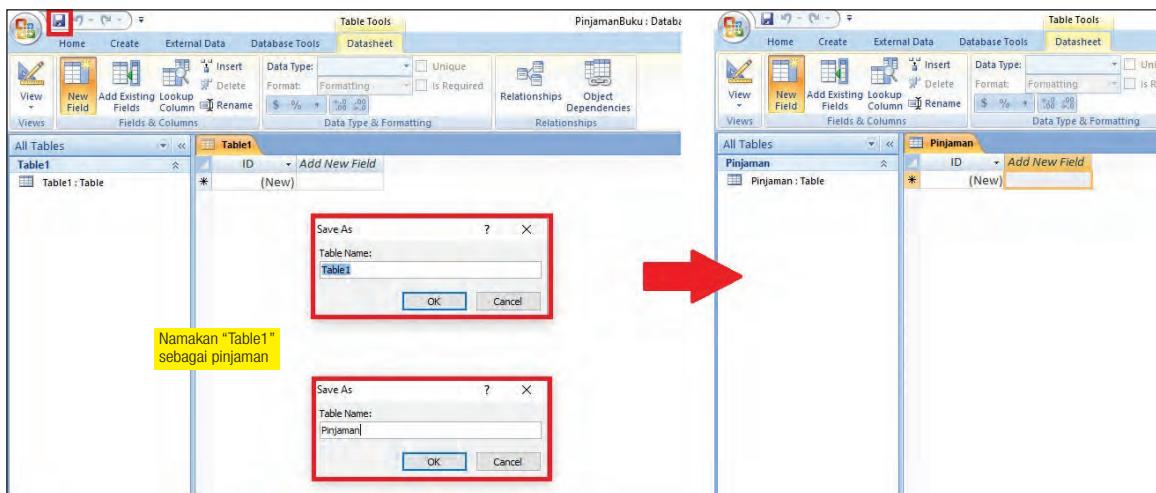
Anda telah mengetahui bahawa salah satu aplikasi untuk DBMS ialah *Microsoft Access*. Berikan nama-nama aplikasi yang lain untuk DBMS. Nyatakan kelebihan dan kekurangan setiap aplikasi yang diberikan.



entiti. Kejadian entiti-entiti yang mempunyai koleksi atribut yang sama akan berada di dalam jadual hubungan yang sama sebagai rekod.

1 Cipta jadual baharu

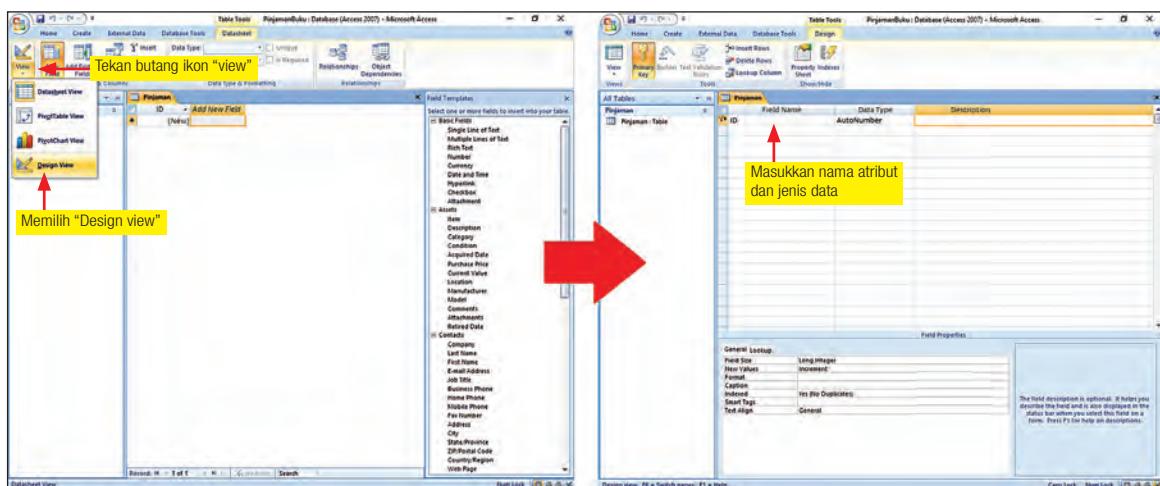
Rajah 2.45 menunjukkan jadual kosong dengan nama “Table1”. Nama “Table1” dimasukkan secara automatik. Nama tersebut ditukar dengan mengklik ikon “Save” untuk menyimpan jadual dengan nama yang bersesuaian. Dalam Rajah 2.45, jadual “Table1” akan dinamakan semula dengan nama baharu, iaitu “Pinjaman”.



Rajah 2.45 Mencipta Pangkalan Data

2 Buka “Design View”

Jadual mengandungi rekod. Setiap baris ialah satu rekod. Setiap rekod mempunyai maklumat yang disusun atur dalam atribut (*fields*). Jadual “Pinjaman” tidak mempunyai sebarang atribut.

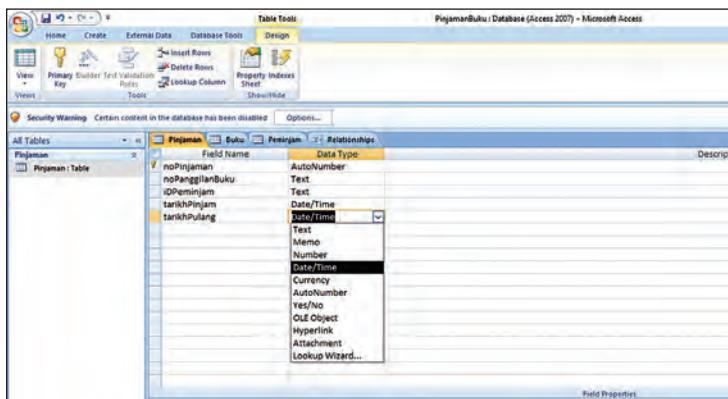


Rajah 2.46 Menukar mod jadual daripada “Datasheet View” kepada “Design View”

Untuk menambahkan atribut-atribut baharu, pastikan jadual berada dalam mod "Design View". Mod asal ialah "Datasheet View". Lihat Rajah 2.46.

3 Set jenis data

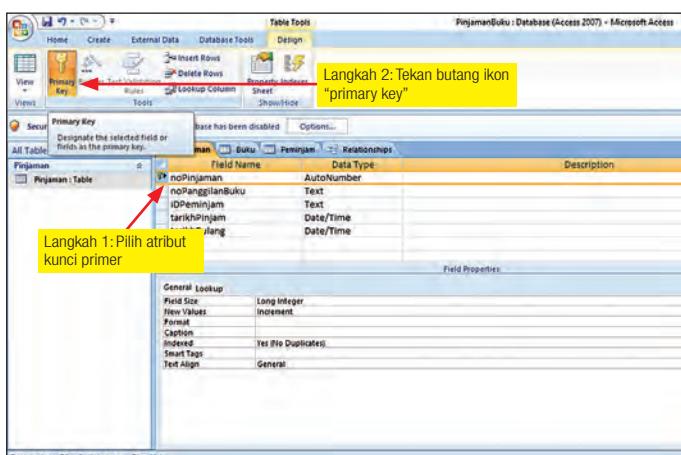
Dalam mod "Design View", pembangun pangkalan data boleh memberikan nama atribut, jenis data dan deskripsi atribut tersebut. Setiap baris adalah untuk atribut yang baharu. Dalam mod "Datasheet View", setiap baris ialah rekod. Lihat Rajah 2.47. Perhatikan bahawa terdapat empat atribut dalam contoh tersebut.



Rajah 2.47 Menambah atribut-atribut baharu

4 Set kunci primer

Setiap jadual memerlukan kunci primer. Kunci primer ialah atribut yang dipilih kerana nilai atribut tersebut unik dan mewakili rekod tersebut. Microsoft Access memilih atribut tertentu sebagai kunci primer secara automatik. Biasanya atribut yang dipilih ialah atribut pertama. Namun, pembangun pangkalan data boleh mengubah dengan mengklik atribut pada baris lain. Kemudian, klik ikon kunci primer seperti dalam Rajah 2.48.



Rajah 2.48 Menjadikan atribut tertentu sebagai kunci primer

Pelbagai jenis data terdapat dalam Access. Antaranya ialah:

Text

- Teks atau kombinasi teks dengan nombor
- Saiz maksimum sepanjang 255 aksara atau seperti yang ditetapkan oleh Field Size' dalam tetingkap 'General Property'
- Boleh diisi atau digunakan sebagai indeks

Memo

- Teks atau kombinasi teks dengan nombor yang panjang
- Panjang maksimum ialah 65,535 aksara
- Tidak boleh diisi atau digunakan sebagai indeks

Number

- Tidak seperti nombor dengan jenis data 'TEXT'
- Boleh diproses dengan menggunakan operasi aritmetik

Date/Time

- Nombor yang khusus untuk tarikh dan masa

5 Menambah jadual-jadual lain

Sebuah pangkalan data biasanya mengandungi lebih daripada satu jadual hubungan ternormal. Untuk menambahkan jadual yang baharu, klik tab "Create" dan ikon "Table". Lihat Rajah 2.49.

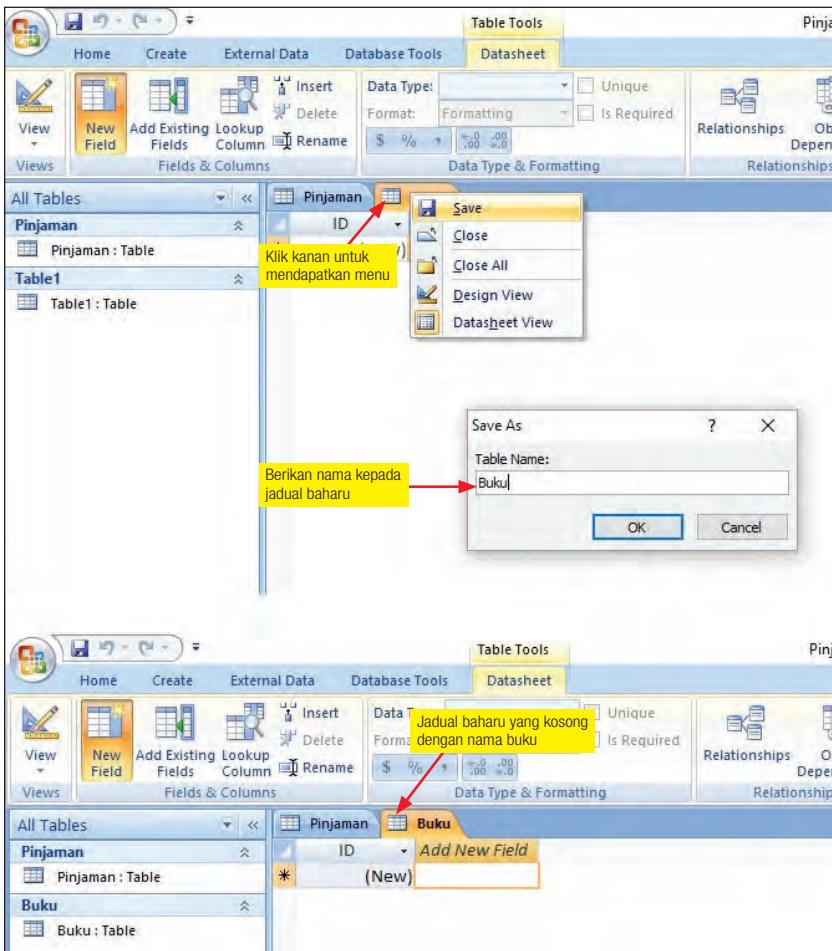
The screenshot shows the Microsoft Access ribbon with the 'Create' tab selected. A callout bubble points to the 'Table' icon in the 'Tables' group. The 'Pinjaman' table is open in Datasheet view, displaying four fields: 'noPanggilanBuku' (Text), 'iDPeminjam' (Text), 'tarikhPinjam' (Date/Time), and 'tarikhPulang' (Text). A large red arrow points from the 'Create' tab down to the 'Database Tools' tab, which is also selected. The 'Data Type & Formatting' section shows a note: 'Jadual kosong terhasil'. The 'Table Tools' tab is highlighted in yellow. The status bar at the bottom shows 'Table1'.

Rajah 2.49 Menambahkan jadual baharu

Kemudian, klik kanan butang tetikus dan pilih ikon "Save" untuk menyimpan jadual baharu dengan memberikan jadual, nama yang bersesuaian. Lihat Rajah 2.50.

4 examples of Database Application
goo.gl/7sked9





Rajah 2.50 Menamakan jadual

Seperti langkah-langkah yang sebelumnya, masukkan atribut-atribut dengan jenis-jenis data yang bersesuaian. Tukar jadual Buku ke mod “Design View” dan masukkan atribut-atribut dan jenis data yang bersesuaian daripada Jadual 2.10. Lihat Rajah 2.51. Set noPanggilan Buku sebagai kunci primer.

Jadual 2.10 Atribut-atribut dalam jadual Buku

Buku	
Atribut	Jenis data
noPanggilanBuku	Teks
namaBuku	Teks
pengarang	Teks
subjek	Teks



Aplikasi Harian

Anda boleh mempelajari cara menggunakan *Microsoft Access* melalui laman sesawang di bawah:

[http://www.profsr.com/
access/accless02.htm](http://www.profsr.com/access/accless02.htm)



Melalui laman sesawang ini, anda akan diterangkan terlebih dahulu mengenai *Microsoft Access* dan kemudian disediakan tutorial menarik dan mudah untuk diikuti. Selamat Mencuba!

The screenshot shows the Microsoft Access application interface in Design View. The ribbon at the top has 'Table Tools' selected under the 'Design' tab. On the left, there's a navigation pane titled 'All Tables' with entries for 'Pinjaman' and 'Buku'. The main area displays the 'Buku' table structure with four fields: 'noPanggilanBuku', 'namaBuku', 'pengarang', and 'subjek'. Each field is defined as 'Text' type with a description provided.

	Field Name	Data Type	Description
Pinjaman	noPanggilanBuku	Text	Nombor panggilan buku memudahkan pencarian dalam sistem.
Buku	namaBuku	Text	Judul buku.
	pengarang	Text	Nama pengarang utama.
	subjek	Text	Kategori pengetahuan buku:- "Am" Or "Komputer" Or "Sains" Or "Matematik" Or "Ekonomi" Or "Sastera" Or "Sejarah" Or "Geografi" Or "Hiburan"

Rajah 2.51 Jadual Buku dalam *Design View*

Sementelahkan tukar jadual Buku kembali kepada Mod “Datasheet View” dan isikan dengan data-data seperti dalam Rajah 2.52

The screenshot shows the Microsoft Access application interface in Datasheet View. The ribbon at the top has 'Table Tools' selected under the 'Datasheet' tab. The main area displays the 'Buku' table data with four columns: 'noPanggilan', 'namaBuku', 'pengarang', and 'subjek'. The data includes four records with sample values.

	noPanggilan	namaBuku	pengarang	subjek	Add New Field
*	IPB123000	Semantic Web	Maryana et. al.	Komputer	
*	IPB125480	C++ Programming	M. A. Baker et.	Komputer	
*	IPB125780	Brain-mind Machinery	G. W. Ng	Komputer	
*	IPB192280	Software Engineering	Roger S.T.	Komputer	

Rajah 2.52 Jadual Buku dalam *Datasheet View*

Ulangi langkah-langkah yang sama untuk mencipta jadual Peminjam. Hasilkan jadual baharu Peminjam. Dalam mod “Design View”, masuk atribut dan jenis data seperti dalam jadual 2.11. Lihat Rajah 2.53. Set idPeminjam dijadikan sebagai kunci primer.

Jadual 2.11 Atribut-atribut dalam jadual Peminjam

Peminjam	
Atribut	Jenis data
idPeminjam	Teks
namaPeminjam	Teks
telefonPeminjam	Teks

Langkah 1: Pilih atribut kunci primer

Langkah 2: Tekan butang ikon Primary Key

Field Name	Data Type	Description
idPeminjam	Text	Nombor peminjam unik. Mesti nombor 6-digit
namaPeminjam	Text	Nama penuh peminjam seperti dalam NRIC.
telefonPeminjam	Text	Nombor telefon bimbit; nombor rumah jika tiada telefon bimbit.

Rajah 2.53 Jadual Peminjam dalam *Design View*

Tukar jadual Peminjam kepada mod “Datasheet View” dan masukkan data. Lihat Rajah 2.54.

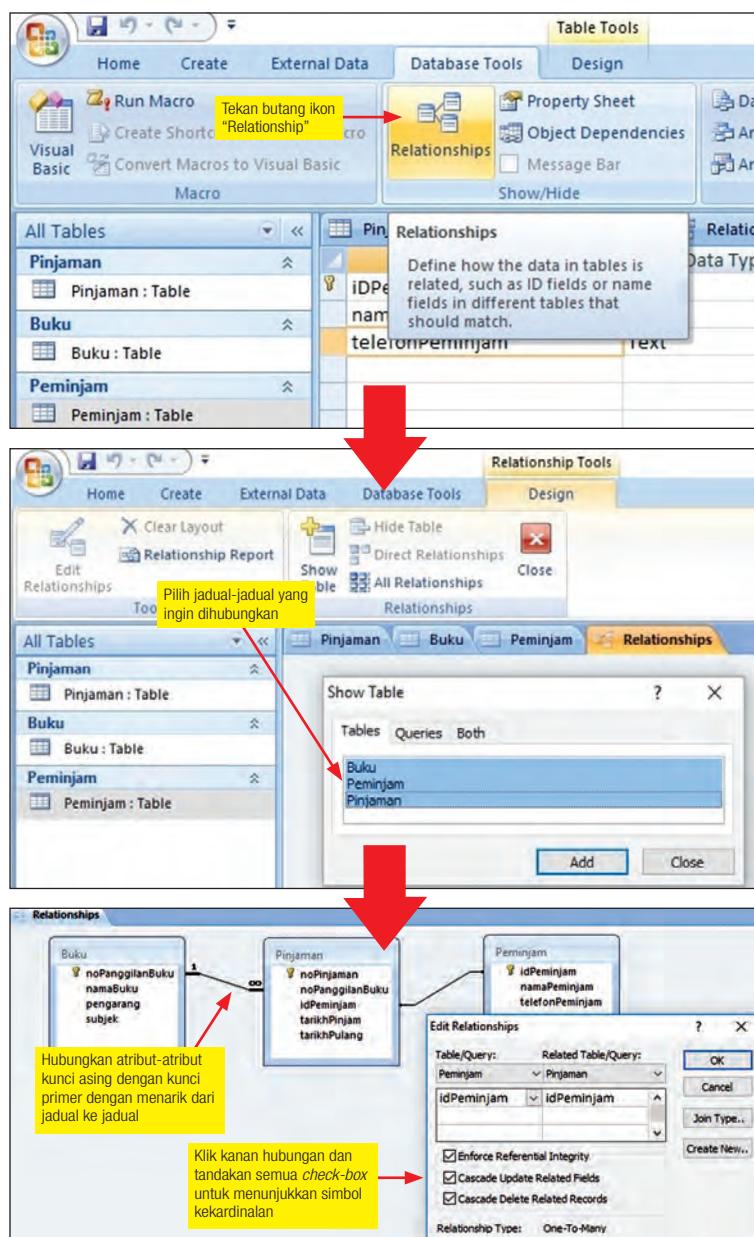
Klik baris baru dan taip data

idPeminjam	namaPeminjam	telefonPem	Add New Field
125001	Azlina Adam	0108899478	
125007	Steve Shiva	0166688888	
125088	Yew Yin Yin	0188888888	
125096	Elvis Tan Kim Hock	0101234567	
125888	Francis Rejang	0129579022	
125099	M. Yusoff Ghazali	0101234563	
*			

Rajah 2.54 Jadual Peminjam dalam *Datasheet View*

6 Membina hubungan antara jadual-jadual

Seterusnya, jadual-jadual dikaitkan melalui hubungan antara atribut. Setiap jadual mempunyai kunci primer. Contohnya untuk menghubungkan jadual A dan jadual B, kunci primer jadual B kp_B akan menjadi satu atribut dalam jadual B. Atribut penghubung ini disebut sebagai kunci asing. Apabila jadual-jadual sebelum itu dirujuk kembali, didapati bahawa terdapat atribut-atribut yang dikongsi oleh dua jadual atau lebih. Biasanya, hubungan ini telah dikenal pasti sewaktu mereka mencipta jadual menggunakan *entity relationship diagram* (ERD). Rujuk Rajah 2.55.



Rajah 2.55 Menghubungkan jadual-jadual (Pinjaman, Buku, Peminjam)

2.3.2 Penciptaan Borang yang Berkaitan dengan Kandungan Jadual



Create a form in Access
goo.gl/6yRsU3

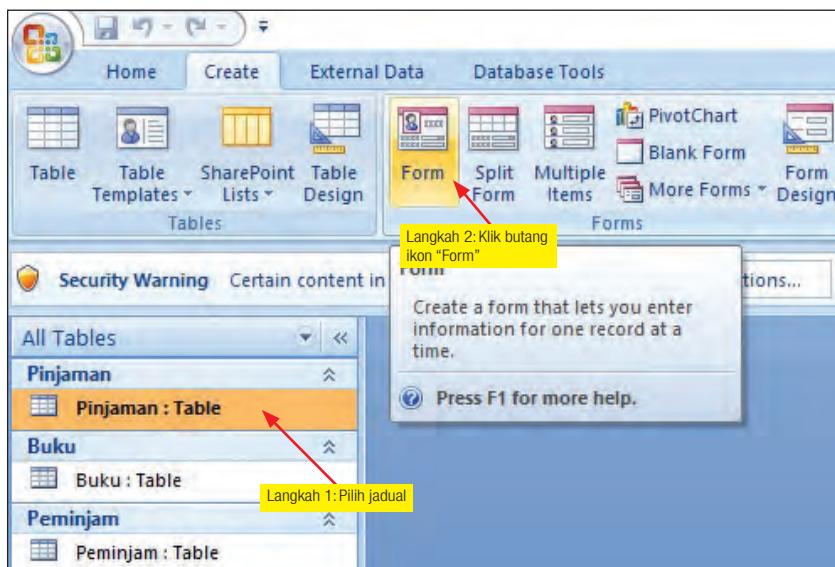
Setelah mempelajari cara membina jadual yang berpandukan skema hubungan menggunakan perisian pangkalan data hubungan, mari kita tinjau cara mencipta borang yang berkaitan dengan kandungan jadual. Borang digunakan dalam pangkalan data untuk:

- a memaparkan data daripada setiap rekod dalam jadual pangkalan data melalui susun atur yang lebih mudah difahami oleh pengguna
- b mengutip data untuk disimpan sebagai rekod dalam susun atur yang lebih mudah difahami oleh pengguna
- c mengesahkan data melalui ciri-ciri borang

1 Mencipta borang

Borang boleh dijana secara automatik menggunakan sistem pangkalan data seperti *MS Access* melalui langkah-langkah yang berikut:

- a Dalam tetingkap “All Tables” di sebelah kiri, pilih jadual Pinjaman.
- b Klik tab “Create” dan ikon “Form”.



A Quick Tutorial on Forms in Microsoft Access 2013
goo.gl/LzNhvJ

Rajah 2.56 Menjana borang daripada jadual Pinjaman

- c Borang Pinjaman dihasilkan daripada atribut-atribut dalam jadual Pinjaman. Namun begitu, nama atribut-atribut dan reka bentuk borang yang dijana sedemikian biasanya kelihatan ringkas dan kurang bermaklumat.



PinjamanBuku : Database (Access 2007) - Microsoft Access

Format

noPinjaman: 1
noPanggilanBuku: IPB123000
IDPeminjam: 125007
tarikhPinjam: 7/23/2016
tarikhPulang: 8/22/2016

Rajah 2.57 Borang Pinjaman yang ringkas

- d Di bawah tab “Format”, ubah fon, saiz teks, susunan teks, warna latar dan logo supaya kelihatan lebih menarik.

Format

Langkah 1: Klik tab "Format"

Langkah 2: Pilih format yang sesuai

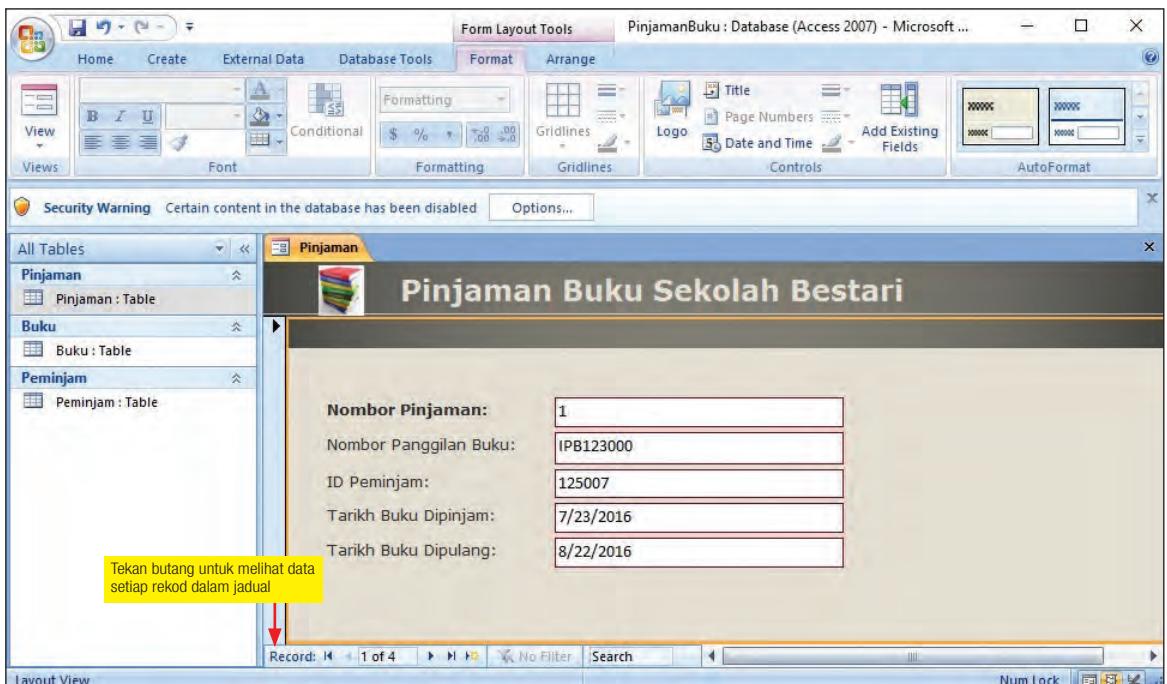
Pinjaman Buku Sekolah Best

Nombor Pinjaman: 1
Nombor Panggilan Buku: IPB123000
ID Peminjam: 125007
Tarikh Buku Dipinjam: 7/23/2016
Tarikh Buku Dipulang: 8/22/2016

Rajah 2.58 Borang Pinjaman yang telah digubal

- e “Save” borang yang telah diubahsuaikan dan berikan nama yang sesuai, misalnya BorangPinjaman. Dalam panel “All Tables”, akan kelihatan ikon BorangPinjaman.

- f** Pada tab BorangPinjaman, klik kanan dan pilih “Close” untuk menutup borang.
- g** Untuk membuka paparan tersebut kembali, kembalilah ke panel “All Tables”, pilih BorangPinjaman dan klik kanan ikon borang Pinjaman. Setelah itu, daripada menu “drop-down”, pilih “Open”. Borang yang telah ditutup tadi akan muncul semula dan boleh digunakan untuk memaparkan data-data dalam jadual Pinjaman. Untuk menerokai setiap rekod dalam jadual, tekan butang “Next record” atau “Previous record” pada baris bawah borang tersebut.



Rajah 2.59 Melihat data-data jadual melalui Borang Pinjaman

Aktiviti **40** Penciptaan borang berkaitan dengan jadual

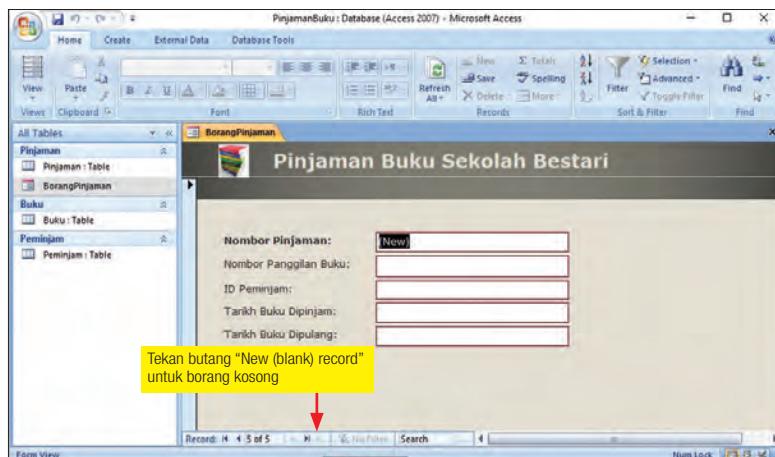
1 (a) Ulangi langkah-langkah yang telah diajar untuk menghasilkan borang untuk jadual Buku secara automatik.
 (b) Gunakan Layout View untuk mengubah reka bentuk borang.

2 (c) Ulangi langkah-langkah yang telah diajar untuk menghasilkan borang untuk jadual Peminjam secara automatik.
 (d) Gunakan Layout View untuk mengubah reka bentuk borang.

2.3.3 Pemasukan Data dalam Jadual Melalui Borang

Selain menerokai rekod-rekod dalam jadual, borang Pinjaman juga boleh digunakan untuk menambahkan rekod baharu. Langkah-langkah untuk menambahkan data baharu adalah seperti yang berikut:

- Pergi ke hujung rekod dalam jadual dengan menekan butang “Last record” pada bahagian baris bawah borang.
- Tekan butang “New (blank) record”.



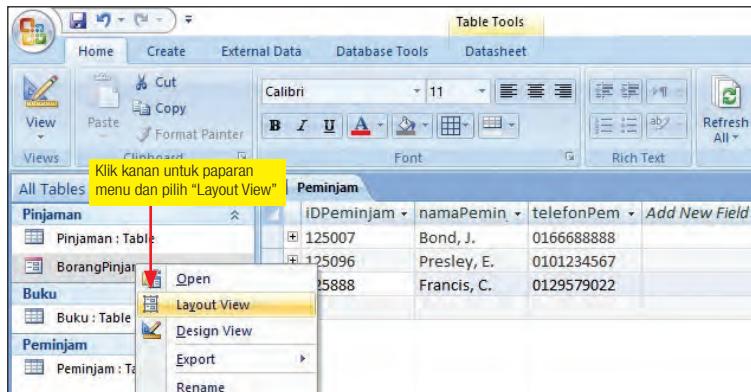
Rajah 2.60 Menambahkan rekod baharu melalui Borang Pinjaman

- Selepas itu, tambah data untuk rekod baharu tersebut.
- Simpan data dengan klik kanan tab BorangPinjaman dan pilih “Save”.
- Ulangi klik kanan dan pilih “Close” untuk menutup borang tersebut.



Rajah 2.61 Menambahkan data dalam rekod baharu melalui Borang Pinjaman

- f** Borang Pinjaman yang dibuka untuk melihat data tidak dapat digubal. Sekiranya masih ingin menggubal borang Pinjaman, klik kanan ikon borang Pinjaman dan pilih “Layout View”.



Rajah 2.62 Membuka Borang Pinjaman semula untuk digubal

Selain contoh borang yang telah digunakan, jenis-jenis borang lain ialah “Split Form” dan “Multiple Items”. Pembangun pangkalan data juga boleh menggunakan “Form Design” untuk mereka bentuk borang yang diingini.

Jadual 2.12 Jenis-jenis borang

Jenis borang	Deskripsi
<i>Split form</i>	Mempamerkan kedua-dua borang dan jadual dalam “Design view”. Lihat Rajah 2.63: Borang “Split form”.
<i>Multiple items</i>	Mempamerkan beberapa rekod jadual dalam satu borang. Lihat Rajah 2.64: Borang “Multiple items”.
<i>Form design</i>	Membolehkan borang direka cipta khusus dengan menentukan rekod-rekod yang ingin dipamerkan daripada satu atau lebih jadual dalam pangkalan data. Lihat Rajah 2.65: Borang “Form design”.

Rajah 2.63 Borang “Split form”

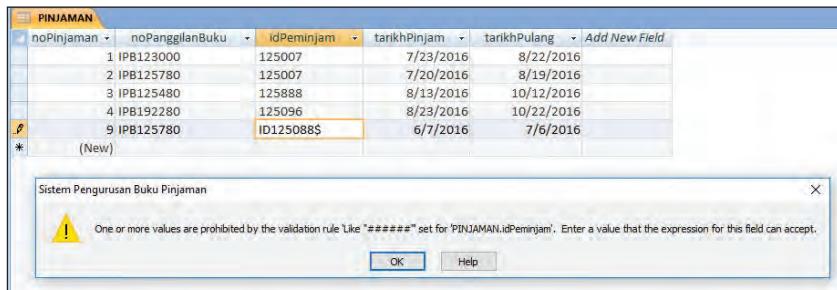
BorangPinjaman

noPinjaman	noPanggilanBuku	iDPeminjam	tarikhPinjam	tarikhPulang
1	IPB123000	125007	7/23/2016	8/22/2016
2	IPB125780	125007	7/20/2016	8/19/2016
3	IPB125480	125888	8/13/2016	10/12/2016
4	IPB192280	125096	8/23/2016	10/22/2016
5	Learn Scalar in 30 Days	125888	7/23/2016	8/22/2016
6	Effective Presentations	125888	7/23/2016	9/22/2016
(New)				

Rajah 2.64 Borang "Multiple items"

Rajah 2.65 Borang "Form design"

Pengesahan data diperlukan untuk memastikan data yang dimasukkan adalah dalam format yang sesuai dan berada dalam julat yang logikal. Hal ini adalah penting untuk integriti data.



Rajah 2.65 Pengesahan input yang rambang

Dalam rajah di atas, kelihatan nilai bagi noPinjaman 9, pengguna memasukkan idPeminjam yang tidak menepati format 6-digit bagi nombor yang ditetapkan. Kesilapan memasukkan data seperti contoh di atas dapat diatasi atau dikurangkan dengan menggunakan pengesahan data sewaktu kemasukan dilakukan.

Pengesahan dibuat menggunakan ungkapan yang dimasukkan oleh pengguna ke dalam *Access*. Pengesahan boleh dibuat bagi setiap atribut, dengan menggunakan *Validation Rule*. Ungkapan *Validation Rule* ialah ungkapan *boolean*. Hasil ungkapan yang digunakan adalah *True* atau *False* sahaja. Sekiranya ungkapan menilai input pengguna kepada *False*, *Access* tidak akan menyimpan input pengguna sebagai rekod tetapi akan prom pengguna supaya memasukkan nilai baru.

Pengesahan bagi setiap medan boleh dibuat dengan menggunakan *Validation Rule*. Terdapat dua jenis *Validation Rule* iaitu *Validation Rule* untuk atribut dan *Validation Rule* untuk rekod.

Validation Rule untuk atribut

- *Validation Rule* untuk atribut memastikan data yang dimasukkan dalam atribut mematuhi kriteria data input.

Validation Rule untuk rekod

- *Validation Rule* untuk rekod merujuk atribut lain dalam jadual yang sama sewaktu membuat pengesahan. Ini melibatkan perbandingan dua atau lebih atribut dalam jadual yang sama.

2.3.4 Penghasilan *Query* untuk Mendapatkan Maklumat Semula

Pangkalan data mengandungi jumlah rekod data yang sangat banyak, iaitu daripada puluhan ke ribuan baris. Hal ini demikian kerana, data sentiasa dikumpulkan daripada pelbagai aktiviti yang memerlukan penyimpanan data. Data yang disimpan mungkin terdiri daripada data lama yang dikutip beberapa tahun terdahulu. Hal ini menyebabkan kesukaran membuat carian. Kebanyakan carian memerlukan data tertentu sahaja disebabkan oleh tugas yang perlu dilaksanakan oleh pengguna.

Contoh

Pangkalan data perpustakaan mengandungi rekod buku, peminjam dan pinjaman untuk semua murid sekolah. Seorang pustakawan ditugaskan untuk menghasilkan:



Uji Minda

Apakah yang terjadi sekiranya *query* tidak dicipta?
Adakah kita mempunyai cara yang lain untuk mencari data yang lama?

a carian yang memaparkan ID peminjam, nama peminjam, nama buku, tarikh pinjam dan tarikh pulang untuk semua pinjaman.

b carian yang memaparkan ID peminjam, nama peminjam, nama buku, tarikh pinjam dan tarikh pulang untuk pinjaman lewat sahaja.

Untuk menyelesaikan tugas ini, gunakan objek *query*. *Query* ialah objek Microsoft Access yang digunakan untuk membuat carian dalam pangkalan data. *Query* merentasi satu jadual dan menghadkan data yang dicari.

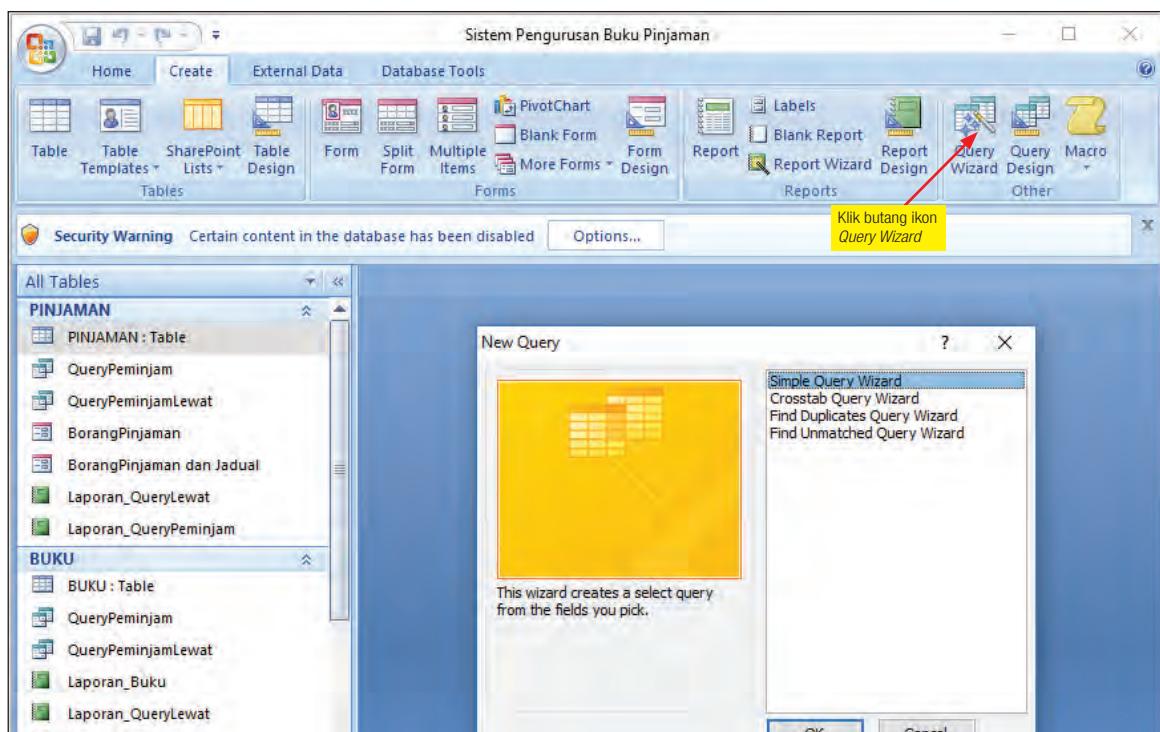
- *Query* boleh dibuat untuk satu atau lebih daripada satu jadual. Contohnya, pustakawan itu perlu menyenaraikan nombor panggilan buku, nama buku, tarikh pulangan dan nama peminjam daripada *query* yang melibatkan gabungan jadual-jadual pinjaman, peminjam dan buku.
- *Query* boleh menggunakan kriteria carian tertentu seperti tarikh, lokasi, individu atau ciri-ciri tertentu.

Untuk menghasilkan *query* dalam Access, dua kaedah boleh digunakan iaitu menggunakan *Query Wizard* ataupun menggunakan *Query Design*.

Menggunakan *Query Wizard*

Query Wizard membolehkan *query* dihasilkan berpanduan arahan dalam tetingkap dialog Microsoft Access.

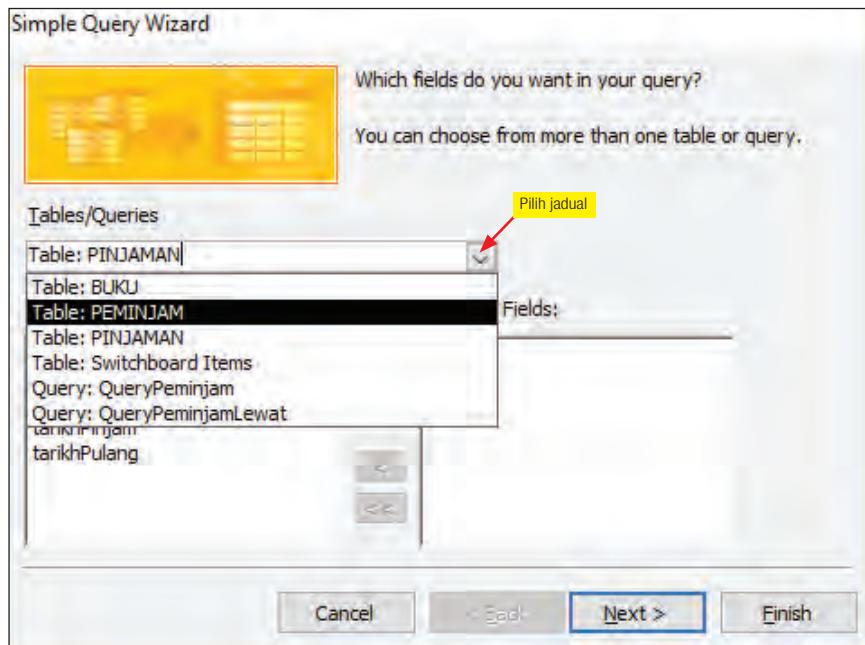
Langkah 1 : Daripada tab “Create”, klik ikon “*Query Wizard*” untuk memulakan proses bagi menghasilkan *query*. Lihat Rajah 2.66.



Rajah 2.66 Menggunakan *Query Wizard* untuk menghasilkan *query*

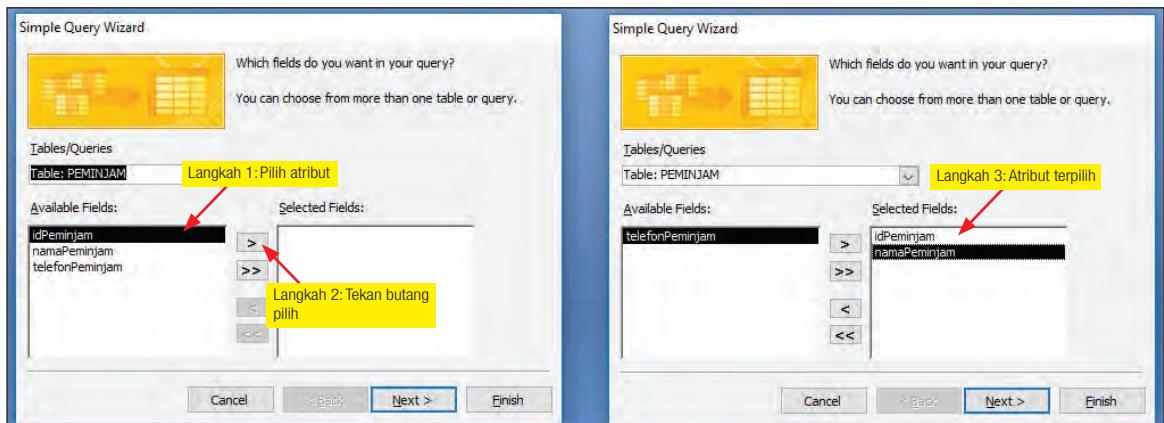


Langkah 2 : Pilih jadual yang terlibat. Lihat Rajah 2.67.



Rajah 2.67 Pilih jadual

Langkah 3 : Pilih atribut-atribut dalam jadual terlibat. Lihat Rajah 2.68.



Rajah 2.68 Memilih atribut-atribut tertentu daripada jadual Peminjam

Langkah 4 : Ulangi langkah-langkah 1 hingga 3 untuk jadual-jadual seperti yang berikut:

Jadual 2.13 Pilihan atribut untuk jadual BUKU dan PINJAMAN

Jadual	Atribut-atribut yang dipilih
BUKU	namaBuku
PINJAMAN	tarikhPinjam, tarikhPulang

Langkah 5 : Simpan kombinasi atribut-atribut sebagai *query*.

Setelah semua atribut yang diperlukan telah dikenal pasti dan ditambah kepada *query*, simpan *query* tersebut dengan nama yang mudah difahami tentang kegunaannya. Sebagai contoh, *query* boleh disimpan menggunakan nama QueryPinjaman.

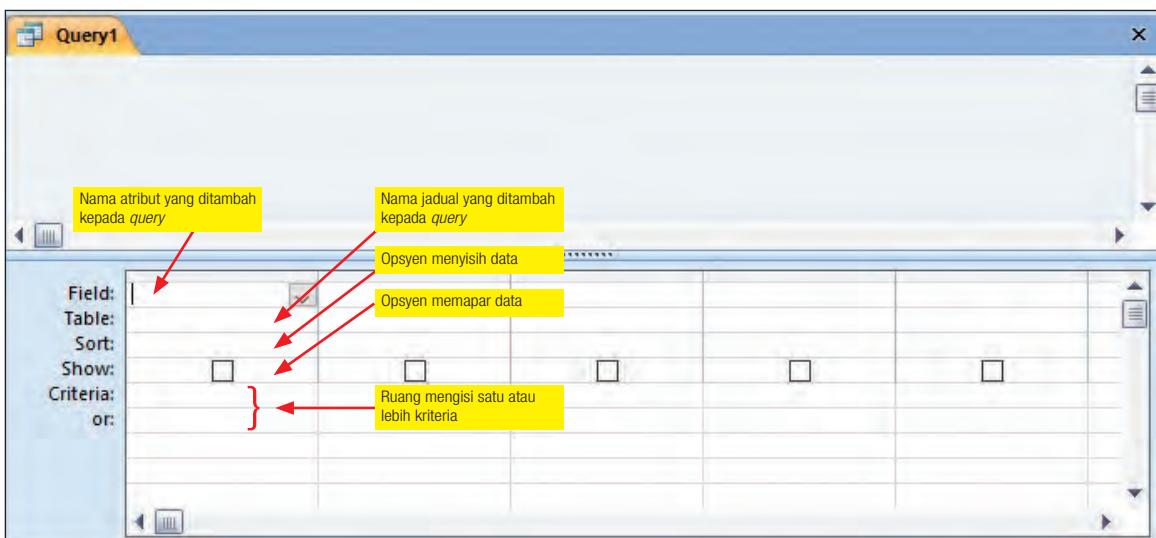
The screenshot shows the Microsoft Access interface with the title bar 'Sistem Pengurusan Buku Pinjaman'. The ribbon has tabs for Home, Create, External Data, Database Tools, and Reports. Under the Create tab, there are buttons for Table, Form, Report, and Query. A security warning message is displayed: 'Security Warning Certain content in the database has been disabled' and 'Jadual-jadual asal tidak mempunyai kombinasi atribut-atribut sedemikian'. The left pane shows 'All Tables' with 'PINJAMAN' selected, and the 'Tables' section of the ribbon. The main area displays a query result for 'QueryPinjaman' with the following data:

IDPinjam	namaPeminjam	namaBuku	tarikhPinjam	tarikhPulang
125007	Steve Shiva	Semantic Web	7/23/2016	8/22/2016
125007	Steve Shiva	Brain-mind Machinery	7/20/2016	8/19/2016
125888	Francis Rejang	C++ Programming	8/13/2016	9/12/2016
125096	Elvis Tan Kim Hock	Software Engineering	8/23/2016	9/22/2016
125088	Jeniffer Yeo Yin Yin	Brain-mind Machinery	6/7/2016	7/6/2016
125001	Azrina Adam	Pengaturcaraan C	9/29/2016	10/28/2016
125001	Azrina Adam	Computer Science for Matriculatio	9/29/2016	10/28/2016

Rajah 2.69 *Query* butiran pinjaman daripada atribut-atribut terpilih

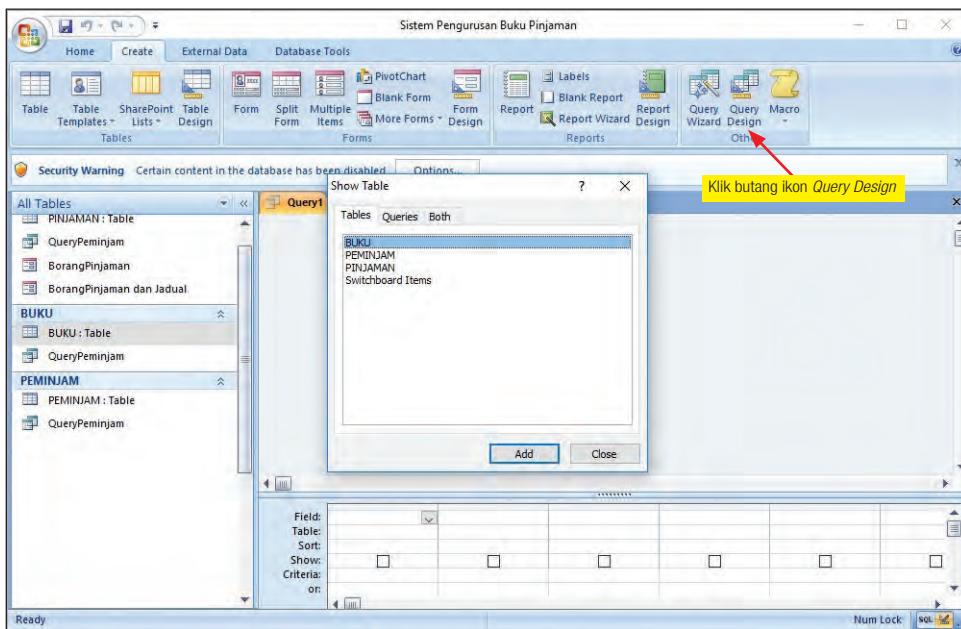
Menggunakan *Query Design*

Query Design membolehkan *query* dihasilkan dengan menggabungkan jadual-jadual dan memilih atribut-atribut yang ingin dipamerkan dalam panel atribut *query*.



Rajah 2.70 Tetingkap *Query Design* yang kosong

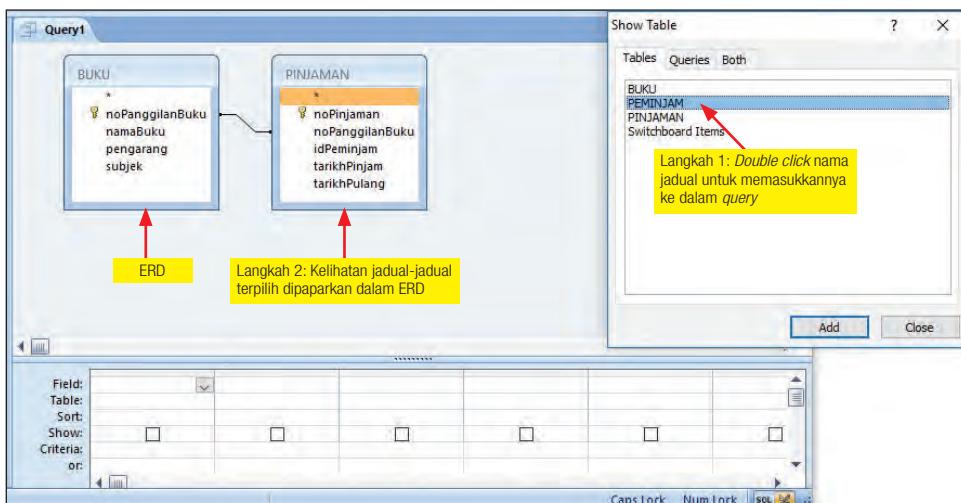
Langkah 1 : Daripada tab “Create”, klik ikon “Query Design” untuk memulakan proses menghasilkan query. Lihat Rajah 2.71.



Rajah 2.71 Melancarkan Query Design untuk membina Query yang baru

Langkah 2 : Pilih jadual yang terlibat. Lihat Rajah 2.72.

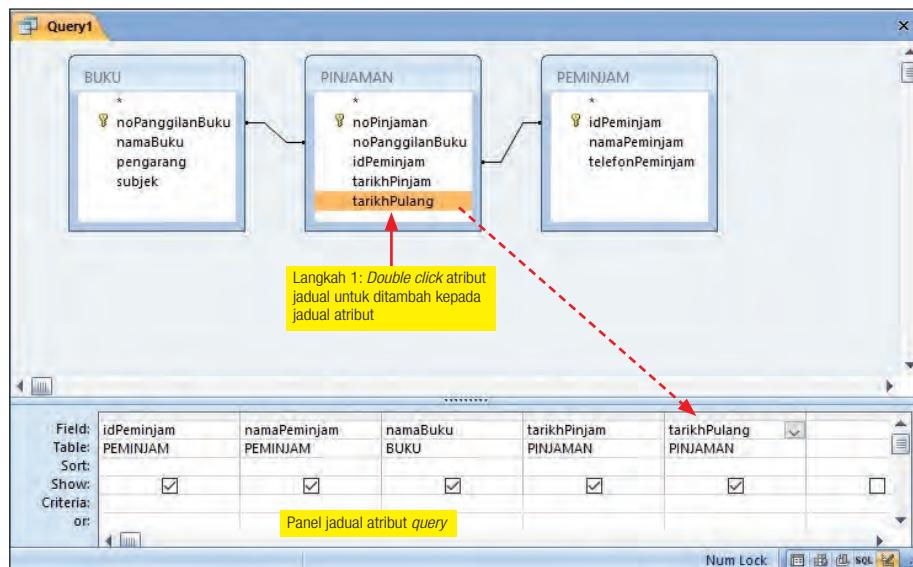
Ini dilakukan dengan *double click* nama jadual dalam dialog *Show Table*. Dalam contoh ini, klik jadual BUKU diikuti oleh PINJAMAN dan akhir sekali, PEMINJAM. Ini akan memaparkan ERD *Access* secara automatik. Dalam ERD *Access*, semua atribut dan kunci primer akan kelihatan.



Rajah 2.72 Memilih jadual untuk menghasilkan query

Langkah 3 : Pilih atribut-atribut dalam jadual terlibat. Lihat Rajah 2.73.

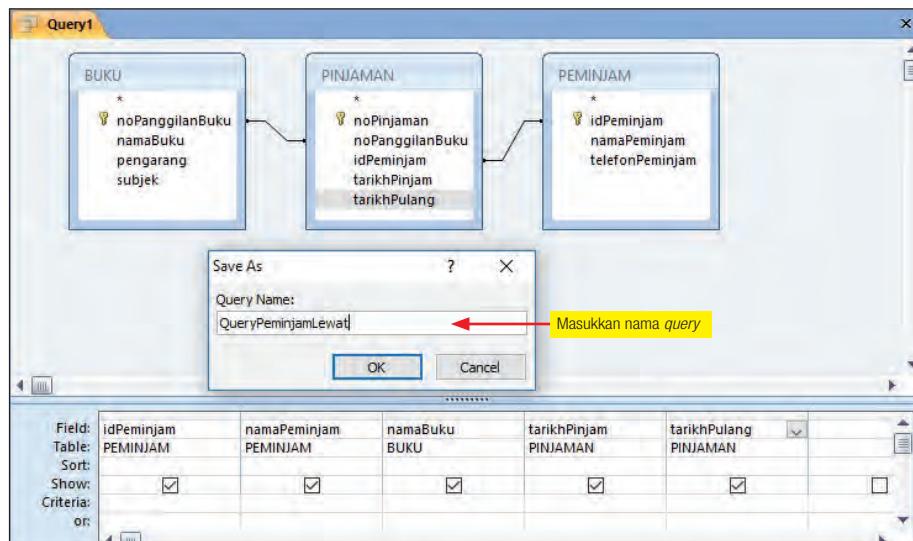
Pilih atribut dalam jadual ERD untuk dimasukkan ke dalam *query* yang sedang dibina. Ini dilakukan dengan membuat *double click* ke atas atribut-atribut pilihan daripada kotak ERD jadual yang berkaitan. Ini akan menambahkan atribut jadual tersebut sebagai atribut *query* dalam jadual pada panel seperti yang ditunjukkan dalam rajah di bawah.



Rajah 2.73 Memilih atribut jadual sebagai atribut *query*

Langkah 4 : Simpan kombinasi atribut-atribut sebagai *query*. Lihat Rajah 2.74.

Klik butang ikon “Save” dan simpan *query* dengan nama “QueryPeminjamLewat”. Dalam pembelajaran seterusnya, kita akan mempelajari kaedah mengeset kriteria dalam jadual *query*.



Rajah 2.74 Menyimpan *query* senarai peminjam lewat

Penggunaan kriteria

Query boleh memaparkan rekod-rekod berdasarkan kriteria tertentu. Kriteria ialah nilai khusus bagi sesuatu atribut yang dijadikan sebagai syarat carian. Dalam panel atribut query yang sama, kriteria boleh ditetapkan untuk menghadkan paparan kepada rekod-rekod yang memenuhi kriteria sahaja.



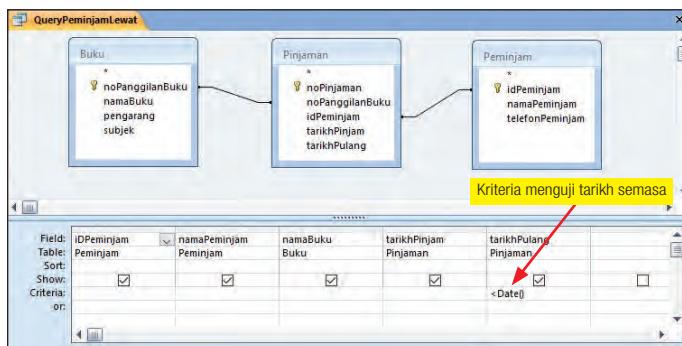
Queries and SQL
goo.gl/SFsvYM

Contoh 6

Hasilkan query pinjaman lewat yang sama dengan query pinjaman tetapi memaparkan pinjaman yang sudah lewat sahaja. Rekod mestilah mengandungi idPeminjam, namaPeminjam, namaBuku, tarikhPinjam dan tarikhPulang.

Penyelesaian:

- 1 Ubah query QueryPeminjamLewat kepada mod Design View.
- 2 Perhatikan bahawa kriteria carian cuma satu, iaitu buku lewat. Kriteria buku lewat adalah tarikhPulang yang mestilah lebih awal daripada tarikh carian.
 - (a) Dalam panel jadual atribut query, perhatikan lajur tarikhPulang
 - (b) Set kriteria supaya memaparkan rekod-rekod dengan tarikhPulang <Date()>.



Memasukkan kriteria ke dalam query

- 3 Uji dengan menekan ikon “Run” dalam tab “Design”. Perhatikan kemunculan keputusan query. Kali ini, cuma maklumat peminjam lewat sahaja dipaparkan.

Hasil ujian query menunjukkan pinjaman yang lewat pada tarikh sistem

Diberikan jadual BUKU yang mengandungi data berikut.

BUKU		namaBuku	pengarang	subjek
	noPanggilan			
[+]	IPB111000	The Curious Kid's Science Book	Asia Citro	Sains
[+]	IPB111100	100 Greatest Science Inventions of All Time	Kendall H.	Sains
[+]	IPB115000	PMR 2003-2005 Matematik	Nawai Z.	Matematik
[+]	IPB115100	Fokus Matematik Tambahan	Wong M.	Matematik
[+]	IPB121000	Computer Science for Matriculation	Yew K. H.	Komputer
[+]	IPB123000	Semantic Web	Maryana et. al.	Komputer
[+]	IPB125480	C++ Programming	M. A. Baker et. al.	Komputer
[+]	IPB125490	Pengaturcaraan C	Marini et. Al.	Komputer
[+]	IPB125780	Brain-mind Machinery	G. W. Ng	Komputer
[+]	IPB192280	Software Engineering	Roger S.T.	Komputer

Jadual BUKU di atas menunjukkan senarai buku di perpustakaan sekolah. Hasilkann *query-query* yang memaparkan buku-buku daripada kategori subjek tertentu sahaja.

- (a) Memulangkan rekod-rekod buku subjek "Komputer" sahaja.
- (b) Memulangkan rekod-rekod buku untuk kedua-dua subjek "Sains" dan "Matematik".

Nota: Carian boleh dibuat menggunakan dua atau lebih kriteria.

2.3.5 Penjanaan Laporan Berdasarkan Hasil Query

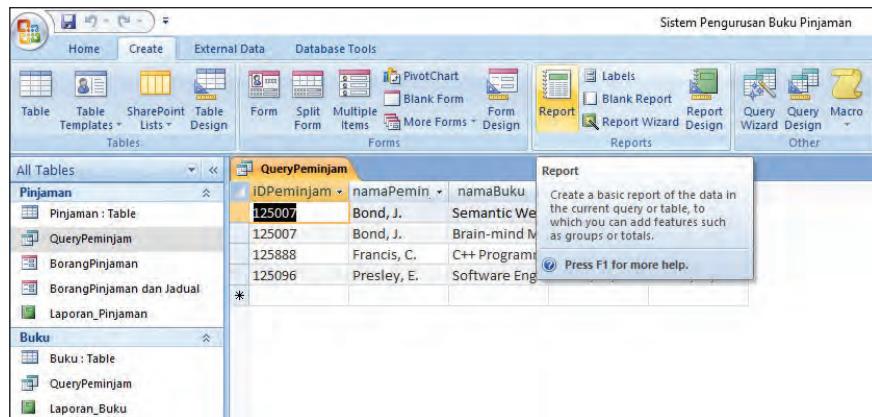
Query boleh digunakan untuk menjana laporan. Laporan menerbitkan data dalam format yang lebih mudah difahami supaya dapat,

- a** menghasilkan laporan untuk mesyuarat.
- b** membantu organisasi membuat keputusan.
- c** membantu dalam penerbitan.

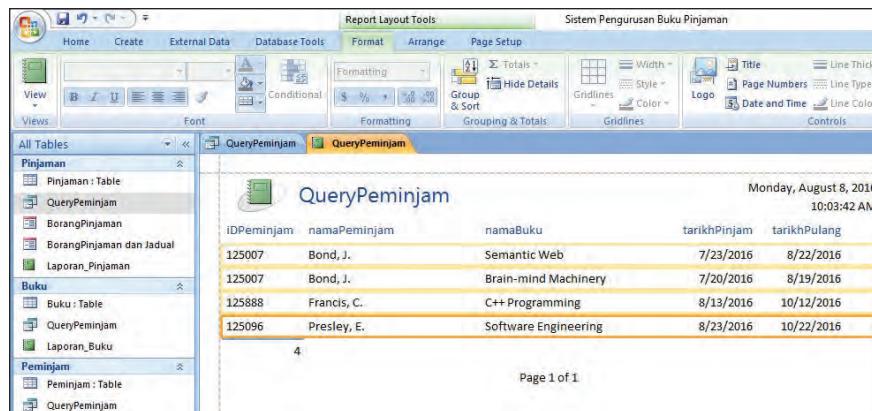
Langkah-langkah untuk menjana laporan daripada *query* adalah seperti yang berikut:

- a** Klik *query* yang ingin digunakan pada panel "All Tables".
- b** Daripada bar menu, pilih tab "Create", klik ikon "Report".
- c** Laporan akan dijana secara automatik.
- d** Jika perlu, persembahan laporan ini boleh diubah secara manual dalam *Design View*.

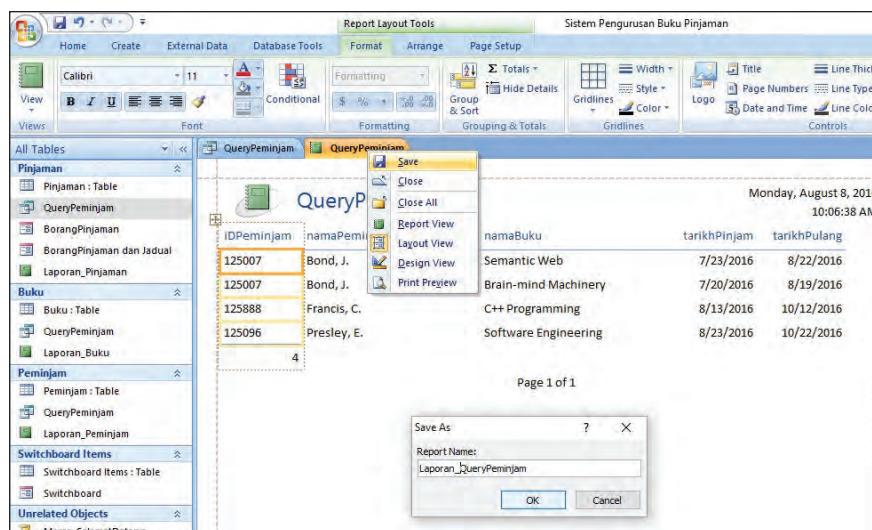




Rajah 2.76 Memulakan penjanaan laporan secara automatik

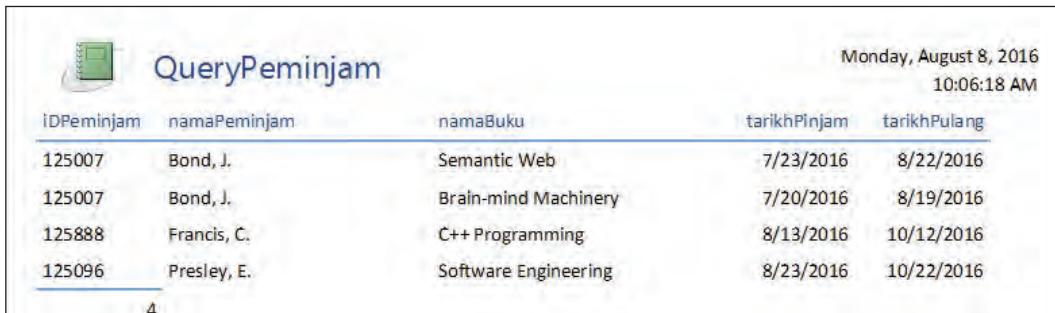


Rajah 2.77 Format laporan boleh diedit



Rajah 2.78 Menyimpan laporan

Laporan yang dihasilkan boleh dicetak. Untuk melihat previu cetakan, pilih tab “Home” dan klik ikon “View” diikuti “Print Preview”. Untuk mencetak, klik ikon “Print”.



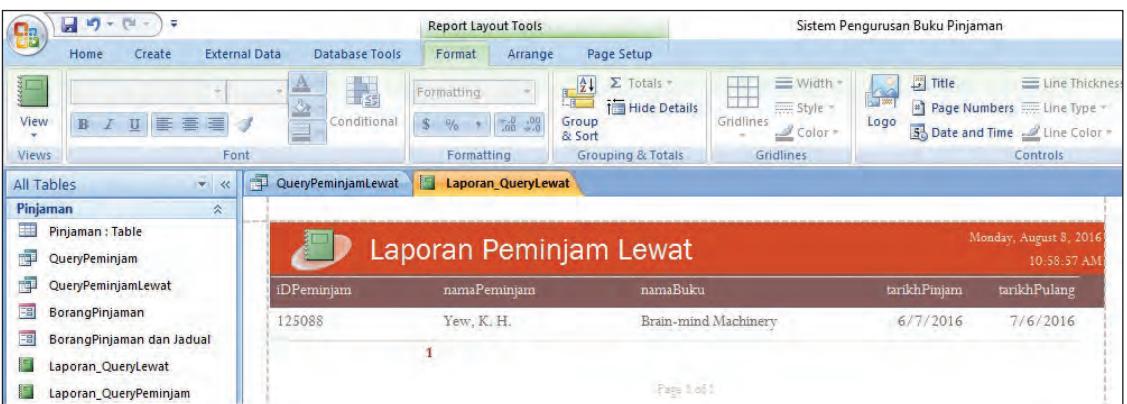
The screenshot shows a Microsoft Access query results window titled "QueryPeminjam". The results are as follows:

IDPeminjam	namaPeminjam	namaBuku	tarikhPinjam	tarikhPulang
125007	Bond, J.	Semantic Web	7/23/2016	8/22/2016
125007	Bond, J.	Brain-mind Machinery	7/20/2016	8/19/2016
125888	Francis, C.	C++ Programming	8/13/2016	10/12/2016
125096	Presley, E.	Software Engineering	8/23/2016	10/22/2016

Rajah 2.79 Melihat previu cetakan

Aktiviti 42 Query Pinjaman Lewat

Hasilkann laporan bagi *query* pinjaman lewat. Pastikan *query* tersebut memaparkan pinjaman yang lewat sahaja seperti yang ditunjukkan dalam contoh di bawah. Sahkan dengan menganalisis data-data asal dalam jadual.



The screenshot shows the Microsoft Access ribbon with the "Report Layout Tools" tab selected. A preview of the report "Laporan_QueryLewat" is displayed, showing the following data:

iDPeminjam	namaPeminjam	namaBuku	tarikhPinjam	tarikhPulang
125088	Yew, K. H.	Brain-mind Machinery	6/7/2016	7/6/2016

2.3.6 Sistem Maklumat Melalui Makro dan Switchboard

Mencipta dan menggunakan *macro*

Makro ialah siri arahan *Microsoft Access* yang dilaksanakan secara automatik apabila suatu data dimasukkan atau butang ditekan. Makro memudahkan automasi pangkalan data dan interaksi dengan pengguna. Makro boleh terkandung dalam objek makro ataupun dimasukkan ke dalam komponen borang atau laporan *Access*.

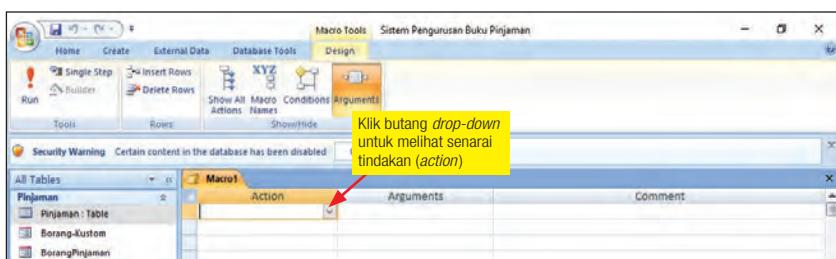
Makro baharu boleh dibina sekiranya makro-makro dalam senarai tindakan tidak menepati kehendak atau citara pembangun pangkalan data. Ikuti langkah-langkah berikut untuk menghasilkan makro yang baharu:

- 1 Di bawah menu bar, klik tab "Create" dan ikon "Macro".



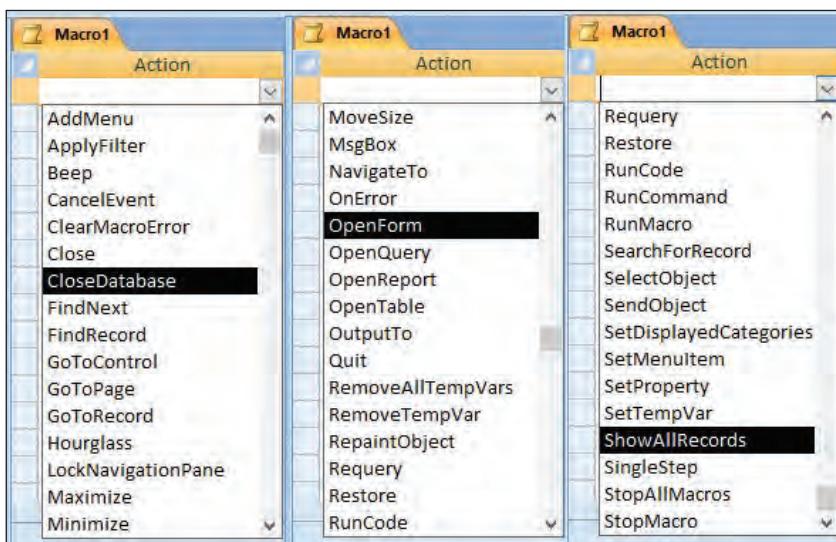
Rajah 2.80 Memasukkan objek makro ke dalam pangkalan data

- 2 Ini akan melancarkan *Macro Builder*. *Macro Builder* mengandungi jadual kosong untuk menempatkan makro-makro yang bakal digunakan oleh aplikasi. Lihat Rajah 2.81.



Rajah 2.81 Menggunakan *Macro Builder* dalam projek pangkalan data

- 3 Dalam pembina makro, klik sel kosong yang pertama dalam lajur tindakan. Klik anak panah untuk memaparkan senarai tindakan yang tersedia, kemudian pilih tindakan yang anda ingin gunakan.



Rajah 2.82 Senarai tindakan (action) dalam makro Access

Tahukah Anda?

Antara muka *switchboard* akan muncul secara automatik selepas anda membina pangkalan data.



The Switchboard
goo.gl/1zpHqB

- 4 Dalam tetingkap pembina makro, terdapat pelbagai tindakan makro *Action* yang boleh digunakan. Sebagai contoh, pengguna boleh menggunakan makro untuk menghasilkan bunyi “beep”, menutup pangkalan data, mencari rekod, membuka borang, membuka laporan, membuka *query*, membuka jadual dan pelbagai lagi. Lihat senarai panjang dalam rajah berikut.
- 5 Di bawah lajur *Arguments*, pengguna boleh memasukkan argumen untuk setiap tindakan makro *Action* yang terpilih. Tidak semua *Action* memerlukan argumen. Lajur *Comment* membolehkan nota berkeraan *Action* dicoretkan untuk rujukan pembangun pangkalan data.
- 6 Makro boleh diuji menggunakan perintah *Run* pada tab *Design*.

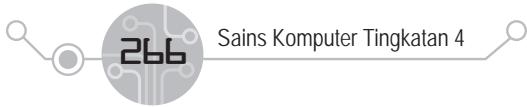
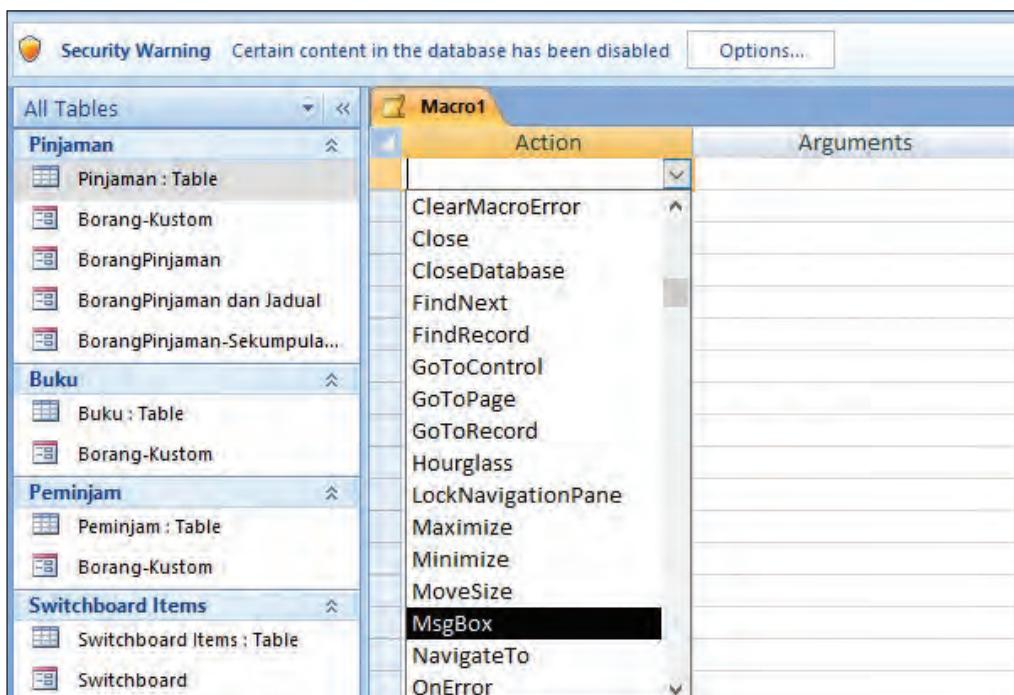
Untuk memahami pembinaan makro baharu dengan lebih jelas lagi, lihat contoh-contoh yang berikut:

Contoh (7)

Hasilkan makro mudah yang akan mengucap “Selamat Datang” kepada pengguna melalui *Message Box*.

Penyelesaian:

- 1 Di bawah lajur *Action*, pilih tindakan “*MsgBox*”.
- 2 Di bawah lajur *Arguments*, isi nilai-nilai *argument* dalam panel *Action Arguments*.



The screenshot shows the Microsoft Access 'Macro Tools' ribbon tab selected. In the left pane, under 'All Tables', there are several tables listed. In the center pane, a table named 'Macro_SelamatDatang' is displayed with one row. The row contains the macro name 'Msg_SelamatData', the action 'MsgBox', and arguments 'Selamat datang., Yes, Informat'. A comment 'Dipaparkan sewaktu pelancaran pangkalan data.' is also present. Below the table, the 'Action Arguments' section is visible, showing message content 'Selamat datang.', beep status 'Yes', type 'Information', and title 'Sistem Pinjaman Buku Sekolah Bestari'. A note 'Enter a comment in this column.' is shown in the bottom right.

Mengisi nilai-nilai *argument* dalam panel *Action Arguments*

This screenshot shows the same Microsoft Access interface as the previous one, but with a context menu open over the macro entry in the list. The 'Run' option is highlighted with a yellow arrow. The menu also includes 'Design View', 'Export', 'Rename', 'Hide in This Group', 'Delete', 'Cut', 'Copy', 'Paste', and 'Object Properties'. A note at the bottom left says 'Klik kanan makro dan pilih Run'. In the bottom right, a message box titled 'Sistem Pinjaman Buku Sekolah Bestari' displays the message 'Selamat datang.' with an 'OK' button. A yellow box labeled 'Output berhasil' with an arrow points to the message box.

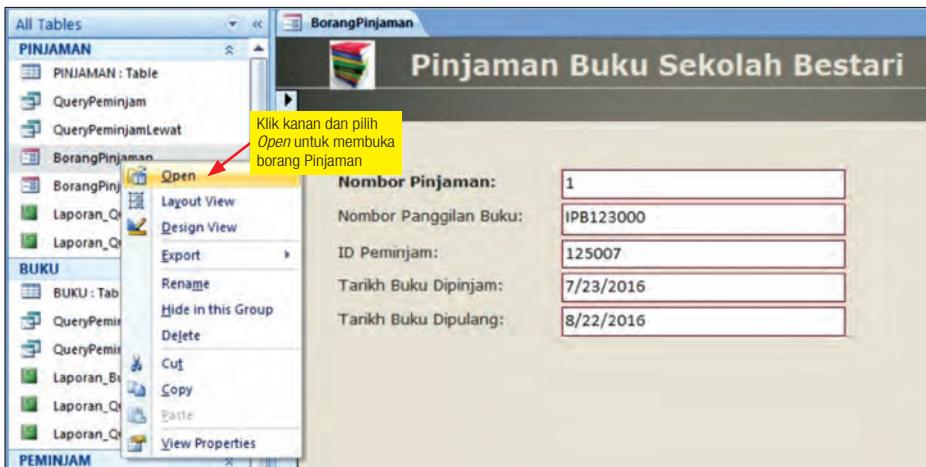
Menguji makro baru

CONTOH (8)

Rajah di bawah menunjukkan borang pinjaman buku Sekolah Bestari dalam mod data *Edit*. Hasilkan objek makro yang secara automatik menjadikan borang pinjaman buku kosong untuk menambah pinjaman baharu. Mod borang ini adalah *Add*.

Nota: Borang (Form) mempunyai tiga mod data (*Data Mode*) iaitu *Read-Only*, *Edit* ataupun *Add* yang boleh diset dalam panel *Action Arguments* sewaktu membina makro.

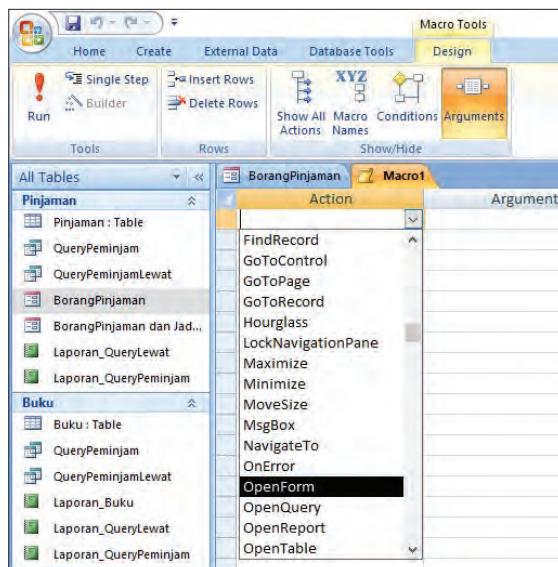
Penyelesaian



Membuka borang Pinjaman dalam mod *Edit*

Langkah-langkah:

- 1 Dari tab *Create*, klik "Macro".
- 2 Dari senarai *Arguments*, pilih "OpenForm" di bawah *Action*.



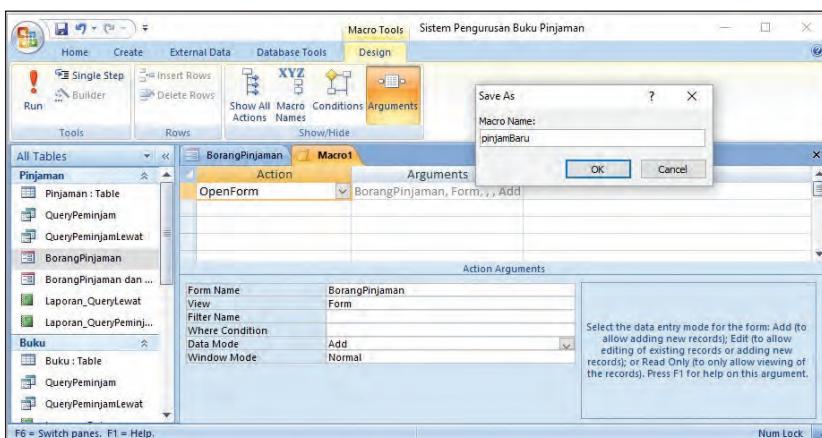
Memilih arahan Access



- 3 Di dalam tetingkap Action Arguments, isikan atribut-atribut argumen yang kosong.

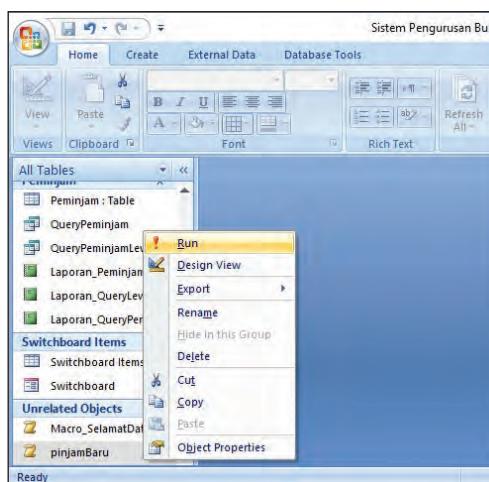
Jenis argumen arahan	Nilai yang diisi
Form name	Nama borang terpilih dalam Access.
View	Mod untuk melihat borang: <i>Design, Datasheet, Print Preview, Layout, Pivotal Chart, Pivotal Layout</i> ataupun <i>Form</i> . Pilih <i>Form</i> untuk melihat borang.
Data Mode	Sama ada dalam bentuk <i>Read-Only</i> , <i>Edit</i> ataupun <i>Add</i> . Pilih <i>Add</i> supaya menunjukkan borang baharu yang kosong.
Window Mode	Normal

- 4 Simpan makro tersebut sebagai pinjamBaru. Ini dilakukan dengan mengklik ikon *Run* di bawah tab *Design* dan suatu *dialog box* muncul untuk menamakan makro baharu tersebut. Objek makro PinjamBaru akan muncul dalam panel *All Tables* di penjuru bawah sebelah kiri Access.



Menamakan dan menyimpan makro baharu sebagai pinjamBaru

- 5 Uji makro pinjamBaru. Klik kanan objek pinjamBaru dan pilih item “Run” daripada menu *pop-up*.



Menguji objek makro pinjamBaru



Borang pinjaman buku kosong dalam mod Add

CONTOH 9

Apabila borang pinjaman buku Sekolah Bestari berada dalam mod data Add, perhatikan bahawa kursor komputer berada dalam kotak teks pertama iaitu Nombor Pinjaman. Lihat rajah yang ditunjukkan di bawah. Hasilkan objek makro yang secara automatik memindahkan kursor daripada kotak teks Nombor Pinjam kepada kotak teks ID Peminjam.

Penyelesaian

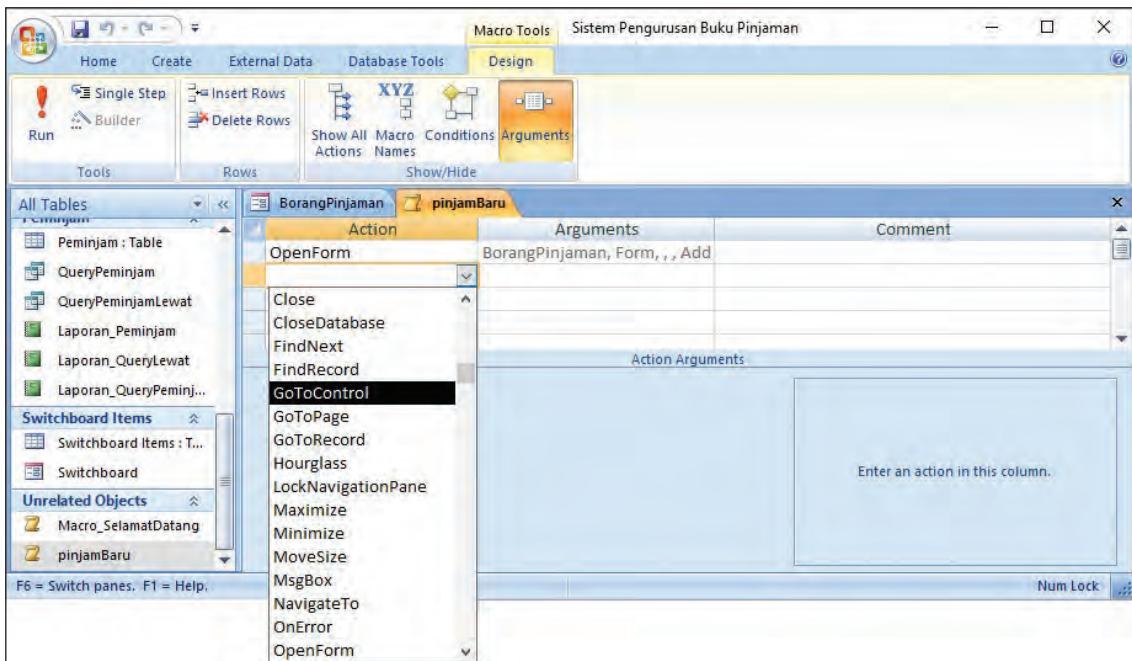


Membuka semula objek makro

Langkah-langkah:

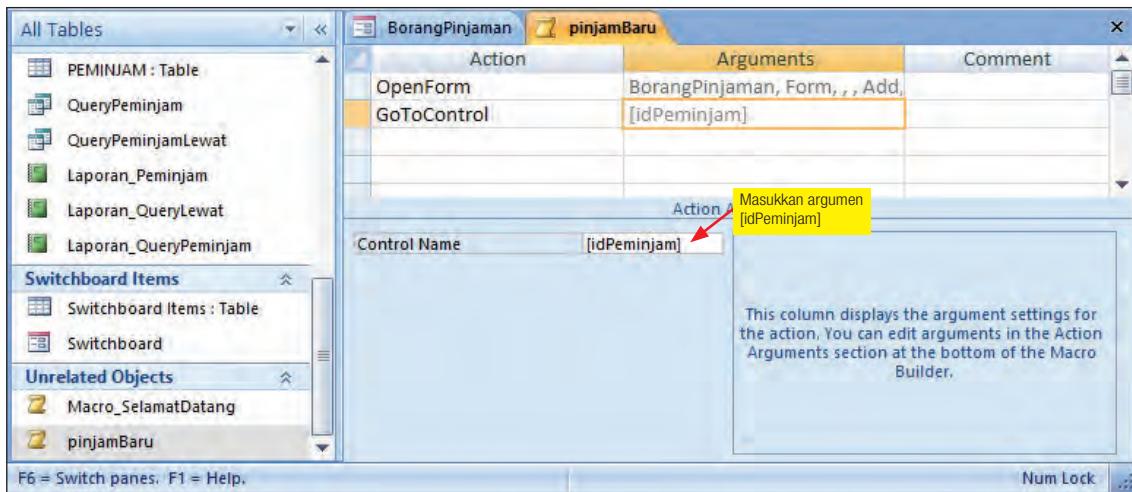
- 1 Pilih objek makro, klik kanan dan pilih *Design View*.
- 2 Dalam senarai arahan makro pinjamBaru, pilih baris baru dan klik *GoToControl* daripada senarai arahan *Action*.





Menambah arahan baharu *GoToControl* dalam makro pinjamBaru

- 3 Untuk lajur *Arguments* dalam makro pinjamBaru bagi arahan *GoToControl*, masukkan *[idPeminjam]*. Nama argumen mesti berada dalam kurungan square brackets “[” dan ”]”.



Memasukkan argumen dalam arahan baharu *GoToControl*

Nota: Nama argumen ialah nama *control* ataupun atribut atas borang. Ini boleh dikenal pasti daripada borang dalam *Design View*.

Menyemak nama kotak teks di atas borang sebagai nilai argumen

- Setelah itu, simpan makro baharu dan uji semula. Perhatikan bahawa kursor berada dalam kotak teks ID Peminjam. Lihat rajah yang ditunjukkan di bawah.

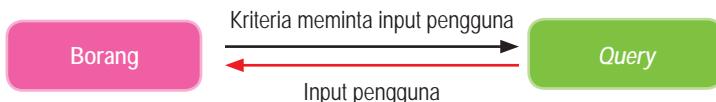
Menyimpan dan menguji arahan GoToControl dalam makro pinjamBaru

Setelah membina beberapa makro yang ringkas, marilah menggabungkan makro dengan borang yang menghasilkan laporan yang tepat dan menarik berdasarkan permasalahan dalam Sistem Pinjaman Buku Sekolah Bestari. Contoh 10 akan menunjukkan bagaimana untuk,

- menghasilkan kotak kombo dalam borang yang mempamerkan nilai-nilai atribut daripada jadual.



- menghasilkan *query* yang menggunakan kriteria daripada borang input pengguna.



- menghasilkan templat laporan *query* yang menggunakan *query* dengan kriteria.



- memanggil makro daripada butang *command* dalam borang untuk menghasilkan laporan *query* secara automatik berdasarkan input pengguna.



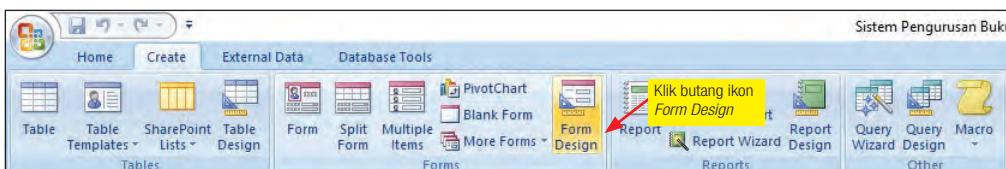
CONTOH (10)

Hasilkan borang (*Form*) Access baru dengan nama BorangSemakPeminjam. Borang ini menyemak butiran pinjaman untuk seseorang peminjam. Pengguna memilih ID peminjam daripada senarai kotak kombo. Setelah itu, pengguna boleh mengklik butang Butiran Pinjaman untuk memaparkan laporan pinjaman untuk peminjam tersebut.

Nota: Gunakan kriteria jenis parameter dalam *query*. Kriteria jenis parameter menggunakan ungkapan yang merujuk objek borang. Dalam contoh ini, ungkapan perlu merujuk nilai yang telah dipilih daripada kotak kombo borang BorangSemakPinjaman.

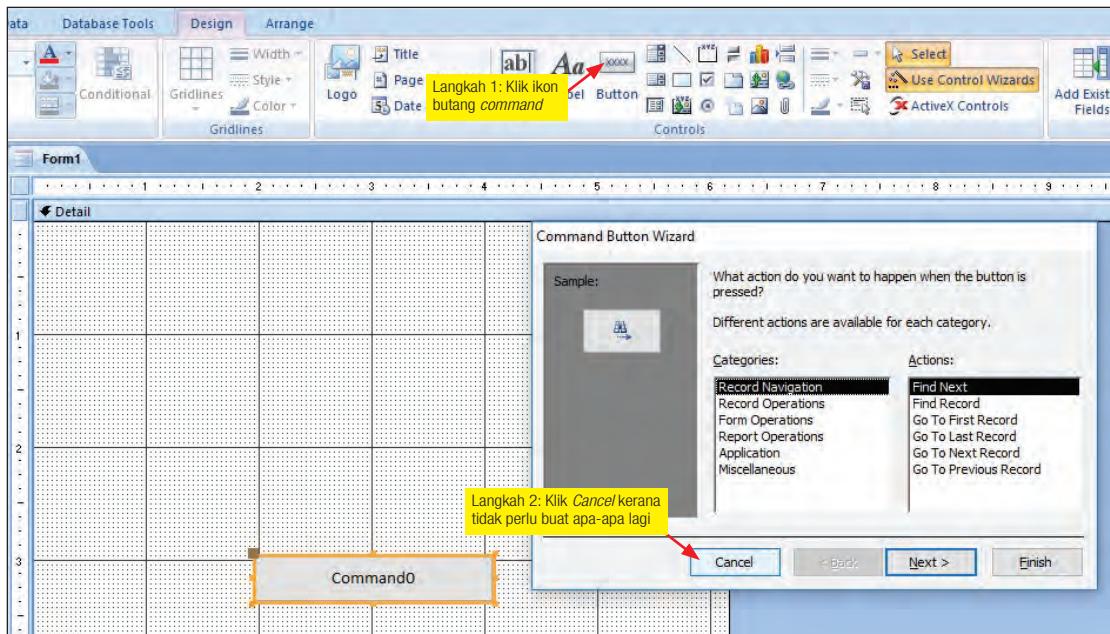
Penyelesaian

- 1 Klik butang ikon *Form Design* untuk menghasilkan borang kosong yang baharu.



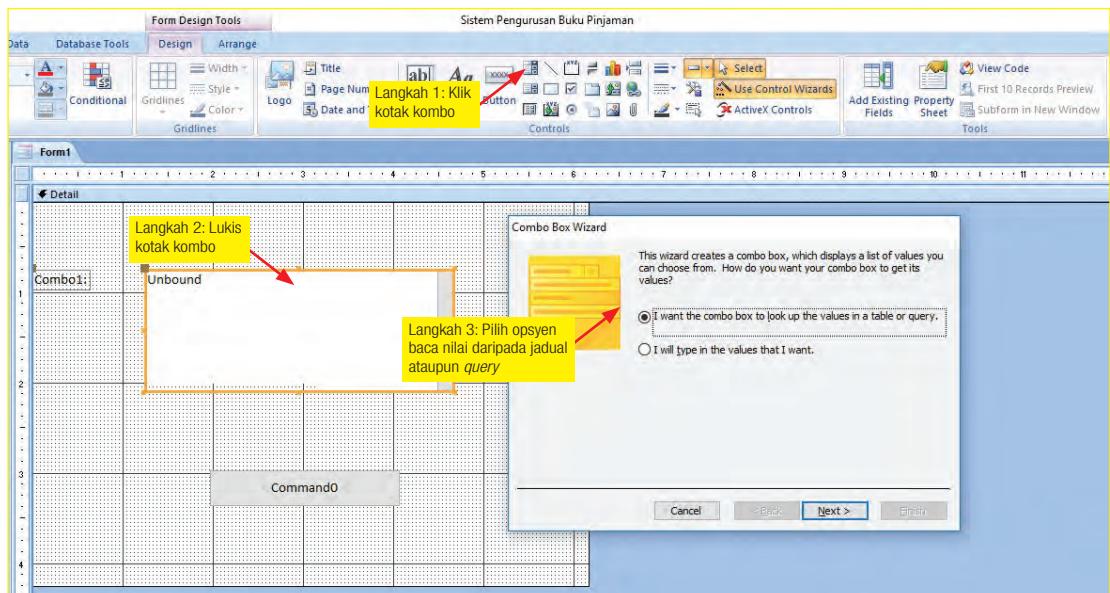
Membina borang kosong dalam *Form Design View*

- 2 Tambah butang command baru.



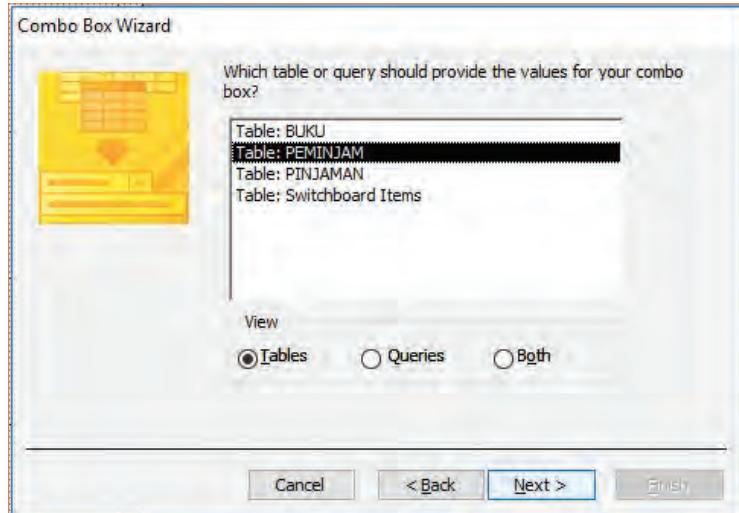
Menambah butang *command* kepada borang kosong

- 3 Tambah kotak kombo yang baharu. Pilih opsyen membaca nilai daripada jadual atau query.



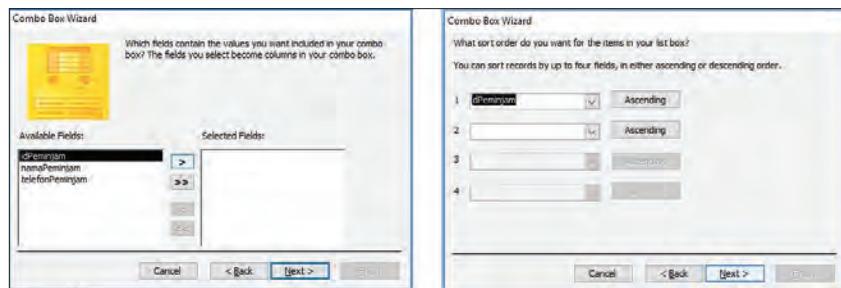
Menambah kotak kombo kosong kepada borang

4 Pilih jadual PEMINJAM.



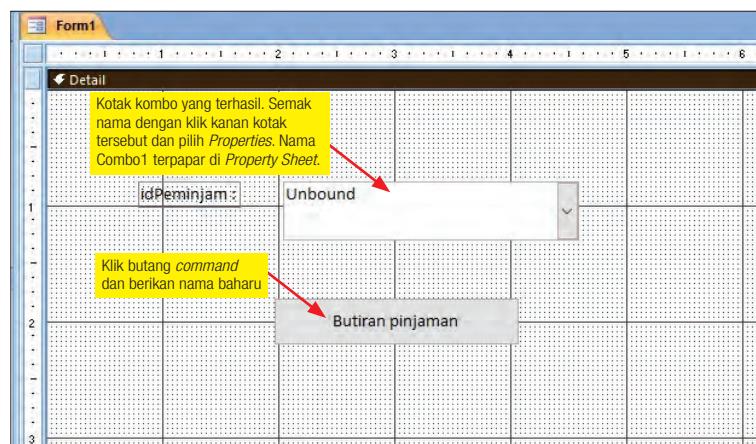
Pilih jadual PEMINJAM

5 Pilih atribut idPeminjam daripada jadual PEMINJAM.



Pilih atribut idPeminjam

6 Berikan nama idPeminjam kepada kotak kombo. Tukar nama label butang command kepada Butiran Pinjaman.



Berikan nama kotak kombo dan nama butang command

- Simpan nama borang sebagai BorangSemakPeminjam.
- Sebelum ini, ingatkah kita telah menghasilkan QueryPeminjam. Buat salinan baru dengan kaedah Copy-Paste, QueryPeminjam dalam panel All Tables dan namakan sebagai QueryPeminjamSpesifik.

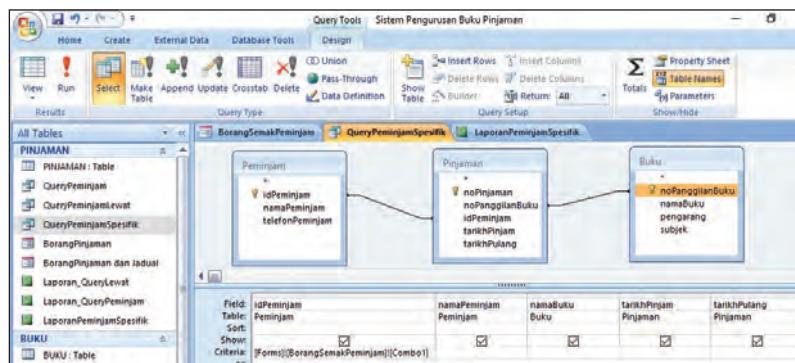
	IDPeminjam	namaPeminjam	namaBuku	tarikhPinjam	tarikhPulang
*	125007	Steve Shiva	Semantic Web	7/23/2016	8/22/2016
	125007	Steve Shiva	Brain-mind Machinery	7/20/2016	8/19/2016
	125888	Francis Rejang	C++ Programming	8/13/2016	9/12/2016
	125096	Elvis Tan Kim Hock	Software Engineering	8/23/2016	9/22/2016
	125088	Jeniffer Yeo Yin Yin	Brain-mind Machinery	6/7/2016	7/6/2016
	125001	Azlinna Adam	Pengaturcaraan C	9/29/2016	10/28/2016
	125001	Azlinna Adam	Computer Science for Matriculation	9/29/2016	10/28/2016

QueryPeminjam

- Dalam QueryPeminjamSpesifik, tukar ke mod *Design View*. Perhatikan lajur idPeminjam, lihat rajah yang ditunjukkan di bawah. Dalam baris *Criteria*, masukkan ungkapan yang berikut:

`[Forms]![BorangSemakPeminjam]![Combo1]`

Simpan QueryPeminjamSpesifik.



Set kriteria dalam QueryPeminjamSpesifik

- Hasilkan laporan baru daripada QueryPeminjamSpesifik.

Langkah 1: Pilih **QueryPeminjamSpesifik** daripada panel **All Tables**.

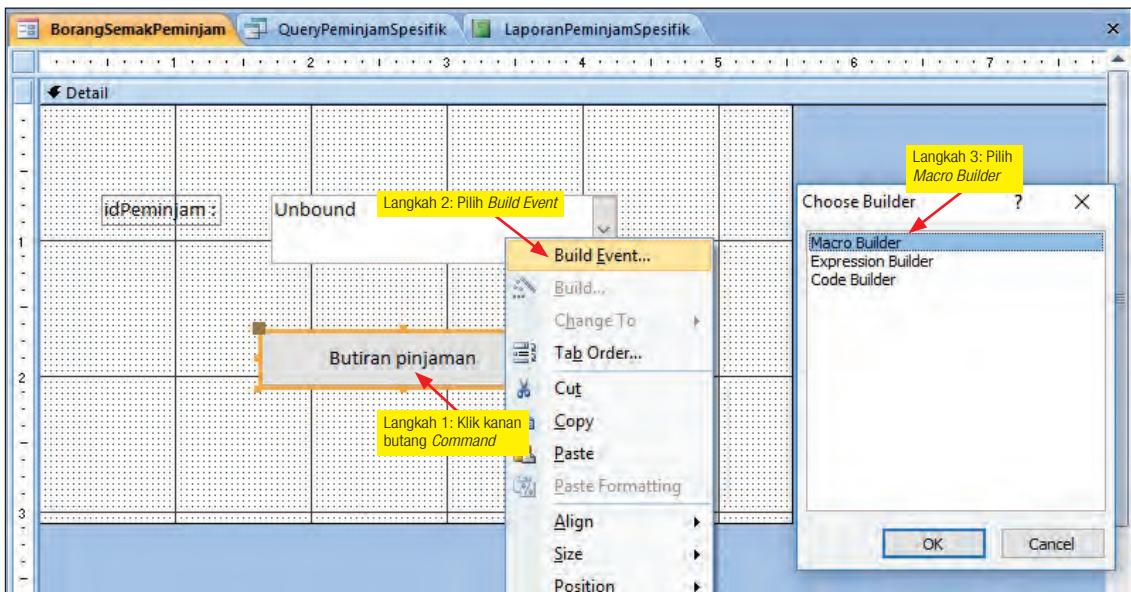
Langkah 2: Klik butang **Report**

Langkah 3: Sunting **report** yang terhasil. Klik dan ubah mana-mana bahagian

Bina laporan QueryPeminjamSpesifik

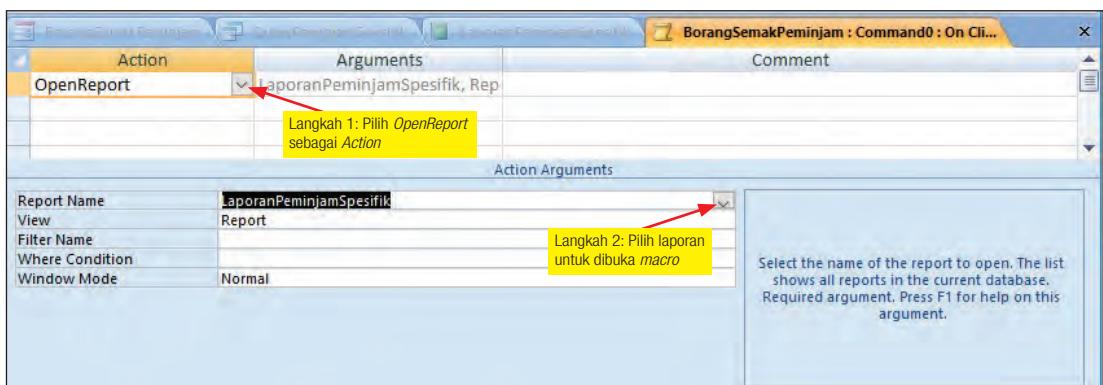


11 Kembali kepada BorangSemakPinjaman. Klik kanan butang Command "Butiran pinjaman" dan pilih Build Event. Daripada kotak dialog Choose Builder, klik Macro Builder.



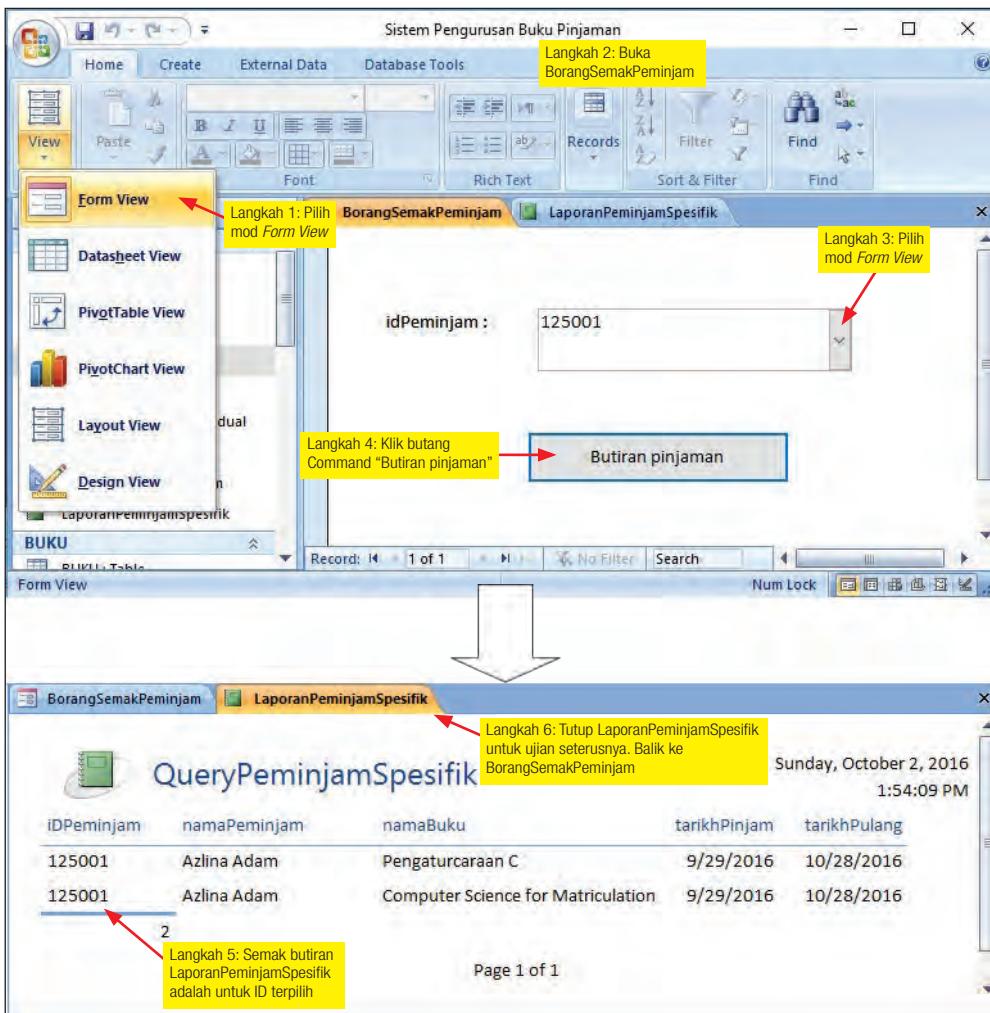
Butang butiran pinjaman memanggil makro

12 Dalam Macro Builder, lihat rajah di bawah, pilih Open Report daripada menu drop-down. Seterusnya, isikan butiran dalam panel Action Arguments dengan memilih nama report LaporanPeminjamSpesifik. Simpan makro tersebut.



Memilih laporan LaporanPeminjamSpesifik untuk dibuka makro

13 Uji butang command "Butiran pinjaman" dalam borang BorangSemakPeminjam. Lihat rajah di bawah.



Menguji BorangSemakPeminjam untuk membuka laporan LaporanPeminjamSpesifik

- (a) BorangSemakPeminjam dalam mod "Form View".
- (b) Pastikan LaporanPeminjamSpesifik telah ditutup.
- (c) Dari BorangSemakPeminjam, pilih idPeminjam daripada kotak kombo dan klik butang command "Butiran peminjam".
- (d) Perhatikan bahawa LaporanPeminjamSpesifik telah dibuka dan cuma mengandungi butiran untuk ID peminjam yang diminta sahaja.
- (e) Tutup LaporanPeminjamSpesifik setiap kali mencuba ID peminjam baru daripada BorangSemakPeminjam.

Mencipta dan Menggunakan *Switchboard*

Setelah menghasilkan borang dan laporan, pembangun pangkalan data perlu memikirkan “antara muka” yang menyatukan kesemua borang, laporan dan fitur-fitur penggunaan lain untuk kemudahan pengguna. Selain daripada bebas ralat, “antara muka” mestilah memaparkan data yang diperlukan secara logik dan lengkap. Pengguna dapat melihat ciri-ciri “antara muka” untuk mencari dan mengemaskinkan data dalam pangkalan data tersebut.

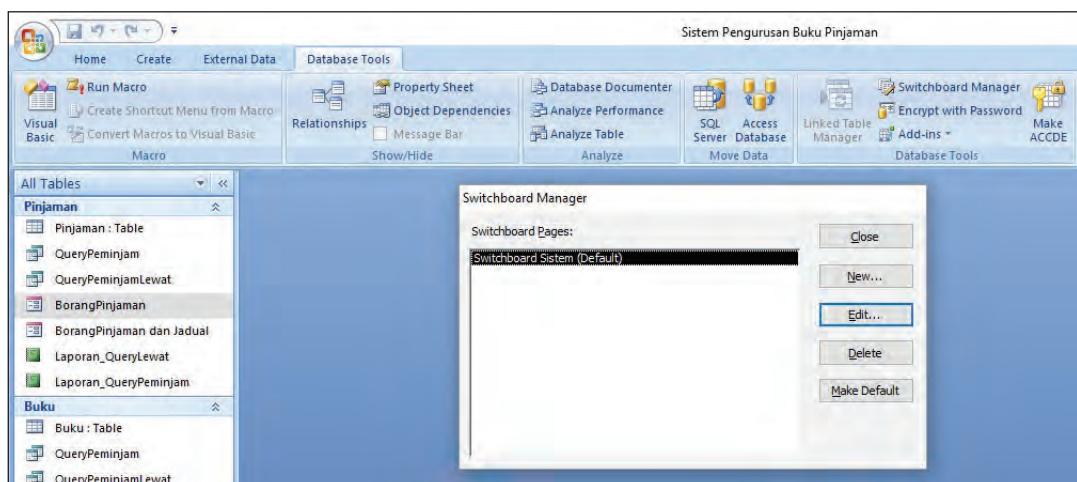
Tiga aspek yang perlu dipertimbangkan dalam penghasilan sistem antara muka ialah,

- a Setelah memasuki pangkalan data, pengguna boleh meninjau baris-baris data dalam jadual pangkalan data melalui borang yang bersesuaian. Jika perlu, pengguna boleh mengemaskinkan, membetulkan, menambahkan atau menghapuskan rekod data.
- b Sistem antara muka membantu pengguna menghasilkan laporan-laporan yang bersesuaian.
- c Pengguna dapat keluar daripada sistem selepas perubahan yang dibuat telah disimpan. Ciri-ciri seperti butang, memudahkan pengguna bernavigasi dan melaksanakan tugas berkenaan pangkalan data.

Switchboard ialah sistem antara muka bagi *Microsoft Access*. *Switchboard* boleh dimasukkan ke dalam pangkalan data untuk memudahkan pengguna mencari dan melaksanakan tugas-tugas berkait penggunaan pelbagai borang dan laporan dalam pangkalan data. *Switchboard* menggabungkan borang-borang, laporan-laporan, *query-query* dan lain-lain komponen *Access* dalam “antara muka” yang lebih mesra pengguna.

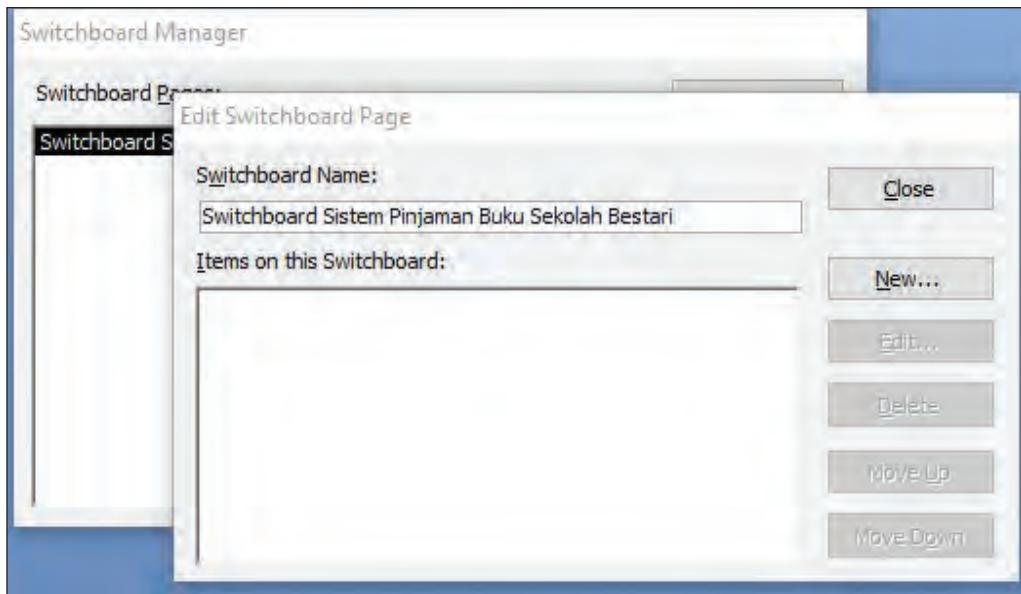
Langkah-langkah:

- 1 Kenal pasti borang-borang dan laporan-laporan yang diperlukan. Jika perlu, lakarkan di atas sekeping kertas. Pastikan susunan borang dan laporan adalah logikal dan memudahkan pengguna.
- 2 Dari tab *Database Tools*, klik *Switchboard Manager*. Tetingkap *Switchboard Manager* akan muncul.



Rajah 2.83 Menggunakan *Switchboard Manager* untuk menghasilkan *Switchboard* dan item

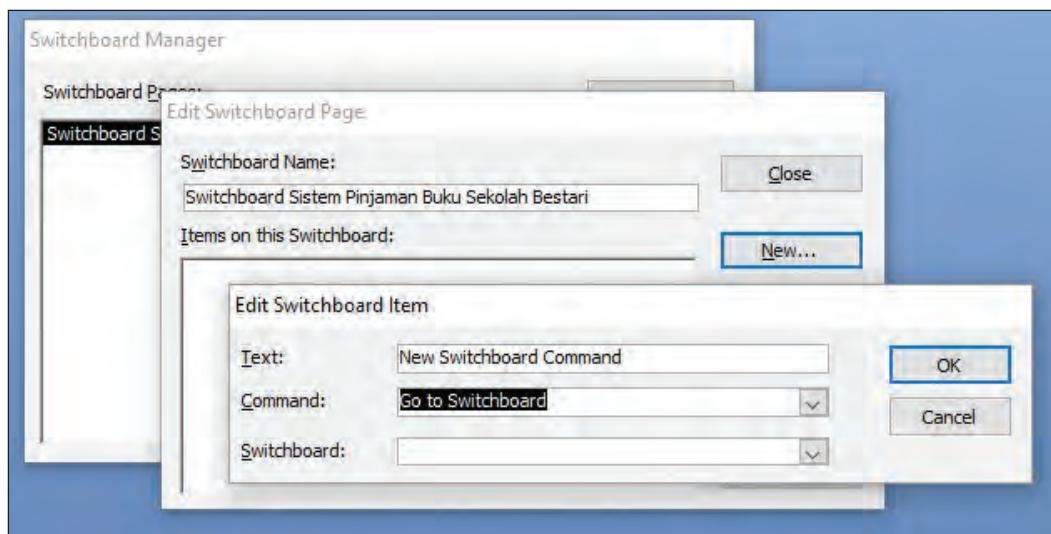
3 Tekan butang *Edit*.



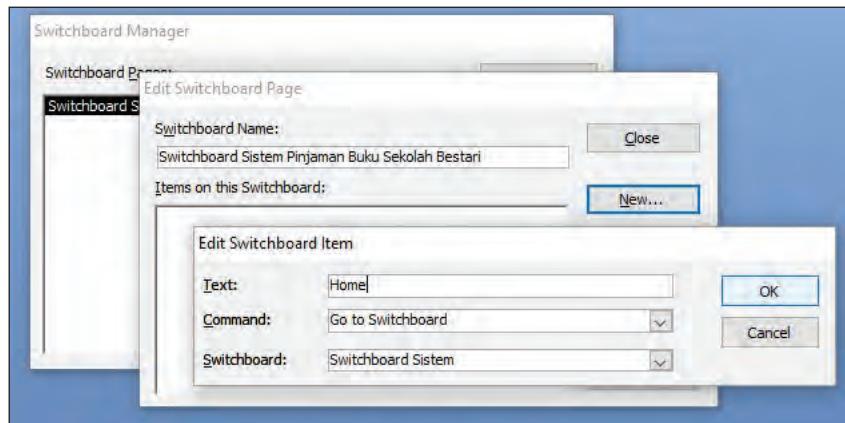
Rajah 2.84 Memberikan nama kepada *Switchboard*

4 Berikan nama yang bersesuaian dengan *Switchboard*.

5 Item menu berfungsi untuk melakukan pelbagai tugas seperti membuka *Switchboard* yang lain, membuka borang, membuka laporan, mereka bentuk aplikasi, melancarkan makro, melancarkan kod dan keluar daripada sistem. Untuk menambahkan item menu atas *Switchboard*, tekan butang *New*. Tekan *New*, masukkan nama item, kemudian pilih arahan yang sesuai daripada senarai *Command*. Klik *OK*. Perhatikan bahawa item baharu itu ditambah dalam senarai *Items on this Switchboard*.

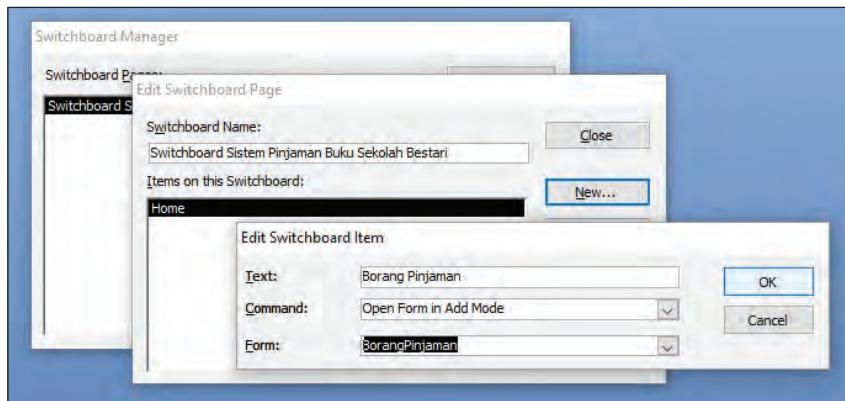


Rajah 2.85 Tekan *New* dan masukkan nama item



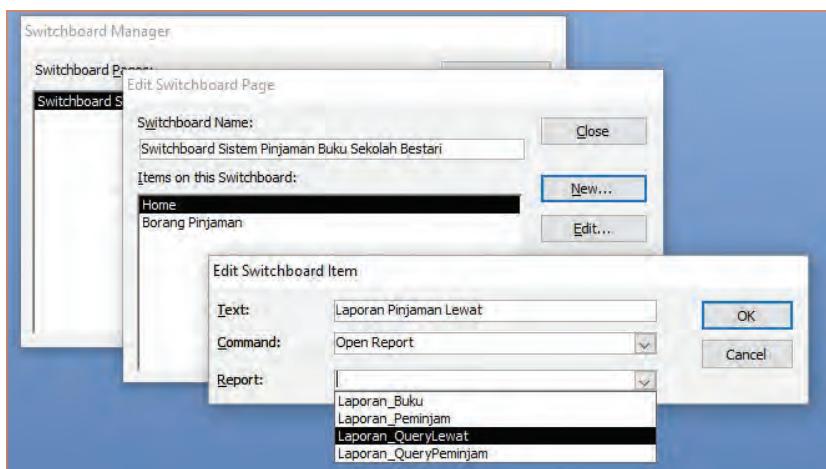
Rajah 2.86 Menambah item menu pertama - *Switchboard*

- 6 Untuk menambahkan item-item yang lain, klik *Add* sekali lagi. Sebagai contoh, untuk melihat borang pinjaman.

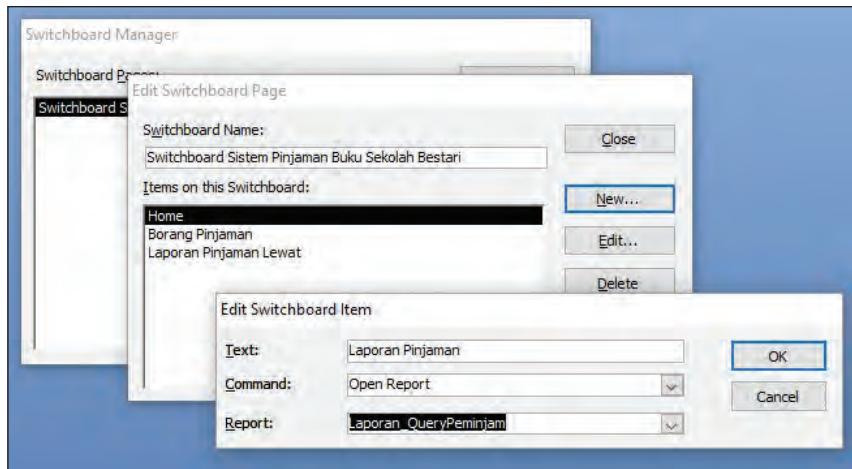


Rajah 2.87 Menambahkan item Borang Pinjaman

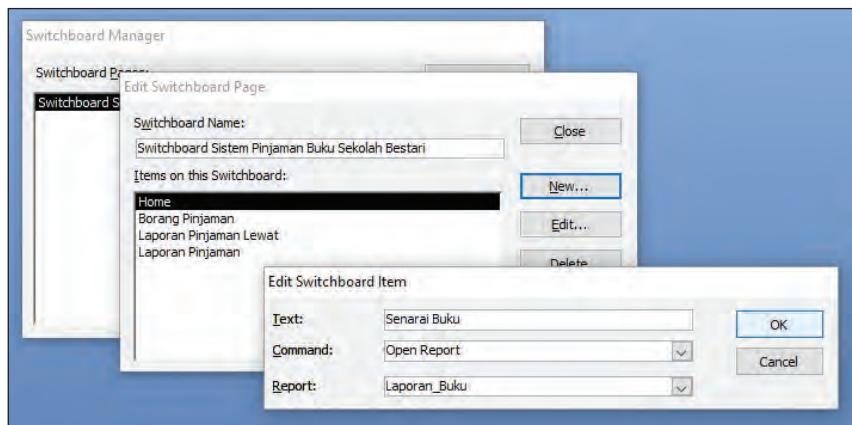
- 7 Klik *OK* dan ulangi langkah-langkah yang sama untuk menambahkan arahan-arahan yang lain.



Rajah 2.88 Menambahkan item Laporan Pinjaman Lewat

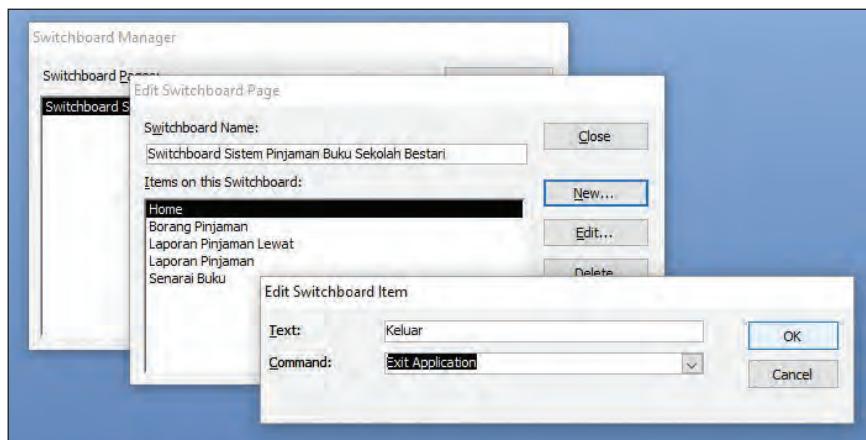


Rajah 2.89 Menambahkan item Laporan Pinjaman



Rajah 2.90 Menambahkan item Senarai Buku

- 8 Selain daripada melancarkan komponen-komponen Access, item *Switchboard* juga digunakan untuk keluar daripada aplikasi.



Rajah 2.91 Keluar aplikasi

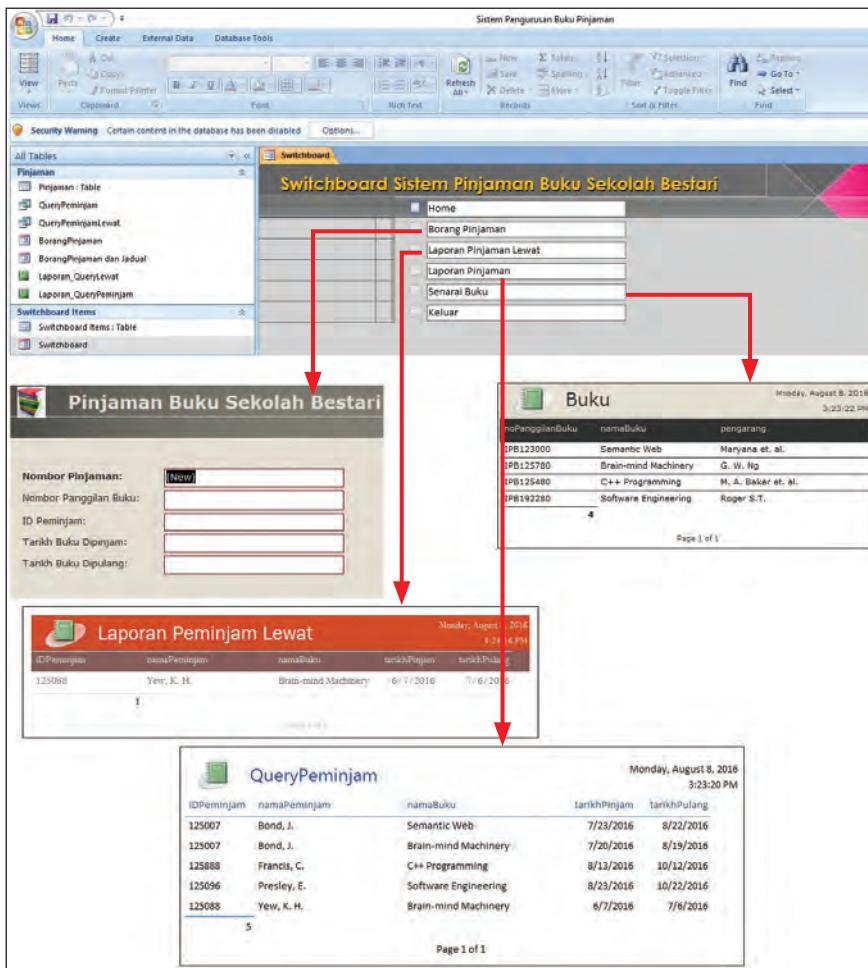


- 9 Hasilnya, *Switchboard* dengan item-item yang telah ditambah.



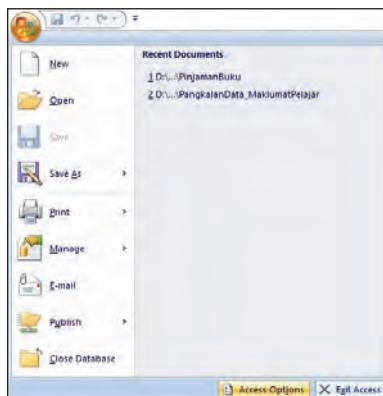
Rajah 2.92 *Switchboard* mempamerkan item-item yang boleh dilancarkan

- 10 Uji item-item *Switchboard* dengan menekan setiap satu. Sekiranya tiada ralat, borang, laporan ataupun respon yang dipilih melalui *Switchboard Manager* akan dilaksanakan.



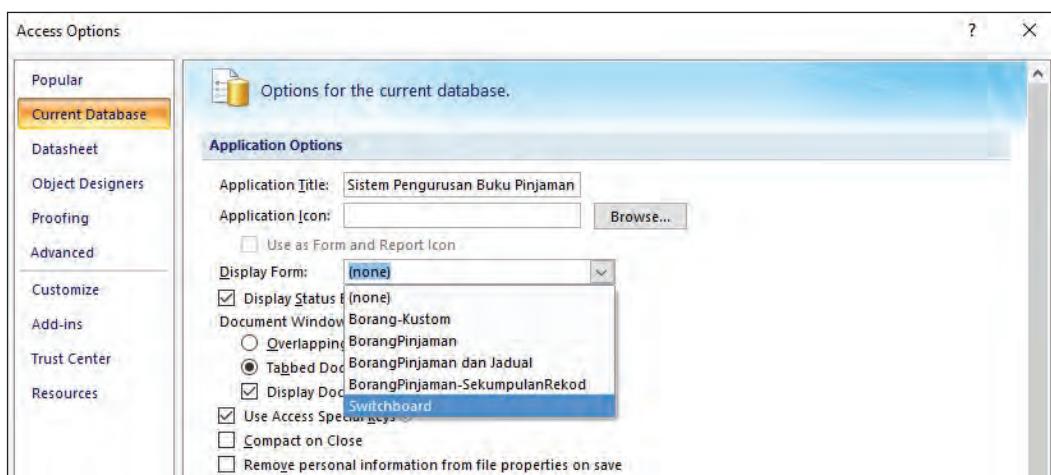
Rajah 2.93 *Switchboard* melancarkan item-item yang diklik

- 11** Untuk menjadikan *Switchboard* terbuka secara automatik setiap kali pangkalan data dibuka, tukar *Access* setting.
- Klik imej butang *Microsoft Office Button*.
 - Melancarkan *Access Options*.



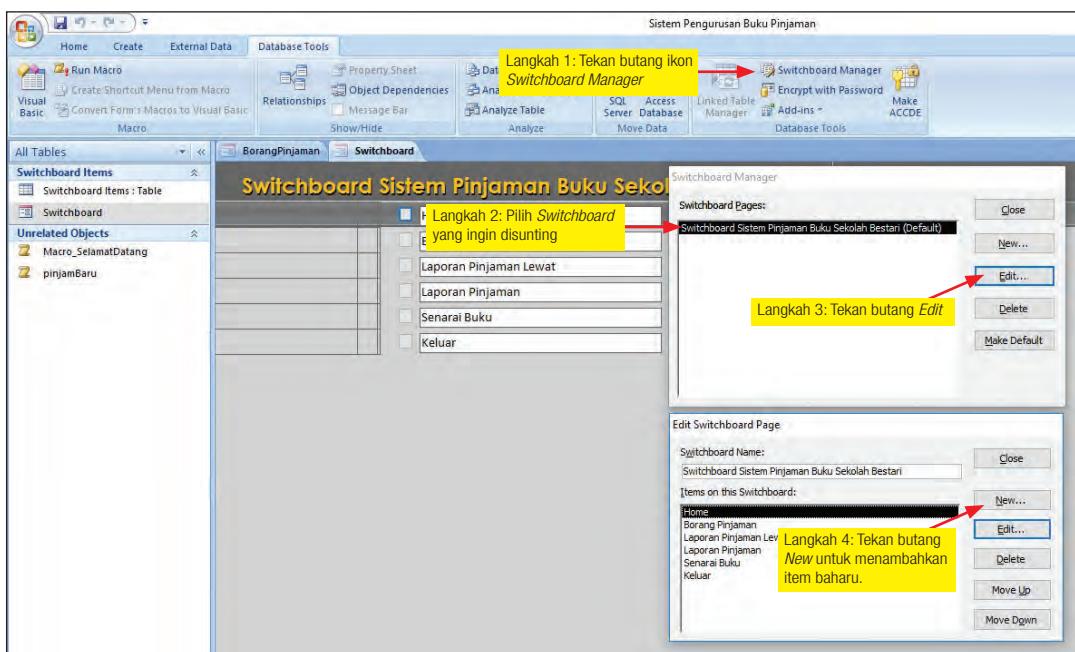
Rajah 2.94 Opsyen untuk membuka pangkalan data

- Klik *Access Options*.
- Klik *Current Database*.
- Pilih *Switchboard* daripada senarai *drop-down* di *Display Form*.
- Klik OK.
- Tutup dan buka semula pangkalan data untuk memastikan *Switchboard* terbuka secara automatik.



Rajah 2.95 Melancarkan *Switchboard* daripada menu *Current Database*

- 12** Bagaimanakah menyunting *Switchboard* untuk menambahkan atau membuang item? Gunakan *Switchboard Manager*. Lihat langkah-langkah yang menggunakan *Switchboard Manager* dalam rajah yang berikut:



Rajah 2.96 Menyunting *Switchboard* menggunakan *Switchboard Manager*



43

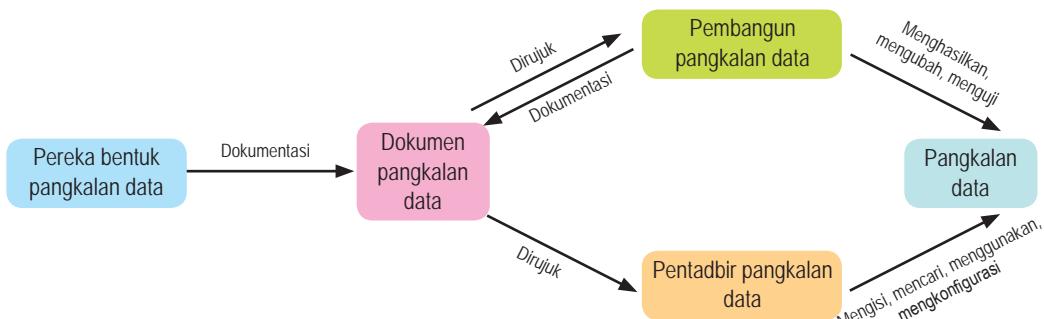
Makro dan *Switchboard*

Aktiviti Individu

- 1 Ubah suai *switchboard* dalam contoh yang sebelumnya. *Switchboard* asal untuk Sistem Pinjaman Buku Sekolah Bestari tidak mempunyai sebarang butang untuk pinjaman baharu dan butang untuk menyemak butiran pinjaman berdasarkan input ID pinjaman.
 - (a) Tambah makro pinjamanBaru kepada *switchboard* yang baharu.
 - (b) Tambah borang borangSemakPeminjam kepada *switchboard* yang baharu.
- 2 Perhatikan bahawa pangkalan data Pinjaman Buku tidak memperuntukkan data untuk mengesam sama ada peminjam sudah memulangkan buku. Oleh itu, anda dikehendaki mengkaji reka bentuk dan mengimplementasi pangkalan data untuk menambahkan data tersebut.
 - (a) ERD
 - (b) Skema hubungan.
 - (c) Borang
 - (d) Makro yang dibenamkan dalam butang kawalan
 - (e) Hasilkan *switchboard* yang bersesuaian

2.3.7 Mendokumentasikan Hasil Kerja

Pangkalan data akan digunakan untuk jangka waktu yang lama selepas pembangunan. Pembangunan melibatkan pemasukan dan pencarian data. Struktur pangkalan data jarang diubah tetapi setelah tempoh tertentu, pengemaskinian ataupun perubahan struktur pangkalan data mungkin diperlukan untuk memenuhi keperluan dan objektif baharu. Akan tetapi, pembangun pangkalan data (*database developer*) yang mengetahui reka bentuk pangkalan data mungkin telah ditukar.



Rajah 2.97 Mendokumentasikan kerja memudahkan pembangunan, pentadbiran dan pengubahan pangkalan data

Oleh itu, pentadbir pangkalan data perlu merujuk dokumentasi pangkalan data untuk memahami struktur. Pengubahan perlu dibuat secara optimum supaya tidak mengganggu integriti data asal. Bayangkan kesukaran yang bakal dihadapi sekiranya dokumentasi tidak dibuat sewaktu pembangunan pangkalan data?

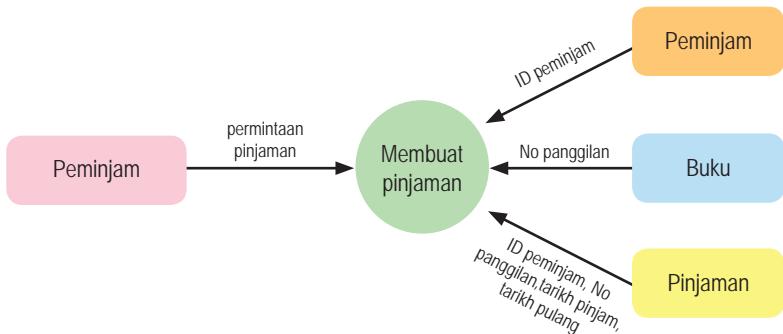
Dokumentasi membantu kefahaman terhadap pangkalan data melalui analisis ke atas grafik, jadual dan spesifikasi teks. Umumnya, dokumentasi harus merangkumi perkara seperti yang berikut:

- Nama pangkalan data:** Berikan butiran seperti tarikh, versi, pemilik dan pengguna pangkalan data.
- Objektif pangkalan data:** Gunakan teknik-teknik seperti soal selidik, temu bual, bengkel, pemerhatian dan kajian dokumentasi daripada vendor ataupun persekitaran organisasi.

Contoh:

- Pangkalan data membolehkan penyimpanan, pengemaskinian dan pencarian yang cekap ke atas rekod pinjaman buku, senarai buku dalam perpustakaan dan rekod peminjam.
 - Pangkalan data menjamin integriti data supaya rekod yang dimasukkan dan dikemas kini dari semasa ke semasa tidak akan hilang ataupun korup.
 - Pangkalan data boleh diguna pakai ataupun dikembangkan untuk keperluan ataupun aplikasi baharu di masa hadapan.
- Rajah alir data (Data flow diagram):** Kenal pasti bentuk data dan arus data dalam proses manual. Data-data yang diperlukan biasanya tercatat dalam dokumen atau formula. Hasilkan rajah alir data.





Rajah 2.98 Rajah alir data (Data flow diagram)

- (d) **Kamus data (data dictionary):** Kamus data ialah katalog bagi semua atribut untuk digunakan dalam pangkalan data. Kamus data mengandungi penjelasan terperinci untuk setiap atribut selepas analisis rajah alir data. Kamus data digunakan sebagai asas untuk menyediakan model konsep dan skema hubungan.

Contoh:

No.	Nama atribut	Jenis data	Format	Penjelasan
1	ID Peminjam	Char(6)	999999	Nombor ID unik murid
2	No panggilan	Char(6)	999999	Nombor panggilan unik setiap buku di perpustakaan sekolah
3	Tarikh pinjam	TIMESTAMP	dd-mm-yyyy	Tarikh pinjaman dibuat
4	Tarikh pulang	DATE	dd-mm-yyyy	Tarikh pulang adalah dua minggu selepas tarikh pinjaman

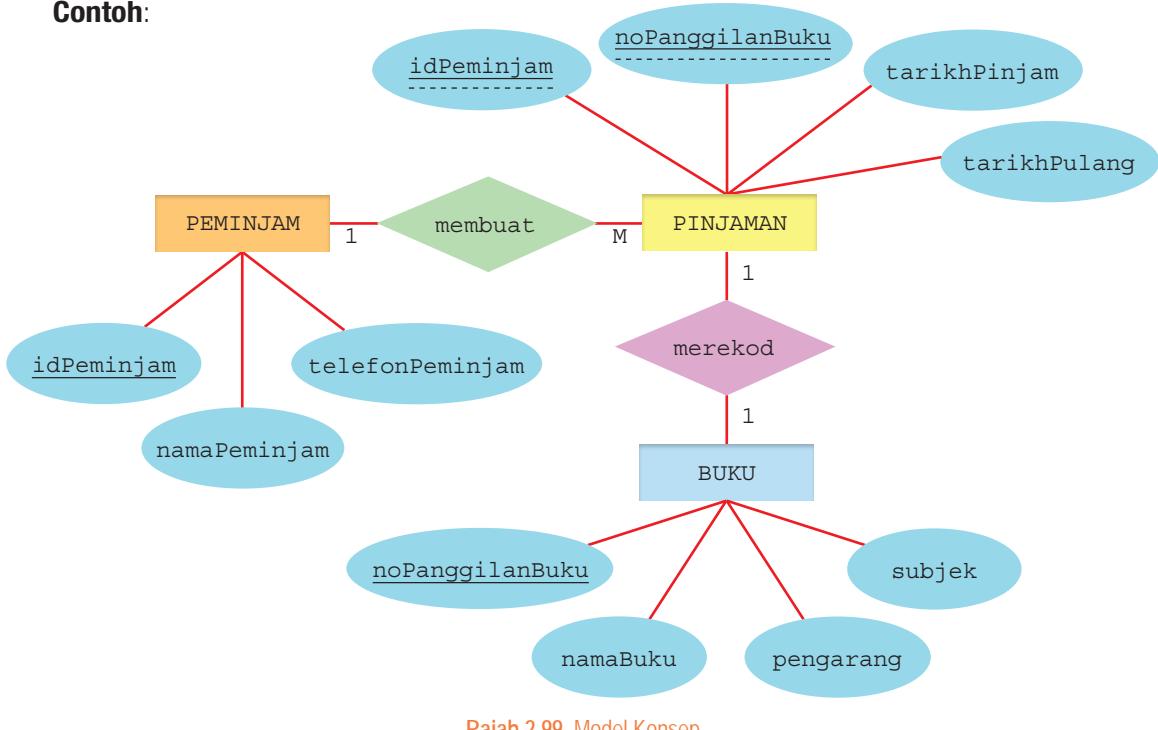
- (e) **Spesifikasi entiti dan atribut:** Hasilkann jadual yang merumuskan entiti dan atribut yang dikenal pasti dalam kajian.

Contoh:

Entiti	Atribut	Keterangan
Peminjam	ID murid peminjam, nama murid, nombor telefon bimbit	Entiti digunakan untuk merekod maklumat peminjam untuk memudahkan pengesahan identiti peminjam dan urusan menghubungi peminjam apabila perlu.
Buku	Nombor panggilan buku, nama buku, nama pengarang, subjek	Entiti digunakan untuk merekod maklumat buku untuk memudahkan pengesahan buku.
Pinjaman	Nombor panggilan buku, ID peminjam, tarikh pinjam buku, tarikh buku dipulangkan	Entiti digunakan untuk mengingati rekod tarikh buku dipinjam dan bakal dipulangkan.

- (f) **Model konsep:** Buat analisis dan lukis pemerhatian dalam bentuk ERD. ERD menunjukkan entiti, atribut dan perhubungan antara entiti. Tunjukkan atribut kunci primer, atribut kunci asing dan kardinaliti.

Contoh:



Rajah 2.99 Model Konsep

(g) **Reka bentuk logikal (struktur jadual):** Tunjukkan spesifikasi dalam bentuk skema hubungan yang ternormal. Gunakan skema hubungan.

Contoh:

- PEMINJAM(idPeminjam <KP>, namaPeminjam, telefonPeminjam)
- BUKU(noPanggilanBuku <KP>, namaBuku, pengarang, subjek)
- PINJAMAN(idPeminjam <KP><KA>, noPanggilanBuku <KP><KA>, tarikhPinjam, tarikhPulang)

(h) **Jadual hubungan ternormal dan sampel data:** Jadual-jadual hubungan berdasarkan skema hubungan dan mengandungi data-data contoh.

Contoh:

PEMINJAM

ID Peminjam	Nama Peminjam	Telefon Peminjam
125007	Harris bin Aman	016-6666666
125096	Francis Embong	012-2222222

BUKU

No Panggilan Buku	Nama Buku	Pengarang	Subjek
IPB124044	Java Programming	Maruyama et al	Komputer
IPB257868	C Programming	M. A. Bakar	Komputer
IPB192254	Computer Graphics	Shirley Jr	Komputer
IPB051375	Software Engineering	Roger Estain	Komputer
IPB051325	Information Systems, an Introduction	Farah et al	Komputer

PINJAMAN

No Panggilan Buku	ID Peminjam	Tarikh Pinjam	Tarikh Pulang
IPB124044	125007	27-Sep-2016	10-Okt-2016
IPB257868	125007	27-Sep-2016	10-Okt-2016
IPB192254	125007	27-Sep-2016	10-Okt-2016
IPB051375	125096	28-Sep-2016	11-Okt-2016
IPB051325	125096	28-Sep-2016	11-Okt-2016

- (i) Reka bentuk *query*, borang (*form*) dan laporan:
- Untuk *query*, lakaran menunjukkan atribut-atribut daripada jadual yang berasingan dan kriteria yang digunakan untuk memilih data yang ingin dipamerkan.
 - Untuk borang, lakaran menunjukkan bentuk atau rangka untuk memudahkan pengguna menyunting, memasukkan atau melihat data.
 - Untuk laporan, lakaran menunjukkan bentuk cetakan dengan memasukkan atribut-atribut tertentu, ouput pengiraan, tajuk, graf, gambar rajah dan maklumat-maklumat tambahan.



43

Dokumentasi

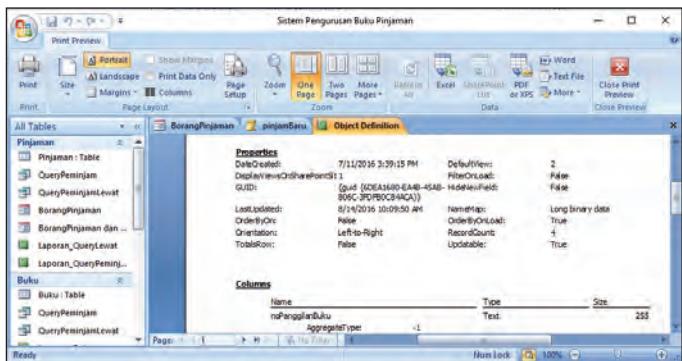
Aktiviti Individu

MS Access mempunyai fitur penjanaan dokumentasi untuk membolehkan pembangun pangkalan data melihat laporan struktur pangkalan data. Ikuti langkah-langkah berikut:

- 1 Pilih tab *Database Tools*.
- 2 Pilih ikon *Database Documenter*.
- 3 Daripada tetingkap *Documenter*, pilih jadual, *queries*, borang, laporan, makro dan objek-objek yang perlu didokumentasikan.
- 4 Klik butang OK.
- 5 Microsoft Access secara automatik akan menghasilkan dokumentasi.
- 6 Hasil dokumentasi boleh dirujuk terus atau dicetak.



Menggunakan fitur penghasilan dokumentasi dalam Access



Dokumen yang dijana secara automatik

Laporan ini boleh dirujuk silang dengan skema-skema hubungan dan rajah ERD untuk memastikan pangkalan data telah dibangunkan mengikut reka bentuk yang ditetapkan. Ini adalah penting untuk memastikan reka bentuk dalam 3NF dan sedia diguna pakai kelak.

Latihan Formatif 2.3

- 1 Terangkan fungsi-fungsi Sistem Pengurusan Pangkalan Data (DBMS).
- 2 Bincangkan kekurangan Sistem Pengurusan Pangkalan Data (DBMS).
- 3 Jelaskan masalah-masalah dalam sistem pemprosesan fail yang diselesaikan oleh Sistem Pengurusan Pangkalan Data (DBMS).
- 4 Senaraikan empat jenis medan jadual.
- 5 Apakah tujuan pengesahan data?
- 6 Kenal pasti operasi-operasi lazim bagi sesebuah Sistem Pengurusan Pangkalan Data (DBMS).

- 7 Terangkan langkah-langkah yang diperlukan untuk membangunkan suatu pangkalan data baharu daripada skema pangkalan data yang dinormalkan.



STANDARD PEMBELAJARAN

2.4.1 Menghasilkan sebuah pangkalan data yang ternormal

2.4.2 Membangunkan sistem pangkalan data dengan antara muka bergrafik menggunakan perisian pembangunan sistem pangkalan data mengikut SDLC untuk menyelesaikan masalah

Pembangunan Sistem Pangkalan Data

Pangkalan data menyimpan rekod-rekod secara tersusun supaya data-data mudah diisih, mudah dicari dan mudah dikemas kini. Pangkalan data menjadi komponen yang penting bagi sesebuah sistem maklumat kerana terdapat banyak data yang perlu diuruskan dalam sistem maklumat. Banyak contoh sistem maklumat yang menggunakan pangkalan data, misalnya sistem perbankan atas talian, sistem penempahan tiket penerbangan, laman sosial, sistem pinjaman buku perpustakaan, sistem maklumat murid sekolah dan sistem markah peperiksaan.

Pembangun perisian boleh membina perisian sistem atau perisian aplikasi yang boleh berhubung dengan sistem pangkalan data. Sistem pangkalan data seperti *Microsoft Access*, *Microsoft SQL*, *MySQL*, *Oracle SQL* dan sebagainya membantu penghasilan pangkalan data dan “antara muka” untuk melihat atau menyunting rekod pangkalan data. Jenis-jenis “antara muka” termasuklah borang (*form*), *query*, *view* dan laporan (*report*).

Oleh sebab kepentingan peranan pangkalan data dalam sistem maklumat, pembangunan sistem pangkalan data perlulah dilakukan secara sistematik supaya memenuhi keperluan pengguna. Fasa-fasa utama ialah merancang, menganalisis, mereka bentuk, membina dan menyelenggara sistem pangkalan data. Lihat Rajah 2.4.1.



Rajah 2.4.1 Fasa-fasa pembangunan pangkalan data untuk sistem maklumat

2.4.1 Penghasilan Sebuah Pangkalan Data yang Ternormal

Ingatkah lagi langkah-langkah untuk menghasilkan pangkalan data?

1 Merancang dan menganalisis



Rujuk kepada sumber persekitaran:
(a) Selidik keperluan pengguna.
(b) Kenal pasti objektif dan kegunaan.
(c) Kenal pasti data, borang dan laporan yang diperlukan dalam sistem.



Tahukah Anda?

Panduan untuk menukar hubungan 2NF kepada 3NF.

- 1 Kenal pasti kebergantungan fungsi transitif dan penentu yang bukan kunci primer.
- 2 Kumpulkan atribut yang bergantung secara transitif bersama satu salinan penentunya ke dalam satu hubungan baharu. Atribut yang tinggal dikumpulkan sebagai satu hubungan baharu.

2 Mereka bentuk

- (a) Reka bentuk konsep dengan menggunakan ERD. Analisis entiti, atribut, hubungan dan kekardinalan.
- (b) Reka bentuk logikal menggunakan skema hubungan. Terjemahkan ERD kepada skema hubungan dan buat penormalan.

3 Membina

- Terjemahkan skema kepada jadual hubungan:
- (a) Pilih Sistem Pengurusan Pangkalan Data yang sesuai.
- (b) Bina pangkalan data berdasarkan skema hubungan.
- (c) Bina antara muka yang menggunakan objek seperti query, borang dan laporan.
- (d) Masukkan data.
- (e) Uji pangkalan data secukupnya.

4 Menyelenggara

- (a) Pantau prestasi sistem pangkalan data dan membuat penalaan.
- (b) Mengubah suai sistem pangkalan data apabila perlu.

ID Murid	Nama	ID Murid	Alamat
10359	Adam	10359	134 South Query, Ipoh, Perak
10337	Alan	10337	10-2 Block C, Panorama, Subang Jaya, Selangor
10555	Sarah	10555	227 Prima Condo, Kubang Kerian, Kelantan
ID Murid	Gred		
10359	A-		
10337	A-		
10555	A		

Senario

Pembangunan modal insan di bawah Rancangan Malaysia ke-9 dan ke-10 menganjurkan bengkel-bengkel kemahiran seperti program *1MalaysiaGrip* untuk belia negara bagi meningkatkan keupayaan dan penguasaan ilmu pengetahuan. Untuk mengetahui keberkesanannya program tersebut, satu sistem pangkalan data dibangunkan untuk menyimpan rekod pencapaian peserta dalam ujian bagi kursus yang diambil. Pangkalan data menyimpan butiran peribadi peserta, butiran kursus yang ditawarkan dan butiran pencapaian peserta dalam ujian.

Sistem pangkalan data membolehkan pengguna memasukkan maklumat peserta, maklumat kursus dan maklumat pencapaian peserta bagi ujian yang diambil bagi setiap kursus. Pengguna boleh mencari butiran peribadi dan pencapaian ujian bagi semua peserta ataupun setiap peserta individu dalam laporan pencapaian.

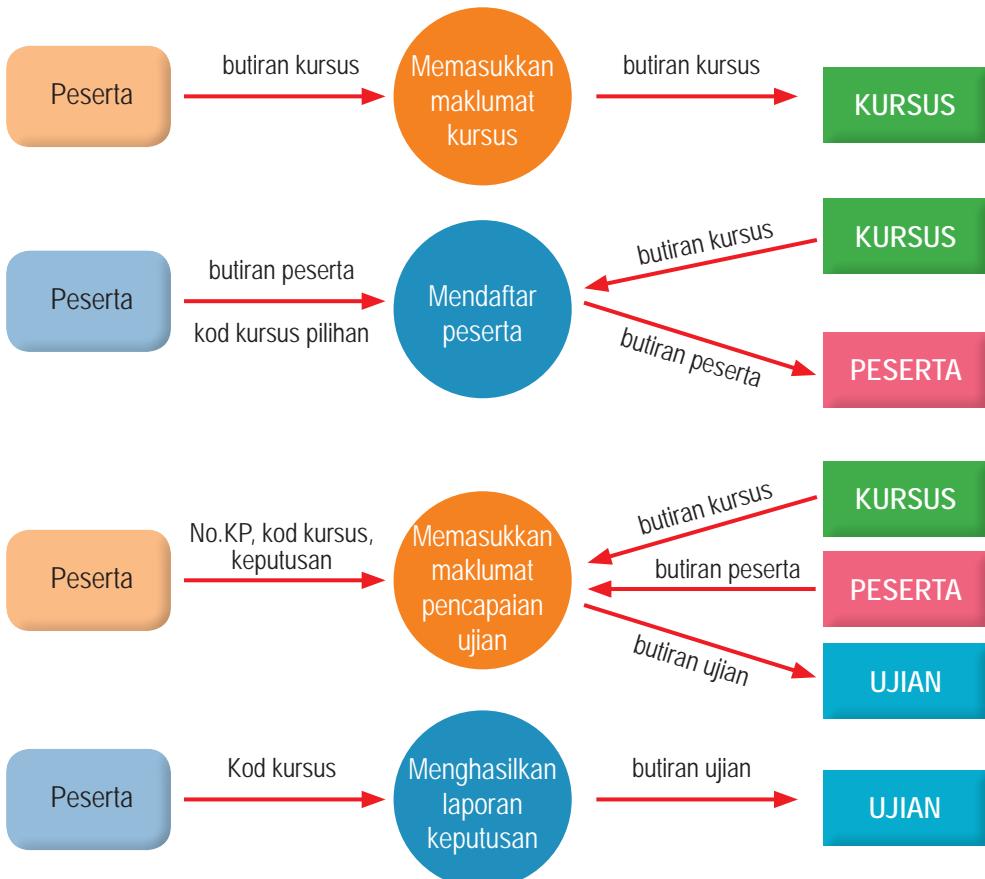
Merancang dan menganalisis

- Soal selidik pengguna tentang objektif, kegunaan dan sumber data untuk pangkalan data yang akan dibina.

Nama sistem	Sistem Pengurusan Peserta Kursus Kemahiran.
Objektif	Untuk merekod semua peserta program, kursus yang diambil dan pencapaian.
Kegunaan	<ul style="list-style-type: none"> (a) Memasukkan maklumat kursus. (b) Mendaftar peserta. (c) Memasukkan keputusan ujian. (d) Mencari rekod pencapaian peserta.
Sumber data	Borang pendaftaran peserta, senarai maklumat pengajar, senarai penyertaan bengkel, senarai keputusan ujian.

- Selidik penggunaan dan data-data yang diperlukan.

- Selidik bagaimana data dihasilkan dan digunakan oleh sistem dan entiti-entiti yang berkaitan.
- Selidik borang data dan sampel laporan manual.
- Hasilkann carta pengaliran data.



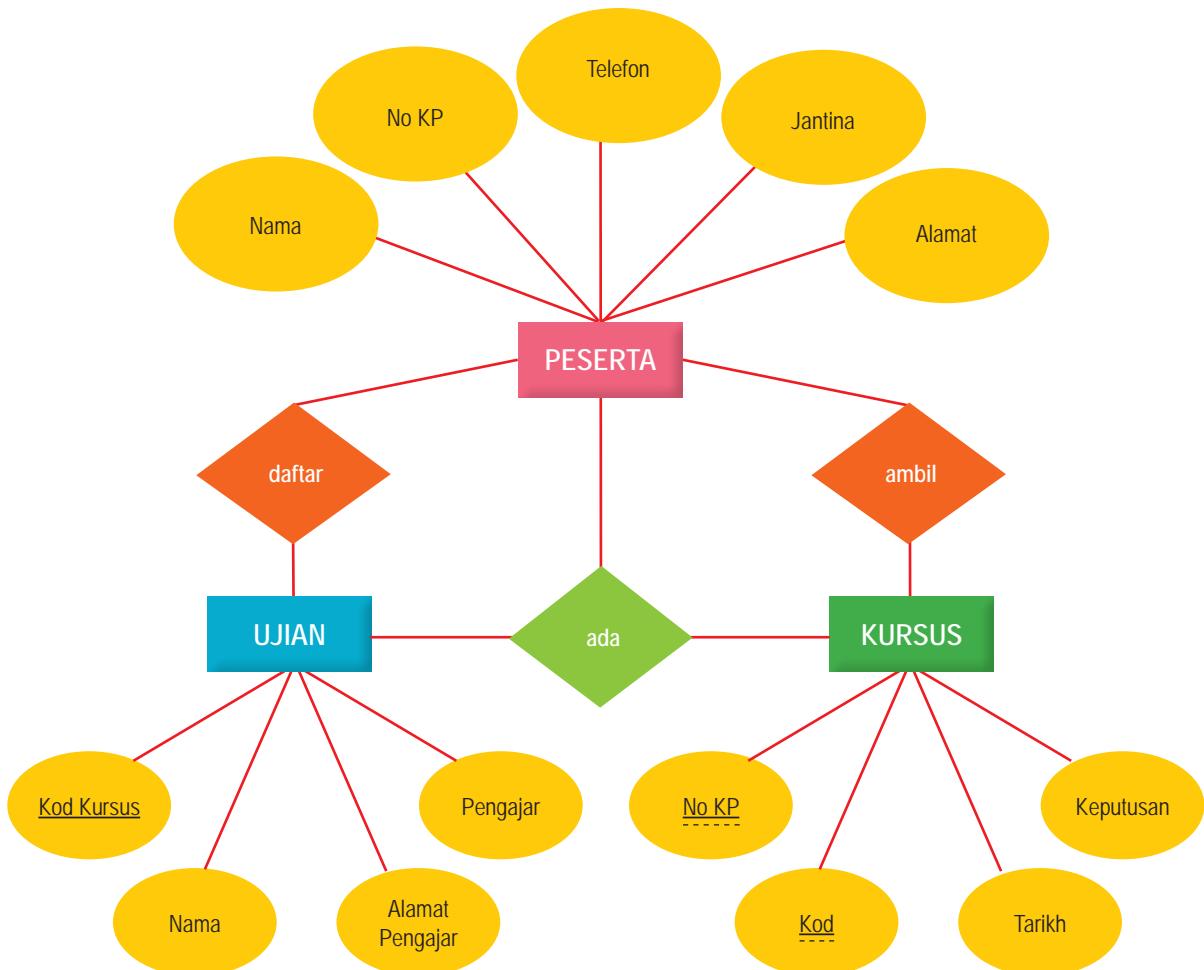
Rajah 2.4.2 Rajah alir data untuk Sistem Pengurusan Kursus Kemahiran

Mereka bentuk

Reka bentuk konsep

- Kenal pasti senarai entiti: PESERTA, KURSUS, UJIAN.
- Kenal pasti atribut bagi setiap entiti, sebagai contoh PESERTA mempunyai No.KP, nombor telefon, jantina dan alamat.
- Kenal pasti atribut yang menjadi identiti unik kepada entiti untuk dijadikan atribut kunci. Sebagai contoh, No. KP menjadi identiti unik kepada entiti PESERTA; kod sebagai identiti unik kepada entiti KURSUS; dan kedua-dua No. KP dan kod sebagai identiti unik kepada entiti UJIAN.

Untuk memudahkan reka bentuk konsep, lakukan rajah ERD dan buat analisis. Lihat Rajah 2.4.3.



Rajah 2.4.3 ERD untuk pangkalan data bagi program bengkel-bengkel pembangunan profesional

Dalam ERD di atas, PESERTA boleh mengambil lebih daripada satu KURSUS dan setiap KURSUS mempunyai lebih daripada satu PESERTA (kekardinalan M:N); setiap PESERTA boleh mengambil satu atau lebih UJIAN dan setiap UJIAN boleh diambil oleh satu atau lebih PESERTA (kekardinalan M:N); dan setiap KURSUS mempunyai satu UJIAN sahaja (kekardinalan 1:1).

Lakarkan sebuah jadual yang boleh digunakan untuk menyimpan data-data dalam ERD. Uji jadual dengan mengutip data contoh dan tampilkan data-data yang terkumpul dalam jadual tersebut. Berikan nama jadual sebagai UJIAN. Jadual UJIAN yang mentah ditunjukkan seperti dalam Jadual 2.4.1. Jadual di bawah ialah jadual tidak ternormal ataupun dalam bentuk ONF.

Jadual 2.4.1 Jadual UJIAN Tidak-Ternormal ONF

Nama	No. KP	Telefon	Jantina	Alamat	Kod Kursus	Kursus	Pengajar	Alamat Pengajar	Tarikh Ujian	Keputusan
Abdul Salam	850415-02-9234	012-1111111	M	2, Seksyen 13, 40100 Shah Alam, Selangor	4040	Rangkaian Komputer	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-Mar-16	Lulus
					3592	Juruteknik Komputer Bersijil	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	15-Apr-16	Lulus
Adam Majid	900926-01-8577	012-2222222	M	10B, Cheras, 56000 Kuala Lumpur	4040	Rangkaian Komputer	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-Mar-16	Lulus
Amy Tan	900426-02-5555	011-3333333	F	12, Taman Kenyalang, 93400 Kuching, Sarawak	3952	Juruteknik Komputer Bersijil	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	1-Mar-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani, 31650 Ipoh, Perak	3953	Sekuriti Komputer	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	2-Mar-16	Lulus
					4042	Teknologi Android	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-May-16	Lulus
					4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	5-Jun-16	Gagal
					4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	15-Aug-16	Lulus
Charles Gunalan	950216-05-6552	010-5555555	M	18, Taman Tun Dr Ismail, 60000 Kuala Lumpur	3954	Rangkaian Kompute	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	3-Mar-16	Lulus
Chin Xiao Ying	931211-07-1523	012-3333222	F	8, Taman SEA, SS2, 47410 Petaling Jaya, Selangor	4160	Pembangunan Perisian Microsoft	PEAK INNOVATION	Lot 1, Technologi Park MALAYSIA, 57000 Kuala Lumpur	4-Mar-16	Lulus
Dorothy Teng	890705-08-7755	016-3333566	F	3A, Damansara Utama, 47400 Petaling Jaya, Selangor	4656	Komunikasi Bahasa Inggeris Perniagaan	CAMBRIDGE ENGLISH	9, Merchant Square, 47410 Petaling Jaya, Selangor	8-Mar-16	Lulus
Edward Xavier	900502-03-5556	011-3244566	M	C3, Lorong Alor Akar, 25050 Kuantan, Pahang	4555	Pengaturcaraan Java Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	29-Feb-16	Lulus
Marissa O'Donnell	900422-02-5532	016-1133568	M	12-A, Bukit Kiara Residences, Sri Hartamas, 50490 Kuala Lumpur	4555	Pengaturcaraan Java Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	29-Feb-16	Lulus
Vivien F.	900413-02-3355	012-6678999	M	3, Meru Jaya, 31650 Ipoh, Perak	4555	Pengaturcaraan Java Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	29-Feb-16	Lulus

Menukar jadual ONF kepada struktur 1NF

Ciri-ciri struktur 1NF:

Setiap baris mengandungi data atomik. Maksudnya, dalam satu baris, tidak dibenarkan satu atribut mengandungi lebih daripada satu data. Jadual mesti sekurang-kurangnya mempunyai satu atribut kunci.

Jadual 2.4.2 Jadual UJIAN berstruktur 1NF

Nama	No. KP	Telefon	Jantina	Alamat	Kod Kursus	Kursus	Pengajar	Alamat Pengajar	Tarikh Ujian	Keputusan
Abdul Salam	850415-02-9234	012-1111111	M	2, Seksyen 13, 40100 Shah Alam, Selangor	4040	Rangkaian Komputer	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-Mar-16	Lulus
Abdul Salam	850415-02-9234	012-1111111	M	2, Seksyen 13, 40100 Shah Alam, Selangor	3592	Juruteknik Komputer Bersijil	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	15-Apr-16	Lulus
Adam Majid	900926-01-8577	012-2222222	M	10B, Cheras, 56000 Kuala Lumpur	4040	Rangkaian Komputer	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-Mar-16	Lulus
Amy Tan	900426-02-5555	011-3333333	F	12, Taman Kenyalang, 93400 Kuching, Sarawak	3952	Juruteknik Komputer Bersijil	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	1-Mar-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani, 31650 Ipoh, Perak	3953	Sekuriti Komputer	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	2-Mar-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani, 31650 Ipoh, Perak	4042	Teknologi Android	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-May-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani, 31650 Ipoh, Perak	4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	5-Jun-16	Gagal
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani, 31650 Ipoh, Perak	4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	15-Aug-16	Lulus
Charles Gunalan	950216-05-6552	010-5555555	M	18, Taman Tun Dr Ismail, 60000 Kuala Lumpur	3954	Rangkaian Kompute	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	3-Mar-16	Lulus
Chin Xiao Ying	931211-07-1523	012-3333222	F	8, Taman SEA, SS2, 47410 Petaling Jaya, Selangor	4160	Pembangunan Perisian Microsoft	PEAK INNOVATION	Lot 1, Technologi Park MALAYSIA, 57000 Kuala Lumpur	4-Mar-16	Lulus
Dorothy Teng	890705-08-7755	016-3333566	F	3A, Damansara Utama, 47400 Petaling Jaya, Selangor	4656	Komunikasi Bahasa Inggeris Perniagaan	CAMBRIDGE ENGLISH	9, Merchant Square, 47410 Petaling Jaya, Selangor	8-Mar-16	Lulus
Edward Xavier	900502-03-5556	011-3244566	M	C3, Lorong Alor Akar, 25050 Kuantan, Pahang	4555	Pengaturcaraan Java Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	29-Feb-16	Lulus
Marissa O'Donnell	900422-02-5532	016-1133568	M	12-A, Bukit Kiara Residences, Sri Hartamas, 50490 Kuala Lumpur	4555	Pengaturcaraan Java Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	29-Feb-16	Lulus
Vivien F.	900413-02-3355	012-6678999	M	3, Meru Jaya, 31650 Ipoh, Perak	4555	Pengaturcaraan Java Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	29-Feb-16	Lulus

Reka bentuk logikal 1NF

Rujuk kepada data-data yang dikumpul dalam jadual 1NF dan kenal pasti nama atribut untuk setiap lajur data. Kenal pasti atribut yang unik dalam setiap baris rekod. Perhatikan bahawa kombinasi atribut No. KP dan kod kursus adalah unik bagi setiap baris. Oleh itu, kedua-dua atribut tersebut sesuai dijadikan kunci primer.

UJIAN(nama, nokp <KP>, telefon, jantina, alamat, kodKursus <KP>, kursus, pengajar, alamatPengajar, tarikhUjian, keputusan)

Reka bentuk logikal 2NF

Perhatikan bahawa penukaran kepada 1NF tidak mengurangkan ulangan. Sebagai contoh, disebabkan Encik Abdul Salam menduduki dua ujian kursus, maka butiran Encik Abdul Salam telah diulangi sebanyak dua kali. Satu lagi contoh ialah Encik Bryan Ng. Encik Bryan Ng telah mengikuti tiga ujian bengkel kemahiran. Untuk bengkel kemahiran Pembangunan Rangkai Mobil Bersijil, Encik Bryan Ng terpaksa mengambil ujian sebanyak dua kali kerana gagal pada kali pertama. Oleh itu, jadual 1NF masih tidak sesuai dilaksanakan dalam sistem pangkalan data.

Marilah membuat penormalan 1NF kepada 2NF. Semak sama ada terdapat atribut yang tidak bergantung kepada kedua-dua atribut No. KP dan kodKursus. Kebergantungan fungsi separa adalah petunjuk kepada pemisahan atribut-atribut terlibat untuk membentuk jadual yang baru. Keluarkan kumpulan atribut yang bergantung kepada atribut No. KP sahaja.

Sebagai contoh, lihat atribut-atribut yang berulang bagi nama peserta Abdul Salam dan Bryan Ng dalam jadual yang berikut: Nama, No.KP, Telefon, Jantina dan Alamat. Atribut-atribut ini bergantung kepada No.KP sahaja.

Jadual 2.4.3 Kenali atribut-atribut yang berulang daripada data-data jadual tidak-ternormal

Nama	No. KP	Telefon	Jantina	Alamat	Kod Kursus	Kursus	Pengajar	Alamat Pengajar	Tarikh Ujian	Keputusan
Abdul Salam	850415-02-9234	012-1111111	M	2, Seksyen 13, 40100 Shah Alam, Selangor	4040	Rangkaian Komputer	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-Mar-16	Lulus
Abdul Salam	850415-02-9234	012-1111111	M	2, Seksyen 13, 40100 Shah Alam, Selangor	3592	Juruteknik Komputer Bersijil	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	15-Apr-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani,31650 Ipoh, Perak	3953	Sekuriti Komputer	COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur	2-Mar-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani,31650 Ipoh, Perak	4042	Teknologi Android	ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak	5-May-16	Lulus
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani,31650 Ipoh, Perak	4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	5-Jun-16	Gagal
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani,31650 Ipoh, Perak	4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur	15-Aug-16	Lulus

Dua kumpulan atribut terhasil daripada pemisahan tersebut: Kumpulan lama UJIAN yang telah dinormalkan dan kumpulan baharu yang diberikan nama entiti PESERTA. Kenal pasti atribut kunci dalam kumpulan PESERTA. Kemudian, jadikan atribut kunci No. KP dalam kumpulan PESERTA sebagai kunci asing kepada kumpulan UJIAN.

```
UJIAN( nama, nokp <KP>, telefon, jantina, alamat), kodKursus <KP>, kursus, pengajar, alamatPengajar, tarikhUjian, keputusan)
```



```
UJIAN ( nokp <KP> <KA>, kodKursus <KP>, kursus, pengajar,  
alamatPengajar, tarikhUjian, keputusan)
```

```
PESERTA ( nama, nokp <KP>, telefon, jantina, alamat)
```

Perhatikan skema hubungan yang baharu. Analisis sama ada masih terdapat kebergantungan fungsi separa dalam skema yang baharu. Dalam kumpulan UJIAN, masih terdapat kebergantungan fungsi separa ke atas atribut kunci primer kodKursus.

Sebagai contoh, dalam kumpulan (*nokp <KP> <KA>*, *kodKursus <KP>*, *kursus*, *pengajar*, *alamatPengajar*, *tarikhUjian*, *keputusan*) atribut kursus, pengajar dan alamat pengajar dapat ditentukan secara eksklusif menggunakan kodKursus sahaja. Oleh itu, kumpulan (*kodKursus*, *kursus*, *pengajar*, *alamatPengajar*) boleh diasing sebagai kumpulan baharu. Perhatikan bahawa kodKursus dalam kumpulan baharu itu juga merupakan atribut asing. Kumpulan asal UJIAN mengekalkan atribut kodKursus bersama atribut No. KP untuk menentukan secara eksklusif tarikh ujian dan keputusan ujian peserta.

```
UJIAN(nokp <KP> <KA>, kodKursus <KP>, kursus, pengajar, alamatPengajar,  
tarikhUjian, keputusan)  
PESERTA(nama, nokp <KP>, telefon, jantina, alamat)
```



```
UJIAN(nokp <KP> <KA>, kodKursus <KP> <KA>, tarikhUjian, keputusan)
```

```
KURSUS(kodKursus <KP>, kursus, pengajar, alamatPengajar)
```

```
PESERTA(_nama, nokp <KP>, telefon, jantina, alamat, poskod)
```

Kumpulan skema baharu kini berada dalam bentuk 2NF kerana tiada lagi kebergantungan separa antara mana-mana atribut dengan mana-mana atribut kunci primer:

```

UJIAN (nokp <KP> <KA>, kodKursus <KP> <KA>, tarikhUjian, keputusan)
KURSUS (kodKursus <KP>, kursus, pengajar, alamatPengajar)
PESERTA (nama, nokp <KP>, telefon, jantina, alamat)

```

Reka bentuk logikal 3NF

Dalam jadual Kursus, atribut pengajar dan atribut alamatPengajar adalah atribut-atribut yang biasa sahaja. Namun, kedua-dua atribut amat berkaitan dan oleh itu boleh dijadikan satu kumpulan. Jadikan atribut pengajar sebagai kunci primer dalam kumpulan baharu. Dalam kumpulan asal KURSUS, atribut pengajar menjadi kunci asing. Namakan kumpulan baharu sebagai PENGAJAR KURSUS.

```

UJIAN (nokp<KP> <KA>, kodKursus <KP> <KA>, tarikhUjian, keputusan)
KURSUS (kodKursus <KP>, kursus, pengajar, alamatPengajar)
PESERTA (nama, nokp <KP>, telefon, jantina, alamat)

```



```

UJIAN (nokp <KP> <KA>, kodKursus <KP> <KA>, tarikhUjian, keputusan)
KURSUS (kodKursus <KP>, kursus, pengajar <KA>)
PENGAJAR_KURSUS(pengajar <KP>, alamatPengajar)
PESERTA (nama, nokp <KP>, telefon, jantina, alamat)

```

Jika pengalihan ini tidak dilakukan, mungkin berlaku banyak pengulangan data bagi kedua-dua atribut pengajar dan alamat pengajar dalam jadual Kursus. Pengalihan dua atribut ini daripada kumpulan asal menjadikan skema hubungan dalam bentuk 3NF.

Pertimbangkan data-data dalam jadual 3NF. Adakah masih terdapat data-data yang berulang? Adakah terdapat atribut-atribut kunci dan asing untuk menjalinkan dua jadual yang asalnya tercantum?

Jadual 2.4.4 Jadual berstruktur 3NF bagi PESERTA

Nama	No. KP	Telefon	Jantina	Alamat
Abdul Salam	850415-02-9234	012-1111111	M	2, Seksyen 13, 40100 Shah Alam, Selangor
Adam Majid	900926-01-8577	012-2222222	M	10B, Cheras, 56000 Kuala Lumpur
Amy Tan	900426-02-5555	011-3333333	F	12, Taman Kenyalang, 93400 Kuching, Sarawak
Bryan Ng	870120-02-9352	012-4444444	M	55, Taman Botani, 31650 Ipoh, Perak

Nama	No. KP	Telefon	Jantina	Alamat
Charles Gunalan	950216-05-6552	010-5555555	M	18, Taman Tun Dr Ismail, 60000 Kuala Lumpur
Chin Xiao Ying	931211-07-1523	012-3333222	F	8, Taman SEA, SS2,47410 Petaling Jaya, Selangor
Dorothy Teng	890705-08-7755	016-3333566	F	3A, Damansara Utama, 47400 Petaling Jaya, Selangor
Edward Xavier	900502-03-5556	011-3244566	M	C3, Lorong Alor Akar, 25050 Kuantan, Pahang
Marissa O'Donnell	900422-02-5532	016-1133568	M	12-A, Bukit Kiara Residences, Sri Hartamas, 50490 Kuala Lumpur
Vivien F.	900413-02-3355	012-6678999	M	3, Meru Jaya, 31650 Ipoh, Perak

Jadual 2.4.5 Jadual 3NF UJIAN

Kod Kursus	Tarikh Ujian	Keputusan
4040	5-Mar-16	Lulus
3592	15-Apr-16	Lulus
4040	5-Mar-16	Lulus
3592	1-Mar-16	Lulus
3593	2-Mar-16	Lulus
4042	5-May-16	Lulus
4436	5-Jun-16	Gagal
4436	15-Aug-16	Lulus
3954	3-Mar-16	Lulus
4160	4-Mar-16	Lulus
4656	8-Mar-16	Lulus
4555	29-Feb-16	Lulus
4555	29-Feb-16	Lulus
4555	29-Feb-16	Lulus

Jadual 2.4.6 Jadual 3NF KURSUS

Kod Kursus	Kursus	Pengajar
3592	Juruteknik Komputer Bersijil	COMPUTE SYS
3593	Sekuriti Komputer	COMPUTE SYS
3954	Rangkaian Komputer	COMPUTE SYS
4040	Rangkaian Komputer	ACTIVE SOLUTIONS
4042	Teknologi Android	ACTIVE SOLUTIONS
4160	Pembangunan Perisian Microsoft	PEAK INNOVATION
4436	Pembangunan Rangkai Mobil Bersijil	ICON TRAINING
4555	Pengaturcaraan Java Bersijil	ICON TRAINING
4556	Komunikasi Bahasa Inggeris Perniagaan	CAMBRIDGE ENGLISH

Jadual 2.4.7 Jadual 3NF PENGAJAR_KURSUS

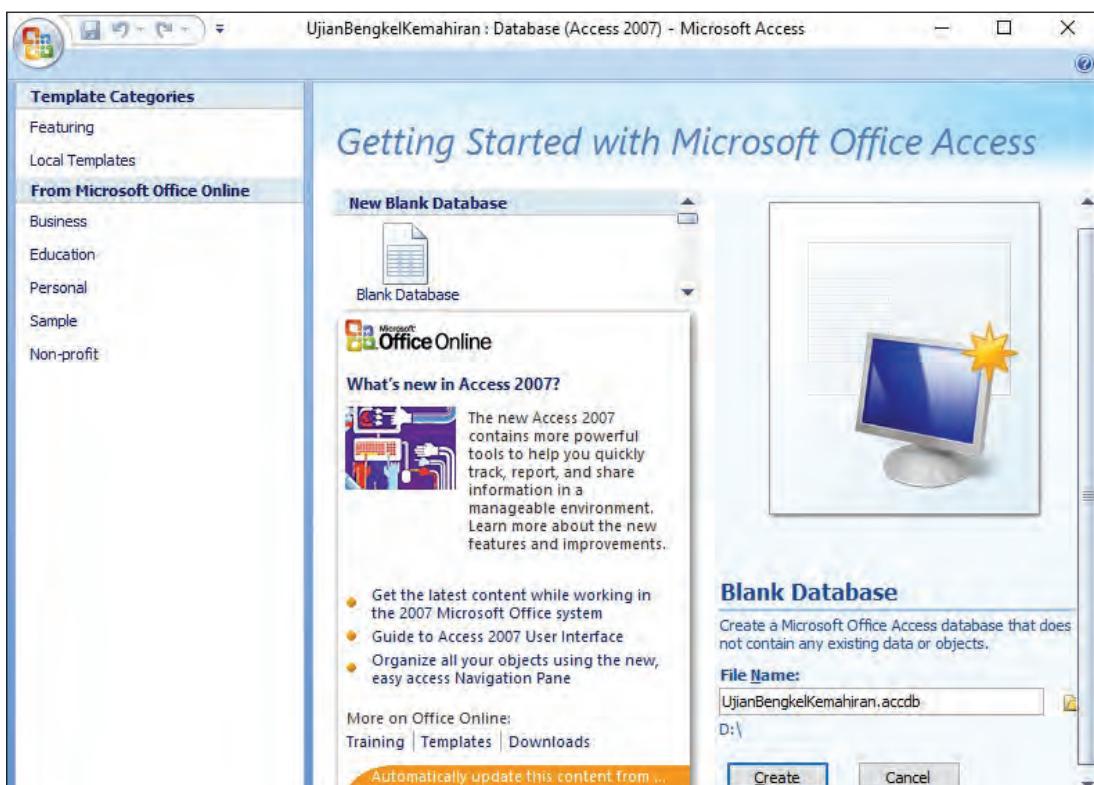
Pengajar	Alamat Pengajar
ACTIVE SOLUTIONS	Lot 453, Pavillion, 93400 Kuching, Sarawak
COMPUTE SYS	C-0806, Wisma Gg, 59200 Kuala Lumpur
ICON TRAINING	10, Mid Valley City, 59200 Kuala Lumpur
PEAK INNOVATION	Lot 1, Technologi Park MALAYSIA, 57000 Kuala Lumpur
CAMBRIDGE ENGLISH	9,Merchant Square, 47410 Petaling Jaya, Selangor

Kesemua jadual 3NF di atas tidak mempunyai baris data yang berulang dan mempunyai kebergantungan fungsi sepenuh kepada atribut kunci primer bagi jadual masing-masing. Jadual-jadual 3NF boleh dihubungkan kepada satu sama lain melalui atribut kunci primer dan asing.

2.4.2 Membangunkan Sistem Pangkalan Data dengan Antara Muka Bergrafik Menggunakan Perisian Pembangunan Sistem Pangkalan Data Mengikut SDLC Untuk Menyelesaikan Masalah

Jadual-jadual berstruktur 3NF kini boleh dilaksanakan dalam sistem pangkalan data. Gunakan *Microsoft Access* untuk menghasilkan jadual berstruktur 3NF di 2.4.1. Langkah-langkahnya ialah,

- 1 Bina pangkalan data, UjianBengkelKemahiran.



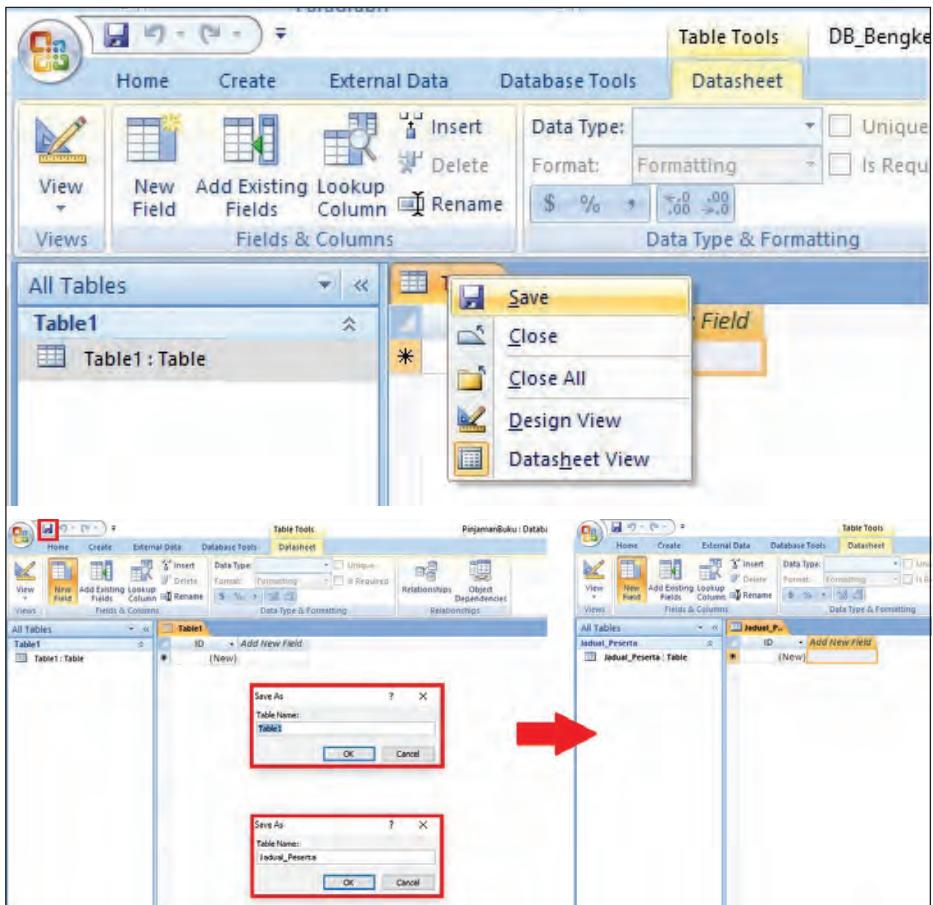
Rajah 2.4.4 Membina pangkalan data UjianBengkelKemahiran

- 2 Bina jadual Peserta:

- (a) Rujuk pada skema:

PESERTA (nama, nokp, telefon, jantina, alamat)

- (b) Cipta jadual baharu bernama PESERTA. Dari menu utama, pergi ke tab *Create* dan klik *Table*. Selepas itu, klik *Save* dan masukkan nama PESERTA.



Rajah 2.4.5 Membina jadual Peserta

(c) Tambahkan atribut-atribut ke dalam jadual. Pastikan jadual berada dalam mod *Design View*.

Field Name	Data Type
id_peserta	AutoNumber
nama	Text
NoKP	Text
telefon	Text
jantina	Text

Rajah 2.4.6 Atribut-atribut jadual Peserta

- (d) Tentukan kunci primer.
(e) Tentukan *properties* bagi setiap atribut.

Jadual 2.4.8 *Properties* untuk atribut-atribut jadual Ujian

Nama atribut	Jenis Data	Wajib?	Saiz	Deskripsi
nama	Teks	Ya	255	Mesti warga Malaysia.
nokp	Teks	Ya	14	Gunakan ungkapan <i>Like "#####-##-####"</i> dalam <i>Validation Rule</i> .
telefon	Teks	Ya	20	Gunakan input masuk 000-0000000
jantina	Teks	Tidak	10	Gunakan ungkapan "Lelaki" or "Wanita" sahaja dalam <i>Validation Rule</i> .
alamat	Teks	Ya	255	Alamat penuh peserta untuk surat-menjurat.

3 Bina jadual Ujian:

- (a) Rujuk pada skema:
UJIAN (*kodKursus*, *tarikhUjian*, *keputusan*)
(b) Cipta jadual baharu bernama UJIAN. Dari menu utama, pergi ke tab *Create* dan klik *Table*. Selepas itu, klik *Save* dan masukkan UJIAN sebagai nama jadual.
(c) Tambahkan atribut-atribut ke dalam jadual. Pastikan jadual berada dalam mod *Design View*.

Field Name	Data Type
NoKP	Number
kodKursus	Text
tarikhUjian	Date/Time
keputusan	Text

Rajah 2.4.7 Atribut-atribut jadual Ujian

- (d) Tentukan kunci primer. Untuk menghasilkan kunci komposit, pilih kedua-dua atribut dan klik ikon kunci dalam tab *Design* menu utama. Kedua-dua atribut dapat dipilih serentak dengan

menekan kunci "Ctrl" pada papan kekunci dan menggunakan tetikus untuk memilih kedua-dua baris atribut.

The screenshot shows the Microsoft Access ribbon with the 'Table Tools' tab selected, specifically the 'Design' tab. In the 'Views' section of the ribbon, the 'Primary Key' icon is highlighted with a red box. Below the ribbon, the 'All Tables' list shows 'Jadual_Peserta' and 'Ujian'. The 'Ujian' table is currently selected, and its design view is displayed on the right. The design view includes columns for 'Field Name' and 'Data Type'. The first two rows of the table ('NoKP' and 'kodKursus') have their respective icons highlighted with green boxes, indicating they are selected for being primary keys.

Rajah 2.4.8 Kunci primer daripada No.KP dan kodKursus

- (e) Tentukan *properties* bagi setiap atribut.

Jadual 2.4.9 *Properties* untuk atribut-atribut jadual Ujian

Nama atribut	Jenis Data	Wajib?	Saiz	Deskripsi
kodKursus	Teks	Ya	10	Kunci komposit dan kunci asing kepada jadual Kursus. Format #####. Gunakan <i>Lookup Wizard</i> untuk menyambung kepada atribut kodKursus dalam jadual Kursus.
tarikhUjian	Teks	Tidak	—	Dalam format bulan(b), hari(h) dan tahun(t) seperti bb/hh/tttt. Contoh: 8/31/2016
keputusan	Teks	Tidak	2	Gunakan ungkapan "Lulus", "Gagal" or "Tidak Hadir" dalam <i>Validation Rule</i> .

4 Bina jadual Kursus:

- (a) Rujuk pada skema:

KURSUS (kodKursus, kursus, pengajar)

- (b) Cipta jadual baharu bernama KURSUS. Dari menu utama, pergi ke tab *Create* dan klik *Table*. Selepas itu, klik *Save* dan masukkan KURSUS sebagai nama jadual.
- (c) Tambahkan atribut-atribut ke dalam jadual-jadual. Pastikan jadual berada dalam mod *Design View*.

The screenshot shows the Microsoft Access ribbon with the 'Table Tools' section selected, specifically the 'Design' tab. On the left, there's a list of tables: 'Jadual_Peserta', 'Ujian', and 'Kursus'. The 'Kursus' table is currently selected. The main area displays the table structure with three columns: 'Field Name' and 'Data Type'. The 'Field Name' column contains 'kodKursus', 'kursus', and 'pengajar', and the 'Data Type' column shows 'Text' for all three.

Rajah 2.4.9 Atribut-atribut jadual Kursus

- (d) Tentukan kunci primer.
- (e) Tentukan *properties* bagi setiap atribut.

Jadual 2.4.10 *Properties* untuk atribut-atribut jadual Kursus

Nama atribut	Jenis Data	Wajib?	Saiz	Deskripsi
KodKursus	Teks	Ya	5	Auto-nombor unik setiap kursus. Kunci primer. Format #####.
NamaKursus	Teks	Ya	100	Nama kemahiran yang dibekalkan.
Pengajar	Teks	Ya	100	Nama organisasi berdaftar yang membekalkan khidmat latihan.

5 Bina jadual PengajarKursus:

- (a) Rujuk pada skema:
`PENGAJAR_KURSUS(pengajar, alamatPengajar)`
- (b) Cipta jadual baharu bernama PengajarKursus. Dari menu utama, pergi ke tab *Create* dan klik *Table*. Selepas itu, klik *Save* dan masukkan PengajarKursus sebagai nama jadual.
- (c) Tambahkan atribut-atribut ke dalam jadual-jadual. Pastikan jadual berada dalam mod *Design View*.

The screenshot shows the Microsoft Access ribbon with 'Table Tools' selected and 'Design' highlighted. On the left, the 'Views' tab is active. In the center, there's a list of tables: Jadual_Peserta, Ujian, Kursus, and PengajarKursus. The PengajarKursus table is currently selected, showing its structure in a grid. The grid has columns for 'Field Name' and 'Data Type'. Two fields are listed: 'pengajar' (Text) and 'alamat pengajar' (Text). The 'alamat pengajar' field is highlighted with a yellow selection bar.

Field Name	Data Type
'pengajar'	Text
'alamat pengajar'	Text

Rajah 2.4.10 Atribut-atribut jadual PengajarKursus

- (d) Tentukan kunci primer.
- (e) Tentukan *properties* bagi setiap atribut.

Jadual 2.4.11 *Properties* untuk atribut-atribut jadual PenyediaKursus

Nama atribut	Jenis Data	Wajib?	Saiz	Deskripsi
'pengajar'	Teks	Ya	255	Nama organisasi berdaftar yang membekalkan khidmat latihan. Nama mestilah unik. Dijadikan kunci primer.
'alamatPengajar'	Teks	Ya	255	Alamat surat-menyurat organisasi yang menjadi pengajar kursus.

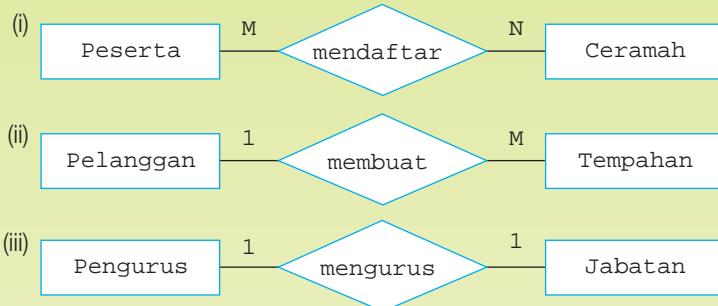
- 6 Paparkan hubungan antara jadual-jadual.
- 7 Paparkan jadual-jadual sebagai borang.
- 8 Masukkan data-data melalui borang.
- 9 Sediakan *query* untuk senarai semua peserta lulus.
- 10 Sediakan laporan untuk senarai semua peserta lulus.
- 11 Gunakan *switchboard* untuk menggabungkan borang-borang dan laporan-laporan dalam satu "antara muka".



Latihan Formatif 2.4

1 (a) Terangkan tujuan penggunaan Gambar Rajah Perhubungan Entiti (*Entity Relationship Diagram - ERD*).

(b) Jelaskan hubungan bagi setiap lukisan ERD yang ditunjukkan.



2 Lihat contoh jadual yang berikut:

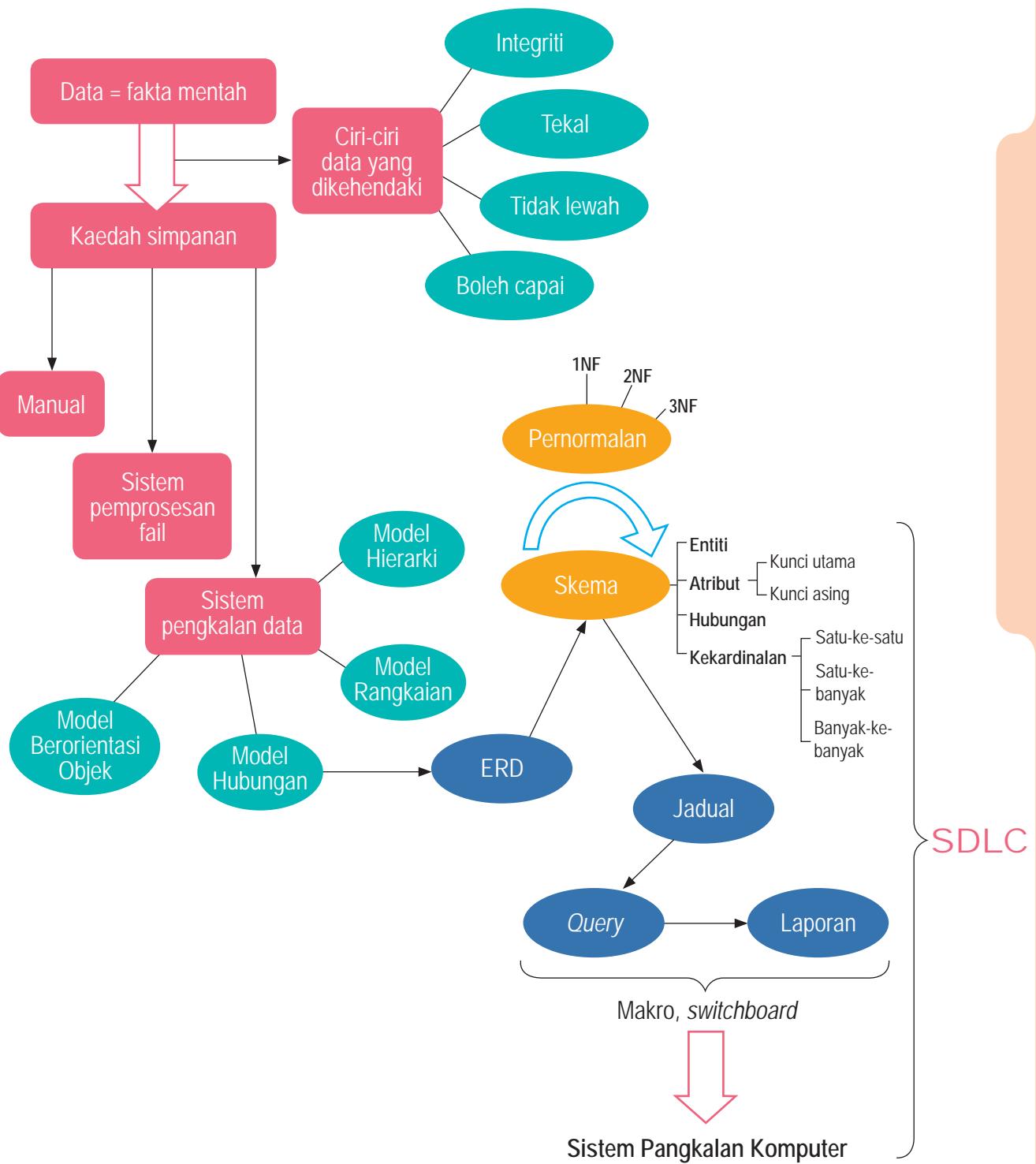
#Order	Tarikh	#Produk	Produk	Kuantiti	#Vendor	Vendor
1001	12/7/2016	605	Pen dakwat Pilot	30	200	Angkasa Sdn Bhd
1001	12/7/2016	606	Pen Angsa	200	300	Rerama Sdn Bhd
1002	13/7/2016	301	Kertas A4 Putih Unggul	30	250	Kinabalu Sdn Bhd
1003	13/7/2016	290	Krayon 12 batang Angsa	10	300	Rerama Sdn Bhd
1004	14/7/2016	288	Pensel warna 12 batang Pelangi	20	300	Rerama Sdn Bhd

Kenal pasti data-data unik yang sesuai sebagai calon kunci-kunci primer.

3 Merujuk soalan 2, hasilkan skema-skema hubungan dalam bentuk normal yang berikut:

- (i) 0NF
- (ii) 1NF
- (iii) 2NF
- (iv) 3NF

IMBAS SEMULA



LATIHAN SUMATIF

2

Soalan Objektif

Jawab **semua** soalan.

- 1 Yang berikut ialah kelebihan menggunakan pangkalan data berbanding senarai KECUALI _____
 - A lebih mudah untuk dibina dan disenggara.
 - B mencegah kelewahan.
 - C mudah berkongsi maklumat.
 - D mengekalkan integriti data.
 - E memudahkan penyusunan semula data.
- 2 _____ ialah kategori maklumat dalam sesuatu pangkalan data.
 - A Kunci primer
 - B Kunci asing
 - C Rekod
 - D Atribut
 - E Jadual
- 3 Suatu kumpulan rekod dikenali sebagai _____
 - A pangkalan data.
 - B kunci primer.
 - C kunci asing.
 - D atribut.
 - E jadual.
- 4 Atribut yang mempunyai nilai unik bagi setiap rekod dalam pangkalan data dikenali sebagai _____
 - A jenis data.
 - B kunci primer.
 - C kunci asing.
 - D atribut.
 - E jadual.
- 5 Pangkalan data yang lazimnya digunakan sekarang ialah _____
 - A senarai.
 - B pangkalan data hubungan.
 - C pangkalan data multimedia.
 - D pangkalan data berorientasi objek.
 - E pangkalan data hubungan objek.
- 6 Sistem pengurusan pangkalan data (DBMS) mempunyai operasi yang berikut KECUALI _____

- A memasukkan data.
- B melihat data.
- C carian data.
- D mengekstrapolasi data.
- E mengemas kini data.

- 7 Yang berikut ialah tatacara pengesahsahihan data KECUALI

- A semakan julat.
- B semakan kesempurnaan.
- C semakan ketekalan.
- D semakan berangka.
- E semakan penormalan.

- 8 _____ ialah model data yang digunakan untuk reka bentuk konsep pangkalan data dengan menunjukkan hubungan antara entiti.

- A Model Hubungan Entiti
- B Model data rangkaian
- C Model data hierarki
- D Model data rasional
- E Model data ternormal

- 9 Dalam sesetengah jadual, kunci primer terdiri lebih daripada satu atribut yang dikenali sebagai _____

- A kunci komposit.
- B kunci primer.
- C kunci asing.
- D lajur.
- E hubungan.

- 10 Yang manakah pernyataan yang BETUL bagi aktiviti-aktiviti seperti menambah, meminda dan memadamkan rekod daripada sesebuah fail?

- A pengesahsahihan
- B backup
- C semakan julat
- D penyenggaraan fail
- E penormalan



Soalan Struktur

Jawab **semua** soalan.

- 1 Pangkalan data boleh didefinisikan sebagai satu koleksi _____. []
- 2 Berikan satu contoh senario penggunaan pangkalan data. []
- 3 Bincangkan kelebihan-kelebihan penggunaan pangkalan data berbanding sistem fail biasa. []
- 4 Tentukan sama ada pernyataan yang berikut “betul” (✓) atau “salah” (✗):
 - (a) Entiti diwakili oleh jadual hubungan. []
 - (b) Entiti terbahagi kepada entiti utama dan entiti asing. []
 - (c) Kunci asing digunakan sebagai penghubung jadual-jadual. []
 - (d) Setiap entiti seharusnya mempunyai satu kunci primer. []
 - (e) Kunci asing mesti unik. []
 - (f) Fungsi pangkalan data termasuk mengumpulkan dan menyusun data semula. []
 - (g) Model data hubungan membolehkan hubungan **banyak-ke-banyak**. []
 - (h) Penormalan ialah proses merekodkan data sekali sahaja untuk mengurangkan data berulang. []
- 5 Isi tempat kosong.
 - (a) Semasa membina jadual pangkalan data, adalah penting untuk memastikan bahawa setiap rekod adalah _____. []
 - (b) Atribut unik yang terkandung dalam setiap rekod dikenali sebagai _____. []
 - (c) Set program yang mengendalikan tanggungjawab pangkalan data sesebuah organisasi dikenali sebagai _____. []
 - (d) Angka seperti 123 boleh disimpan di dalam jenis data _____ atau jenis data _____ jika tiada pengiraan diperlukan. []
- 6 (a) Apakah yang dimaksudkan dengan entiti?
(b) Bezakan atribut dengan entiti menggunakan entiti contoh yang sesuai, bezakan antara atribut dengan entiti.
(c) Bezakan kunci primer dengan kunci asing.
(d) Apakah perbezaan antara kunci primer dengan medan bukan-kunci?
- 7 Terangkan perbezaan antara sistem fail biasa dengan sistem pangkalan data.

Sistem fail biasa	Pangkalan data

- 8 (a) Apakah tujuan penormalan data?
(b) Apakah ciri-ciri jadual Tidak-Ternormal (0NF)?
(c) Bincangkan perbezaan antara terma-terma yang berikut:

Bentuk Normal	Huraian
1NF	
2NF	
3NF	

9 Kenal pasti kekardinalan dalam contoh-contoh yang berikut.

Contoh	Kekardinalan
<pre> graph LR Pengurus[Pengurus] -- 1 --> mengurus{mengurus} mengurus -- 1 --> Jabatan[Jabatan] </pre>	
<pre> graph LR Pelanggan[Pelanggan] -- 1 --> membuat{membuat} membuat -- M --> Tempahan[Tempahan] </pre>	
<pre> graph LR Peserta[Peserta] -- M --> mendaftar{mendaftar} mendaftar -- N --> Ceramah[Ceramah] </pre>	

10 Lukisan ERD untuk senario yang berikut:

- Seorang guru boleh mengajar lebih daripada satu subjek tetapi setiap subjek boleh diajar oleh seorang guru sahaja.
- Seorang pesakit boleh dirujuk kepada satu atau lebih doktor pakar. Seorang doktor pakar merawat banyak pesakit.
- Klinik 1Malaysia mempunyai seorang doktor pelawat sahaja. Doktor tersebut mungkin berkhidmat di klinik-klinik 1Malaysia yang lain pada hari yang berlainan.
- Nadia memohon pasport antarabangsa kerana dia ingin melancong ke Pulau Bali. Setiap pemohon hanya boleh memiliki satu pasport sahaja. Setiap pasport adalah khusus untuk seorang pemohon.

Penyelesaian Masalah

Jawab **semua** soalan.

- Syarikat ABC mempunyai satu fail dengan 30 atribut data dan lebih 1000 rekod. Syarikat ABC perlu selalu mengemaskinikan data untuk perniagaannya.
 - Apakah kekurangan menggunakan sistem fail untuk tugas tersebut?
 - Apakah kelebihan menggunakan pangkalan data untuk tujuan di atas (jika ada)?
- Anda bekerja di sebuah kedai alat tulis di tempat anda semasa cuti semester. Majikan anda merancang untuk membangunkan satu sistem pangkalan data untuk merekodkan semua transaksi. Majikan anda meminta pendapat anda mengenai reka bentuk pangkalan data.
 - Terangkan kepada majikan anda faedah penggunaan Gambar Rajah Perhubungan Entiti (ERD) untuk merancang dan mendokumentkan reka bentuk.
 - Reka satu ERD yang sesuai yang menunjukkan seorang pelanggan membeli satu barang yang mengandungi atribut seperti jenis barang, ID barang, kuantiti dan harga. ERD tersebut hendaklah juga menyimpan tarikh dan masa transaksi serta nama juruwang.
- Bincian pada tahun 2010 menunjukkan bahawa negeri Selangor mempunyai populasi penduduk yang tertinggi di Malaysia. Jadual yang berikut menunjukkan populasi_selangor, menempatkan bandar-bandar utama di Selangor mengikut populasi:



Negeri	Bandar	Tahun	Populasi
Perak	Ipooh	2010	757,892
	Taiping	2010	245,182
Pulau Pinang	George Town	2010	740,200
Selangor	Subang Jaya	2010	1,553,589
	Klang	2010	1,113,851
	Shah Alam	2010	671,282
	Cheras	2010	601,534
	Kajang	2010	448,000
Johor	Johor Bahru	2010	497,067

[Nota: Statistik di atas ialah contoh dan bukan nilai sebenar.]

- Ubah jadual di atas kepada skema hubungan.
- Ubah skema tersebut kepada 1NF.
- Ubah skema tersebut kepada 2NF.

4 Kaji senario yang berikut:



“Seorang pemain bola di peringkat Majlis Sukan-Sukan Malaysia (MSSM) mesti mempunyai butiran seperti nama penuh, nombor kad pengenalan dan nombor jersi. Pemain bola juga ialah salah seorang daripada ahli pasukan. Satu pasukan bola mestilah mempunyai nama, nama sekolah, negeri diwakili, nama jurulatih dan nama guru pengiring. Bilangan maksimum pemain bola ialah 15 orang bagi satu pasukan. Pasukan bola ini akan menyertai siri perlawanan. Setiap perlawanan cuma membenarkan dua pasukan bola untuk bersaing. Butiran perlawanan, ialah tempat, tarikh dan masa.”

- Lukiskan ERD untuk senario di atas.
- Ubah model data di atas kepada skema hubungan. Tentukan kunci primer dan kunci asing bagi setiap kunci hubungan.
- Bolehkah kunci primer ditambahkan kepada suatu entiti? Merujuk senario di atas, cadangkan keperluan untuk kunci primer tambahan.

5 Katakan anda diberikan skema ONF (#Order, Tarikh, #Produk, Produk, Kuantiti, #Vendor, Vendor) dan sampel jadual data yang berikut. Jawab soalan yang diberikan.



#Order	Tarikh	#Produk	Produk	Kuantiti	#Vendor	Vendor
1001	12/7/2016	605	Pen dakwat Juruterbang	30	200	Angkasa Sdn Bhd
1001	12/7/2016	606	Pen Angsa	200	300	Rerama Sdn Bhd
1002	13/7/2016	301	Kertas A4 Putih Unggul	30	250	Kinabalu Sdn Bhd
1003	13/7/2016	290	Krayon 12 batang Angsa	10	300	Rerama Sdn Bhd
1004	14/7/2016	288	Pensel warna 12 batang Pelangi	20	300	Rerama Sdn Bhd

- Kenal pasti kumpulan data yang berulang.
- Ubah skema ONF kepada 1NF. Jelaskan sebab anda berbuat demikian.
- Ubah skema 1NF kepada 2NF. Jelaskan sebab anda berbuat demikian.
- Ubah skema 2NF kepada 3NF. Jelaskan sebab anda berbuat demikian.
- Lakarkan jadual-jadual berserta data-data dalam 3NF.

BAB 3

INTERAKSI MANUSIA DAN KOMPUTER

Interaksi ialah tindakan atau perhubungan aktif antara satu dengan yang lain. Pada era globalisasi ini, manusia dan komputer saling memerlukan di samping menjadi medium utama dalam pembangunan dan perkembangan pelbagai bidang.

Interaksi manusia dan komputer melibatkan kajian tentang cara teknologi komputer mempengaruhi kerja dan aktiviti manusia.

Dalam bab ini, anda akan mempelajari:

- » Keperluan interaksi manusia dan komputer
- » Cara untuk menilai produk interaktif berdasarkan tujuan dan prinsip asas reka bentuk interaktif
- » Aplikasi proses reka bentuk interaksi dalam atur cara yang dibangunkan
- » Cara untuk menghasilkan prototaip paparan dan reka bentuk skrin
- » Laporan penilaian kuantitatif terhadap paparan dan reka bentuk skrin

Fikirkan...

- Apakah kepentingan komputer kepada manusia?
- Apakah yang akan berlaku jika komputer tidak mempunyai reka bentuk interaksi yang menarik dan tidak mesra pengguna?
- Bagaimanakah kita boleh menghasilkan prototaip yang menarik dan dapat memuaskan hati pengguna?

STANDARD PEMBELAJARAN

3.1.1 Mengkaji keperluan interaksi antara manusia dan komputer

3.1.2 Menilai produk interaktif berdasarkan tujuan dan prinsip asas reka bentuk interaktif

Reka Bentuk Interaksi

Reka bentuk interaksi merupakan satu aspek yang penting dan perlu dititikberatkan semasa membangunkan sesuatu produk atau aplikasi kerana reka bentuk yang menarik dan mudah digunakan akan membuatkan pengguna menjadi selesa dan seronok untuk menggunakan produk atau sistem tersebut. Reka bentuk ialah susun atur teks, gambar, butang dan menu dalam sesuatu produk. Manakala interaksi pula ialah tindakan manusia semasa menggunakan produk tersebut.

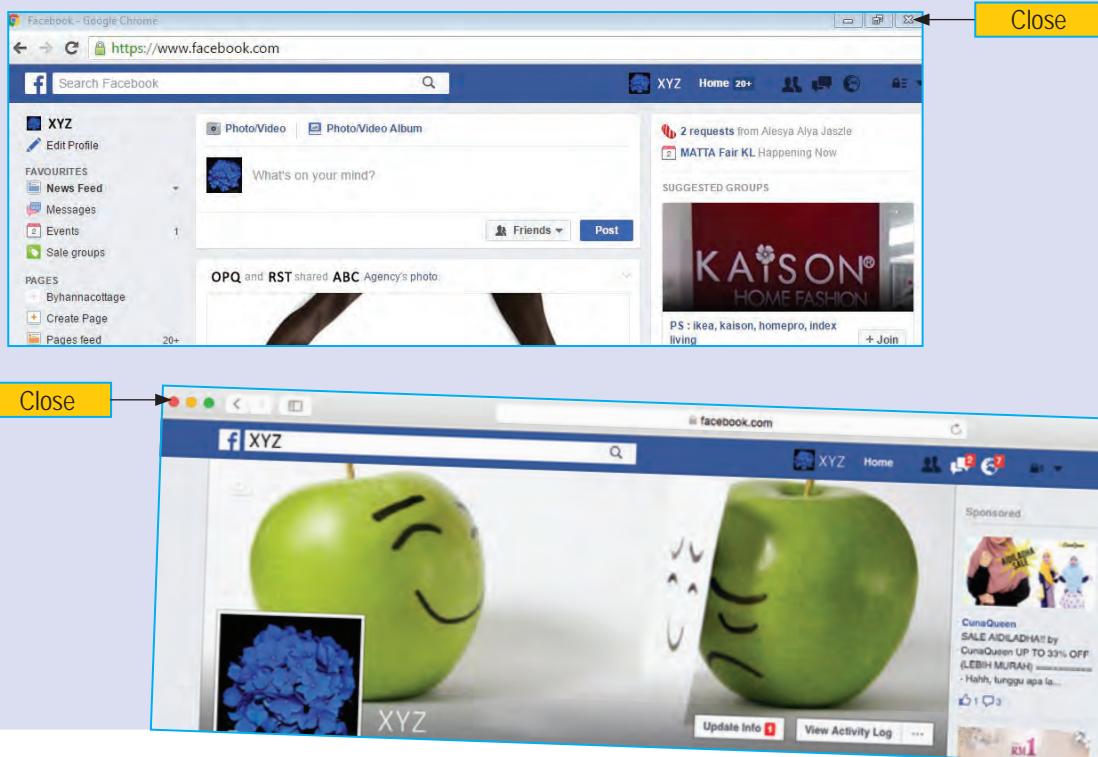
Reka bentuk interaksi membenarkan pengguna untuk berkomunikasi dan berinteraksi dengan produk atau aplikasi. Reka bentuk perlu mengambil kira penerimaan pengguna yang terdiri daripada pelbagai umur dan persekitaran. Reka bentuk dan paparan yang mudah dan menarik akan membuatkan ramai pengguna ingin menggunakan dalam kehidupan seharian. Sebagai contoh, penggunaan *WhatsApp*, *Facebook*, *Instagram*, *Telegram*, sistem penempahan tiket kapal terbang dan sebagainya. Paparan dan menu yang disediakan sangat ringkas dan mudah untuk digunakan oleh pelbagai lapisan masyarakat.

Bagi menghasilkan reka bentuk interaksi yang baik, terdapat beberapa prinsip yang perlu dipatuhi. Setiap produk atau aplikasi perlu memenuhi kepentingan prinsip ini supaya produk yang dihasilkan dapat memenuhi matlamat kebolehgunaan iaitu pengguna merasa mudah untuk belajar, memberi kesan yang baik apabila menggunakannya dan memberikan satu kepuasan dan pengalaman yang menyeronokkan. Salah satu contoh aplikasi yang mencapai tahap kebolehgunaan ini ialah *Instagram* di mana pengguna mudah untuk menggunakan menu-menu yang disediakan, membolehkan pengguna berkomunikasi dan mengikuti perkembangan sesuatu isu dan berasa gembira dan puas apabila selesai menggunakannya.



Oleh itu, prinsip reka bentuk interaksi sangat penting dan perlu diberi perhatian kerana aplikasi yang dibangunkan sangat kompleks. Lima prinsip reka bentuk yang perlu dititikberatkan ialah:

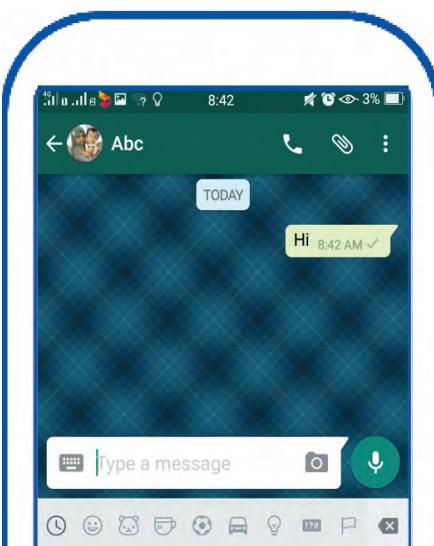
1 Konsistensi (consistency)—Sebarang perubahan yang berlaku pada paparan akan menarik perhatian pengguna atau mengganggu pengalaman pengguna. Semua elemen perlu kekal pada kedudukan yang sama supaya pengguna akan berasa selesa semasa menggunakan aplikasi. Jika elemen-elemen tersebut dialihkan tanpa sebab dan tujuan, fokus pengguna akan terganggu. Reka bentuk perlu konsisten dari segi perseimbahan dan fungsi pada semua antara muka. Sebagai contoh, kedudukan butang *Close* pada antara muka *Microsoft Windows* adalah di penjuru kanan atas. Pengguna *Microsoft Windows* yang baharu menggunakan *Apple MacBook* pasti berasa kekok kerana kedudukan butang *Close* pada antara muka *MacBook* adalah di penjuru kiri atas.



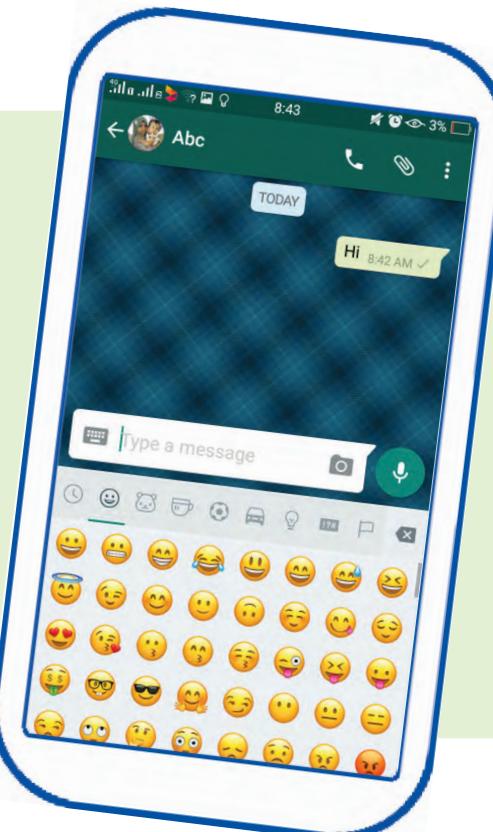
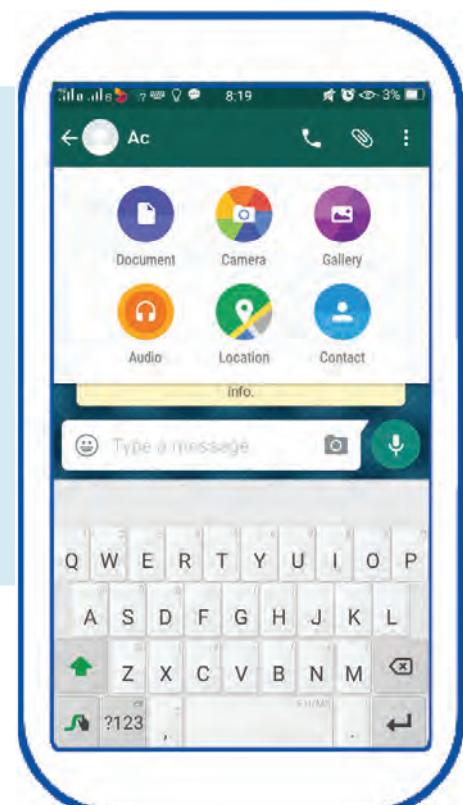
2 Kebolehan membuat pemerhatian (perceivability)—Pengguna yang menggunakan aplikasi buat kali pertama akan memerhatikan segala aspek antara muka. Penunjuk atau butang interaksi perlulah mudah dikenal pasti. Jika penunjuk/butang interaksi sukar dikesan, pengguna akan berasa tidak selesa. Pengalaman pengguna yang terbaik adalah apabila pengguna boleh menggunakan aplikasi dengan selesa dan tanpa ragu-ragu. Sebagai contoh, antara muka *Facebook* telah dikemas kini beberapa kali sejak dilancarkan. Pengguna pasti memerlukan masa untuk mengenal pasti kedudukan butang interaksi yang baharu setiap kali antara muka ini dikemas kini.



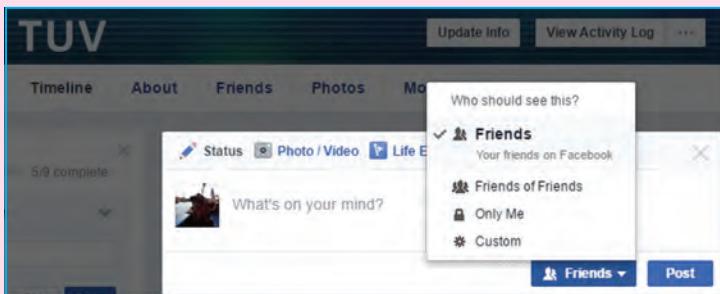
3 Boleh dipelajari (*learnability*) — Secara amnya, reka bentuk interaksi sepatutnya mudah untuk dipelajari, dan mudah untuk diingati. Logiknya, pengguna hanya menggunakan antara muka tersebut sekali, tetapi pada masa yang sama, mereka mempelajarinya dan mengingati apa-apa yang dipelajari untuk selama-lamanya. Contohnya aplikasi *WhatsApp*. Pengguna hanya perlu menggunakan aplikasi tersebut sekali untuk belajar cara aplikasi tersebut beroperasi. Pengguna akan mengingati cara untuk menggunakan sesuatu aplikasi dengan hanya sekali menggunakan aplikasi tersebut.



4 Kebolehan untuk menjangka (*predictability*) — Reka bentuk interaksi yang baik dan berkesan dapat membuatkan pengguna membuat jangkaan perkara yang akan berlaku dalam aliran proses aplikasi tersebut sebelum pengguna menggunakan aplikasi tersebut. Sebagai contoh, pengguna dapat meneka apa yang akan terpapar apabila menekan ikon emosi pada aplikasi *WhatsApp*.



5 Maklum balas (feedback)—Maklum balas boleh memberikan gambaran sebenar mengenai reka bentuk interaksi dan maklumat mengenai hasil reka bentuk interaksi tersebut. Pereka bentuk menggunakan maklum balas untuk melakukan penambahbaikan pada reka bentuk interaksi yang telah dihasilkan. Sediakan maklum balas sekiranya pengguna memerlukannya. Maklum balas yang diberikan sepatutnya bermakna. Kegagalan untuk memberikan maklum balas yang dikehendaki, boleh menyebabkan proses pengulangan yang tidak sepatutnya terhadap sebarang tindakan, kesalahan dan ralat. Sebagai contoh, apabila pengguna menekan butang *public* dalam program *Facebook*, program akan memberikan maklum balas dengan memaparkan beberapa pilihan seperti di bawah. Cuba bayangkan kalau pengguna menekan butang *public* tetapi program tidak memaparkan apa-apa? Contoh maklum balas yang lain seperti '*loading*'. Program menunjukkannya sedang melaksanakan arahan yang diperlukan oleh pengguna.



Kelima-lima prinsip ini bertindak bersama-sama dalam sesebuah sistem. Prinsip reka bentuk telah dipenuhi dengan jayanya apabila pengguna berupaya untuk terus memahami dan menggunakan antara muka dengan mudah, efisien dan selesa serta kandungan aplikasi dapat disampaikan dengan berkesan.



44

Prinsip Reka bentuk Interaksi

Kerja Individu

Arissa dan Hui Ling bercadang untuk melawat rakan mereka iaitu Amani di Kuching, Sarawak sempena menyambut majlis hari jadi Amani. Mereka memilih untuk menaiki kapal terbang Mas AirLine (MAS) untuk pergi ke Kuching. Jadi, mereka membuka laman sesawang www.masairline.com.my untuk menempah tiket secara atas talian.

Berdasarkan situasi di atas, kenal pasti ciri-ciri kebolehgunaan laman sesawang MAS dan bincangkan lima prinsip reka bentuk interaksi

3.1.1 Keperluan Interaksi antara Manusia dengan Komputer

Semasa komputer mula-mula dibangunkan, pakar terlatih sahaja yang berupaya menggunakan komputer. Pada masa itu, komputer bersaiz besar dan mahal. Komputer tersebut hanya boleh diperoleh dalam bidang industri dan penyelidikan. Kini, komputer bersaiz kecil dan mudah didapati serta menjadi peranti penting yang digunakan dalam pelbagai bidang. Majoriti pengguna komputer pada masa kini tidak menerima latihan intensif, tidak seperti 30 tahun yang lalu.

Hal ini menunjukkan bahawa interaksi antara manusia dengan komputer semakin mudah berlaku. Interaksi antara manusia dengan komputer merupakan proses, dialog atau kegiatan yang dilakukan oleh manusia untuk berinteraksi dengan komputer. Interaksi antara manusia dengan komputer perlu bersifat jelas, intuitif dan semula jadi supaya sesuai digunakan oleh pengguna daripada pelbagai lapisan masyarakat. Interaksi antara manusia dengan komputer mengambil aspek kehidupan sehari-hari dan menggunakan aspek tersebut dalam perancangan membangunkan perisian serta peranti. Sebagai contoh, ikon *folder* pada "antara muka" komputer membolehkan pengguna terus memahami bahawa kita boleh menyimpan dokumen digital di dalam *folder*, sama seperti *folder* di pejabat yang menjadi tempat kita menyimpan dokumen kertas.

Terdapat beberapa sebab interaksi antara manusia dengan komputer diperlukan, iaitu:

- 1 Mendapat permintaan dalam pasaran**—Pengguna mahu membeli atau menggunakan produk yang mempunyai kebolehgunaan yang tinggi, selamat, berguna, berfungsi dan dalam jangka masa yang panjang, lebih membantu dan menyeronokkan pengguna dalam kehidupan sehari-hari.
- 2 Meningkatkan produktiviti**—Berkeupayaan untuk menghasilkan output dari satu set input yang diberikan lebih cepat dengan menggunakan sumber pekerja, teknologi dan sistem secara optimum dan menukar input kepada output yang berguna.
- 3 Mengurangkan kos selepas jualan**—Jika produk tidak berfungsi dengan sempurna, pengguna akan menghantar produk untuk dibaiki. Hal ini akan meningkatkan kos selepas jualan.
- 4 Mengurangkan kos pembangunan**—Elakkan daripada melaksanakan ciri-ciri yang tidak disukai pengguna dan produk yang tidak cekap. Jika kedua-dua perkara ini dielakkan, kos pembangunan dapat dijimatkan.



Tetikus ialah peranti kawalan komputer yang penting. Prototaip tetikus dihasilkan pada tahun 1964 dan diperbuat daripada kayu yang mempunyai sebuah butang pada bahagian atas. Dua buah roda besi pada bahagian bawah tetikus akan merekodkan pergerakan paksi X dan Y yang kemudiannya akan menggerakkan anak panah pada paparan komputer. Untuk melihat gambar tetikus tersebut, boleh layari laman sesawang di bawah.



The First Mouse
<http://www.goo.gl/1HClal>

- 5 Mengembangkan aktiviti dan menambahkan pengalaman manusia**—Infrastruktur teknologi bagi satu kawasan yang baharu akan dibangunkan untuk memberikan peluang kepada manusia yang ingin mengembangkan aktiviti dan menambahkan pengalaman dalam kehidupan mereka.
- 6 Pengkomputeran sosial**—Arus perkembangan laman sesawang menjana persekitaran interaktif sosial untuk mewujudkan peluang berkomunikasi atau perhubungan sosial antara manusia. Manusia tidak akan berinteraksi dengan mesin atau komputer sahaja, malahan manusia akan berinteraksi dengan manusia yang lain melalui penggunaan komputer.



Inovasi Sains
Komputer



goo.gl/THbgO8
Video ini menunjukkan perbezaan hubungan interaksi manusia dengan komputer pada zaman dahulu, masa kini dan masa akan datang.

3.1.2 Menilai Produk Interaktif Berdasarkan Tujuan dan Prinsip Asas Reka Bentuk Interaktif

Setiap produk yang dibangunkan perlu dinilai supaya produk itu dapat memberi kepuasan kepada pengguna. Sebagai pengguna yang bijaksana pula, pengguna haruslah mengetahui bagaimana untuk menilai sesuatu produk interaktif. Banyak kaedah yang boleh digunakan untuk menilai sesuatu produk interaktif seperti kaedah temu bual, pemerhatian, soal selidik, perbincangan ahli kumpulan, refleksi dan sebagainya.

Antara produk interaktif yang boleh diambil untuk dinilai ialah *Facebook*. Sebelum menilai produk ini, pengguna perlu tahu apakah tujuan dan keperluan bagi menggunakan laman *Facebook* ini. Antara tujuan utama menggunakan laman ini adalah untuk berkomunikasi dengan rakan yang baharu dan rakan yang lama dari seluruh dunia serta berkongsi pelbagai perkara dan isu. Di samping itu, ada juga pengguna yang suka untuk menggunakan laman *Facebook* bagi mendapatkan populariti, menyebarkan pengaruh positif atau negatif, memberikan tip dan ilmu pengetahuan dan sebagainya yang bergantung kepada niat pengguna itu sendiri.

Berdasarkan pada laman *Facebook*, salah satu kaedah penilaian yang boleh digunakan ialah dengan kaedah soal selidik di mana pengguna boleh menilai kebolehgunaan produk tersebut berdasarkan prinsip asas reka bentuk seperti Jadual 3.1.

Jadual 3.1 Contoh Borang Soal Selidik untuk Menilai Kebolehgunaan Produk

Nama Aplikasi / Laman sesawang/ Sistem / Produk yang dinilai	<i>Facebook</i>		
Kategori Aplikasi / Laman sesawang / Sistem / Produk	<i>Social Networking</i>		
Platform	<i>iOS / Windows / PC / Android ...</i>		
Versi Aplikasi / Laman sesawang / Sistem / Produk (*jika ada)			
Fungsi Utama Aplikasi / Laman sesawang / Sistem / Produk			
Tarikh			
Prinsip Asas	Penilaian	Tandakan (/) jika Ya dan (X) jika Tidak	
		Ya (/)	Tidak (X)
Konsisten	Adakah butang navigasi sentiasa berada di sebelah kiri?		
	Adakah laman web ini kerap menambah butang navigasi yang baru?		
	Jika tetapan bahasa ditukar, adakah semua butang navigasi bertukar mengikut bahasa yang dipilih?		
	Butang-butang navigasi diletakkan dalam satu kumpulan pada paparan aplikasi.		
Kebolehan membuat pemerhatian	Nama dan ikon yang digunakan pada butang amat mudah untuk difahami.		
	Saya masih boleh menggunakan <i>Facebook</i> walaupun menu diletakkan di sebelah atas.		
	Saya tidak dapat memastikan akaun <i>Facebook</i> itu adalah pemilik akaun sebenar atau tidak terutamanya yang melibatkan akaun selebriti.		
	Saya menggunakan kesemua butang navigasi yang disediakan setiap kali saya menggunakan aplikasi ini.		
Boleh dipelajari	Saya mudah memahami aplikasi ini dan tidak perlu diajar banyak kali untuk menggunakan.		
	Tetapan aplikasi ini membenarkan saya mengubahnya mengikut kehendak saya.		
	Saya memahami semua fungsi butang dan menu yang ditunjukkan dalam aplikasi ini.		
	Semakin hari saya semakin cekap menggunakan aplikasi ini.		
Kebolehan untuk menjangka	Apabila menekan apa-apa butang, saya tahu apa yang akan dipaparkan selepas itu.		
	Aplikasi ini membawa saya ke halaman yang tepat mengikut kehendak saya.		
	Sesetengah butang navigasi membuat saya tertanya-tanya, apakah yang akan dipaparkan jika saya menekan butang itu.		
	Kadang-kadang saya berasa marah dengan aplikasi ini kerana iklan yang tidak diperlukan dipaparkan juga.		
Maklum balas	Aplikasi ini membuat carian apabila saya memasukkan nama pada ruang search .		
	Aplikasi ini menyenaraikan tempat yang ada berhampiran dengan saya sewaktu menekan butang nearby places .		
	Saya tidak boleh melihat gambar rakan yang belum menerima permohonan saya untuk menjadi rakannya.		
	Sesetengah navigasi mengambil masa yang lama untuk menghasilkan paparan.		

Soalan tamat.
Sekian, terima kasih

Selepas pengguna mengisi borang soal selidik ini, satu analisis perlu dibuat bagi mengkaji prinsip manakah yang paling dipersetujui (ditandakan “YA”) dan prinsip manakah yang kurang berkesan (ditandakan dengan “TIDAK”). Kemudian, buat penyataan rumusan daripada analisis tersebut.

Sebagai seorang penilai yang bagus, penilai seharusnya mengetahui bagaimana ingin membina sebuah instrumen penilaian terutamanya soal selidik. Sebelum membina soal selidik, penilai perlulah mengetahui dahulu mengenai konsep dan bentuk instrumen yang ingin dihasilkan. Untuk menghasilkan satu soal selidik, beberapa perkara perlu diambil kira iaitu cara membina soalan dan membentuk pilihan jawapan.

Pilihan jawapan terbahagi kepada dua jenis pilihan iaitu:

- 1 Bentuk **Skala Likert** (1932). Bentuk ini digunakan bertujuan untuk melihat sejauh mana persetujuan pengguna terhadap produk yang dibangunkan.

Bulatkan pilihan jawapan anda berdasarkan skala yang berikut:

5 = Amat Setuju

4 = Setuju

3 = Tidak Setuju

2 = Sangat Tidak Setuju

1 = Tidak Pasti

Item	Soalan	Skala				
Konsistensi						
1.		5	4	3	2	1
2.		5	4	3	2	1
3.		5	4	3	2	1
4.		5	4	3	2	1
Maklum balas		Skala				
		5	4	3	2	1
		5	4	3	2	1

- 2 Bentuk **Skala Guttman** (1944). Bentuk soal selidik ini bertujuan untuk mendapatkan data yang jelas, tegas, dan konsisten.

Item	Soalan	Tandakan (/) jika Ya dan (X) jika Tidak	
		Ya (/)	Tidak (X)
Konsisten			
Maklum balas			

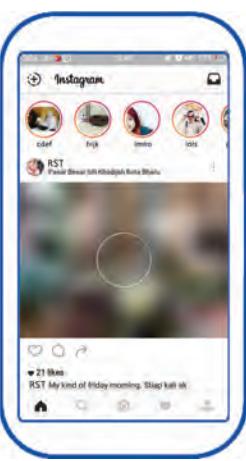
Selepas menentukan skala yang ingin digunakan, penilai bolehlah membangunkan soalan di dalam soal selidik itu. Bina sekurang-kurangnya lima soalan untuk satu kriteria yang diukur. Soalan yang dibina haruslah meliputi nilai positif dan negatif sesuatu produk yang dibangunkan bagi mendapatkan data pengguna yang mempunyai pelbagai penerimaan terhadap produk yang dibangunkan.

Soal selidik pula mestilah mempunyai ciri-ciri yang baik seperti ruang pilihan jawapan yang jelas dan mudah untuk pengguna menanda, soalan yang sesuai dengan responden dan item yang jelas dan tidak berat sebelah. Selain itu, cuba elakkan soalan yang panjang lebar dan soalan yang boleh menampakkan keburukan produk itu sendiri. Setelah selesai soal selidik, jangan lupa untuk menyatakan soalan telah tamat dan memberikan ucapan terima kasih sebagai penghargaan di atas kesudian responden untuk menjawab soal selidik.



Kerja Berkumpulan

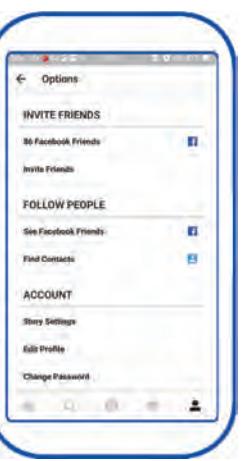
45
Aplikasi Instagram



Paparan Instagram



Paparan memilih gambar dan video



Paparan Setting

Instagram adalah satu produk interaktif yang sedang digunakan pada masa kini. Berdasarkan pemerhatian anda sebagai pengguna yang menggunakan aplikasi *Instagram*, bincangkan tiga soalan di bawah:

(Catatan: Jika perbincangan aplikasi dalam aktiviti ini tidak sesuai digunakan untuk Pengajaran dan Pembelajaran (PdP), guru dan murid boleh memilih sebarang aplikasi lain yang lebih sesuai.)

1 Senaraikan tiga aktiviti interaksi yang boleh dilakukan oleh pengguna terhadap perisian ini yang boleh digunakan.

	Interaktiviti	Penerangan
Contoh	Mengemas kini status	Pengguna boleh mengemas kini status dengan menaip teks, memuat naik audio, video dan foto.
(a)		
(b)		
(c)		



BAB 3 INTERAKSI MANUSIA DAN KOMPUTER



- 2** Membuat penilaian dari segi lima prinsip reka bentuk interaksi. Gunakan jadual di bawah sebagai panduan.

	Prinsip	Penjelasan pemerhatian yang dibuat pada aplikasi <i>Instagram</i>
(a)		
(b)		
(c)		
(d)		
(e)		

- 3** Cadangkan penambahbaikan (perkara yang sedia ada untuk diolah menjadi lebih baik) atau fungsi baharu (perkara yang belum ada tetapi difikirkan sangat perlu dan bermanfaat jika ditambah) yang anda rasa perlu / penting agar aplikasi ini lebih berkesan dan mudah digunakan. Berikan justifikasi anda bagi cadangan tersebut.

Latihan Formatif 3.1

- 1** Pilih salah satu aplikasi interaksi manusia dan komputer di bawah. Kemudian, bincangkan dari segi prinsip asas reka bentuk interaktif yang digunakan dalam aplikasi yang dipilih.

- Perisian menempah tiket kapal terbang
- Perisian membeli tiket wayang

- 2** Padankan pernyataan di bawah dengan prinsip reka bentuk yang betul.

Penyataan	Prinsip
Menggunakan sesuatu aplikasi tanpa sebarang kekeliruan	
Mudah untuk dipelajari dan mudah untuk diingati	
Semua elemen kekal pada kedudukan yang sama	
Bertujuan untuk melakukan penambahbaikan pada reka bentuk interaksi yang telah dihasilkan	
Membuatkan pengguna dapat membuat jangkaan atau anggapan perkara yang akan berlaku	

- 3** Teliti situasi di bawah dan jawab soalan yang berikutnya.

Danial ingin mengeluarkan wang menggunakan mesin ATM. Selepas memasukkan kad dan nombor pin, Danial memasukkan nilai wang yang diperlukan. Selepas beberapa minit, tiada wang dikeluarkan daripada mesin. Paparan mesin pula tidak menunjukkan sebarang mesej.

Berdasarkan situasi di atas, mesin tidak melaksanakan tugas dan tidak memberikan tindak balas kepada permintaan Danial. Oleh itu, kenal pasti beberapa masalah dan prinsip yang tidak dipatuhi oleh mesin ATM di atas. Berikan cadangan penambahbaikan supaya masalah ini tidak berulang lagi. Jawapan anda hendaklah diisi di dalam jadual di bawah.

Prinsip yang berkaitan

Isu / Masalah yang dikenal pasti

Cadangan penambahbaikan

-  **4** Dengan menggunakan peta *i-Think* dan contoh-contoh yang sesuai, jelaskan kepentingan prinsip asas reka bentuk dalam pembangunan aplikasi. Anda boleh jelaskan dengan contoh-contoh yang sesuai.
-  **5** Guru Sains Komputer anda memberikan tugas untuk membina satu permainan ringkas bertajuk *Mari Mengeja* yang dikhususkan untuk murid prasekolah. Dengan menggunakan prinsip asas reka bentuk interaktif, jelaskan elemen dan keperluan untuk membina permainan tersebut. Selepas itu, bandingkan jawapan anda dengan jawapan rakan yang lain. Buat penilaian terhadap jawapan mereka dan justifikasikan penilaian anda tersebut.
- 6** Pilih satu aplikasi berikut dan buat penilaian produk interaktif terhadap aplikasi yang dipilih.

WeChat, Instagram, Twitter

Berdasarkan penilaian anda, jawab soalan yang berikut:

- (a) Apakah tujuan aplikasi ini digunakan? Bagaimanakah pengguna menggunakan aplikasi ini?
- (b) Huraikan sama ada aplikasi ini membangkitkan respons emosi anda?
- (c) Jika respons emosi anda adalah negatif, apakah cadangan penambahbaikan untuk menjadi lebih positif? Apakah yang akan anda ubah bagi membolehkan aplikasi ini boleh diharapkan, boleh dipercayai dan meyakinkan untuk menghasilkan respons emosi yang lebih positif?
- 



STANDARD PEMBELAJARAN

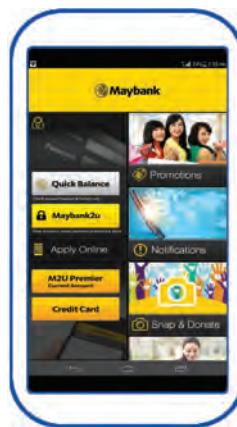
- 3.2.1 Mengaplikasi proses reka bentuk interaksi dalam atur cara yang dibangunkan
- 3.2.2 Menghasilkan prototaip paparan dan reka bentuk skrin
- 3.2.3 Membincangkan laporan daripada penilaian kuantitatif terhadap paparan dan reka bentuk skrin

Paparan dan Reka Bentuk Skrin

Kehidupan manusia kini bergantung lebih kepada penggunaan produk interaktif. Untuk tujuan itu, pembangun bagi produk interaktif perlu mereka bentuk “antara muka” pengguna yang mesra pengguna, mudah, menyeronokkan dan fleksibel. Contoh produk interaktif yang biasa digunakan ialah seperti telefon bimbit, mesin pengeluaran wang automatik (*Automated Teller Machine*), mesin pembelian tiket tren dan sebagainya.



Rajah 3.1 Contoh paparan dan reka bentuk skrin telefon bimbit berdasarkan sistem pengoperasian *Android*



Rajah 3.2 Contoh paparan dan reka bentuk skrin laman sesawang *Maybank2u*

Paparan dan reka bentuk skrin memainkan peranan yang penting dalam membangunkan sesebuah program atau perisian. Reka bentuk yang mudah dan lengkap serta mesra pengguna perlu dititikberatkan oleh seorang pereka bentuk skrin. Walau bagaimanapun, seorang pereka bentuk tidak akan mendapat penyelesaian yang tepat untuk pertama kalinya. Proses mereka bentuk paparan skrin akan melalui proses yang sama berulang kali sehingga benar-benar menjadi produk yang baik dan menepati keperluan pengguna. Proses reka bentuk interaksi terdiri daripada empat aktiviti asas iaitu mewujudkan keperluan (*establishing requirements*), mereka bentuk kaedah atau alternatif yang dapat memenuhi keperluan tersebut, menghasilkan prototaip reka bentuk supaya prototaip tersebut boleh disampaikan selari dengan keperluan produk dan membuat penilaian ke atas reka bentuk produk tersebut.



Rajah 3.3 Proses reka bentuk interaksi



1 Mengenal pasti Keperluan Interaksi

Keperluan produk perlu diberikan penekanan supaya dapat mencapai matlamat produk yang dibina, dan dapat menyelesaikan masalah yang telah dikenal pasti. Pereka bentuk perlu mengetahui pengguna yang disasarkan untuk menggunakan produk yang akan dihasilkan dan matlamat atau tujuan pengguna menggunakan produk tersebut. Pereka bentuk juga perlu membuat pemerhatian kepada produk sedia ada dan mengkaji penyelesaian masalah yang sedia ada dalam produk yang lain. Dalam membangunkan sesuatu produk yang interaktif, keperluan, aspirasi dan jangkaan pengguna perlu dibincangkan, dihalusi, dijelaskan dan mungkin perlu diskopkan semula.

Maklumat mengenai keperluan pengguna terhadap sesuatu produk boleh dikenal pasti melalui beberapa kaedah. Antara kaedah yang sering digunakan ialah melalui borang soal selidik, temu ramah, kajian tindakan, maklum balas, tinjauan dan pemerhatian kumpulan sasaran, borang tinjauan secara atas talian dan banyak lagi kaedah yang boleh diguna pakai. Hasil tinjauan daripada mana-mana kaedah ini akan dianalisis untuk mengetahui keperluan daripada pengguna terhadap apakah aktiviti interaksi yang diperlukan bagi produk yang akan dibangunkan. Jadual 3.2 menunjukkan contoh paparan dan kandungan yang dicadangkan untuk borang tinjauan.

Jadual di bawah ialah contoh paparan dan kandungan yang dicadangkan untuk borang tinjauan. Pengguna akan menjawab setiap soalan yang ditanya.

Jadual 3.2 Contoh paparan dan kandungan yang dicadangkan untuk borang tinjauan

Maklumat Umum Analisis Keperluan Produk	
Tajuk produk	
Kumpulan sasaran	
Isu / masalah / keperluan yang dikenal pasti (daripada pengguna)	
Cadangan penyelesaian	
Cadangan penyelesaian sedia ada daripada produk yang lain	
Maklumat Penetapan Keperluan Interaksi Pada Reka Bentuk yang akan dibina	
Aktiviti fungsi utama	
Aktiviti fungsi yang dimasukkan pada produk (senaraikan)	

Nota: Maklum balas diperolehi daripada beberapa orang pengguna. Daripada maklum balas itu, fungsi utama untuk program akan menjadi kriteria yang utama dalam penetapan keperluan produk.



Uji Minda

Cuba anda bina satu permainan ringkas menggunakan *Java NetBeans*. Sila lihat contoh ini.
goo.gl/7cwcbj



Tahukah
Anda

Terdapat beberapa masalah yang akan timbul apabila reka bentuk tidak diberikan perhatian iaitu,

- (a) mengurangkan kadar produktiviti pengguna
- (b) meningkatkan masa untuk pengguna mempelajari sesuatu sistem
- (c) meningkatkan kadar ralat

Tahukah Anda?

Kepentingan reka bentuk antara muka pengguna ialah:

- Memudahkan dan melicinkan proses pembelajaran untuk memahami aliran navigasi program tersebut.
- Gaya pengemudian yang bebas.
- Menyokong kandungan dan mesej.



Goh Peng Ooi merupakan seorang usahawan perisian komputer yang tergolong dalam 50 orang jutawan di Malaysia.

Beliau memiliki syarikat Silverlake Axis yang menjual perisian komputer kepada bank selama lebih daripada 25 tahun. Beliau juga merupakan jutawan dalam bidang teknologi yang pertama di Malaysia.

2 Membangunkan Reka Bentuk Alternatif

Setelah memperoleh hasil dapatan daripada kaedah yang digunakan semasa mewujudkan keperluan, pereka bentuk perlu memikirkan beberapa alternatif yang dirasakan sesuai digunakan untuk mengatasi masalah yang dihadapi. Reka bentuk alternatif ialah cadangan beberapa idea reka bentuk yang berbeza namun masih dapat menterjemahkan dengan tepat akan keperluan yang telah dikenal pasti pada proses yang pertama.

Bagaimanakah kita dapat menjana reka bentuk alternatif? Penjanaan reka bentuk alternatif boleh datang daripada pemerhatian ke atas reka bentuk produk yang lain, reka bentuk yang sama, dan daripada proses inspirasi dan kreativiti hasil daripada pengalaman pereka sendiri. Sumber-sumber ini boleh menjadi sangat dekat dengan produk baharu yang dimaksudkan, seperti produk pesaing, atau versi terdahulu sistem yang serupa, atau sesuatu yang sama sekali berbeza.

Pada fasa ini, penggunaan papan cerita amat diperlukan untuk memahami aliran kerja produk yang akan dibangunkan. Selain itu juga, lakaran beberapa produk alternatif juga perlu dihasilkan dengan tujuan pengguna dapat membuat pilihan yang terbaik. Fasa ini boleh dibahagikan kepada dua sub komponen iaitu reka bentuk konsep dan reka bentuk fizikal. Reka bentuk konsep melibatkan penghasilan model konsep untuk produk serta menerangkan apa yang boleh dilakukan oleh produk, ciri serta gambaran produk tersebut. Reka bentuk fizikal pula menitikberatkan butir-butir terperinci mengenai produk termasuklah warna, audio, imej, reka bentuk menu dan ikon yang akan digunakan. Setiap aspek yang ada dalam "antara muka" akan diambil kira.

3 Membina Prototaip Interaksi

Prototaip interaksi secara umumnya ialah gambaran awal bagi produk yang ingin dibangunkan untuk membolehkan produk tersebut dikaji dan diuji. Produk awal ini juga digunakan untuk mendapatkan maklum balas daripada pengguna sebelum produk sebenar disebarluaskan kepada umum. Pengguna akan terlibat secara aktif dalam membangunkan prototaip reka bentuk dengan cara memberikan maklum balas mengenai reka bentuk produk dari setiap aspek.

Prototaip yang dihasilkan juga perlu selari dengan keperluan pengguna yang telah ditetapkan seperti hasil analisis yang diperoleh daripada tinjauan terhadap keperluan pengguna. Untuk membangunkan prototaip tidak semestinya perlu menggunakan perisian. Pelbagai teknik boleh digunakan untuk mencapai "interaksi", dan tidak semestinya interaksi ini hanya boleh dicapai apabila menggunakan perisian. Sebagai contoh, lebih mudah sekiranya prototaip dilakarkan di atas kertas. Cara ini lebih cepat dan dapat mengurangkan kos serta masa yang digunakan untuk mengenal pasti sebarang masalah yang mungkin timbul dalam peringkat awal.

4 Membuat Penilaian Reka Bentuk Produk

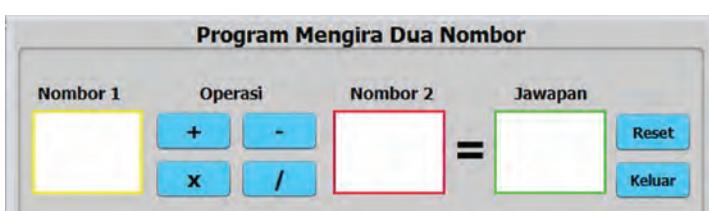
Dalam reka bentuk interaksi, cara di mana pengguna berinteraksi dengan produk ini dianggap penggerak di sebalik reka bentuk dan dengan itu kita menumpukan perhatian kepada tingkah laku yang boleh dilihat dan diukur.

Penilaian ialah proses menentukan kebolehgunaan dan penerimaan produk atau reka bentuk oleh pengguna. Penilaian diukur dalam pelbagai kriteria seperti bilangan kesilapan yang dilakukan oleh pengguna semasa menggunakan produk tersebut, menarik atau tidak produk tersebut, sepadan atau tidak produk itu dengan keperluan dan pelbagai lagi. Reka bentuk interaksi memerlukan penglibatan pengguna yang maksimum semasa proses membangunkannya dan ini akan menambahkan peluang supaya produk tersebut boleh diterima oleh golongan pengguna. Reka bentuk alternatif dinilai menerusi versi reka bentuk yang interaktif dan hasilnya akan diberikan maklum balas dalam reka bentuk yang seterusnya.



Terdapat tiga kata kunci yang sangat penting dalam reka bentuk interaksi, iaitu:

- 1 Fokus kepada keperluan dan penilaian pengguna terhadap prototaip yang dibina.
- 2 Mengenal pasti, meng dokumentasi dan bersetuju dengan kegunaan yang spesifik dan matlamat pengguna berdasarkan pengalaman.
- 3 Lelaran tidak dapat dielakkan. Hal ini adalah kerana reka bentuk yang pertama pasti mempunyai kesilapan.



Rajah 3.4 Prototaip yang telah siap

1 Mengenal pasti Keperluan Interaksi

Daripada fasa analisis SDLC, keperluan produk dan mengetahui sebab sesuatu produk itu dibina perlu dikenal pasti. Cuba fikirkan mengenai produk interaktif yang akan dibina dan apa yang anda inginkan produk itu lakukan untuk anda. Kumpulkan maklum balas daripada pengguna sasaran mengenai fungsi produk yang mereka inginkan. Lihat contoh Jadual 3.3.

Jadual 3.3 Analisis keperluan untuk menyelesaikan masalah

Tajuk produk	Program Mengira Dua Nombor
Kumpulan sasaran	Murid Darjah 3
Isu/masalah/ keperluan/tujuan yang dikenal pasti (daripada pengguna)	<p>Murid dapat:</p> <ol style="list-style-type: none"> 1. Membuat pengiraan matematik yang melibatkan input dua nombor. 2. Memilih operasi matematik yang diingini (iaitu tambah (+), tolak (-), darab (x) dan bagi (\div)) untuk diproses menjadi output. 3. Melihat paparan output bagi hasil operasi matematik yang dipilih.
Cadangan penyelesaian	Menghasilkan program aplikasi yang khusus untuk membantu murid mendapatkan jawapan daripada pengiraan matematik yang melibatkan dua nombor dan empat operasi matematik iaitu tambah (+), tolak (-), darab (x) dan bagi (\div).
Pemerhatian daripada penyelesaian yang sedia ada daripada produk-produk yang lain	<p>Berdasarkan pemerhatian / tinjauan :</p> <ol style="list-style-type: none"> 1. Pengguna boleh menggunakan kalkulator ringkas sebagai salah satu bentuk penyelesaian masalah yang boleh memaparkan hasil pengiraan sebarang nombor yang diinput. 2. Pengguna boleh menggunakan Enjin Carian laman web seperti "Google.com" yang boleh memaparkan hasil jawapan pengiraan dengan memproses pernyataan matematik yang ditaip pada ruang cariannya.
Aktiviti fungsi utama (senaraikan fungsi utama yang produk / program boleh lakukan)	<ol style="list-style-type: none"> 1. Input dua nombor. 2. Pilih operator matematik. 3. Paparkan jawapan pengiraan sebaik sahaja operator dipilih.
Aktiviti fungsi tambahan / khas yang dimasukkan pada produk (senaraikan).	<ol style="list-style-type: none"> 1. Reset : untuk mereset nilai input sedia ada kepada "null". 2. Keluar: pengguna keluar daripada sistem.

2 Membangunkan Reka Bentuk Alternatif

Reka bentuk alternatif ialah lakaran beberapa reka bentuk yang akan dicadangkan oleh pereka bentuk yang dihasilkan melalui gambaran idea pereka bentuk itu sendiri, produk yang sedia ada, hasil tinjauan maklum balas pengguna dan sebagainya. Wujudkan sekurang-kurangnya dua reka bentuk alternatif. Dalam melaksanakan reka bentuk alternatif ini, beberapa contoh lakaran papan cerita telah disediakan. Lihat Rajah 3.5(a) dan Rajah 3.5(b) yang berikut.

Jenis tulisan:

Tajuk: Program Mengira Dua Nombor—*Calibri* 16pts, *bold*, biru

Label: *Calibri*, 14pts, *Bold*

Kotak teks: *Calibri*, 18pts, *Bold*,

Penajaran (*alignment*): tengah

Warna latar belakang: kuning



Rajah 3.5(a) Contoh reka bentuk alternatif 1

No. Label	Keterangan
1	Nombor 1
2	Operasi
3	Nombor 2
4	Jawapan

No. Kotak Teks	Keterangan
1	Pengguna memasukkan input data pertama
2	Pengguna memasukkan input data kedua
3	Jawapan akan dipaparkan selepas pengguna mengklik butang operasi yang diingini.

Jenis tulisan:

Tajuk: Program Mengira Dua Nombor: *Calibri* 16pts, *bold*, biru

Label: *Calibri* 14pts, *Bold*

Kotak teks: *Calibri*, 18pts, *Bold*,

Penajaran (*alignment*): tengah

Warna latar belakang: kelabu cair



Rajah 3.5(b) Contoh reka bentuk alternatif 2

No. Label	Keterangan
1	Nombor 1
2	Operasi
3	Nombor 2
4	Jawapan
5	=

No. Kotak Teks	Keterangan
1	Pengguna memasukkan input data pertama
2	Pengguna memasukkan input data kedua
3	Jawapan akan dipaparkan selepas pengguna mengklik butang operasi yang diingini.

Lakaran reka bentuk alternatif ini akan didemonstrasikan kepada pengguna yang telah dipilih. Lakaran reka bentuk produk sebelum dan selepas input dimasukkan dan output yang terhasil selepas butang operasi yang dipilih dan diklik akan dilakarkan sama ada menggunakan kertas, kadbod, perisian komputer dan sebagainya. Demonstrasi ini dapat membantu pengguna untuk membuat pilihan reka bentuk alternatif yang disukai.



3 Membina Prototaip Interaksi

Setelah reka bentuk alternatif dengan lakaran papan cerita siap, keduanya akan diedarkan kepada pengguna untuk dinilai. Lakaran ini akan diedarkan kepada 20 pengguna yang berbeza iaitu guru dan murid. Seramai lima orang guru dan 15 orang murid telah dipilih untuk membuat pilihan reka bentuk alternatif yang disukai. Pengguna akan diberikan peluang untuk memberi komen terhadap reka bentuk alternatif tersebut. Hasil dapatan akan digunakan untuk menghasilkan reka bentuk prototaip yang lebih baik berdasarkan komen pengguna serta keperluan yang diperlukan. Hasil dapatan pemilihan reka bentuk alternatif adalah seperti dalam jadual di bawah.

Jadual 3.4 Contoh hasil dapatan pemilihan reka bentuk alternatif

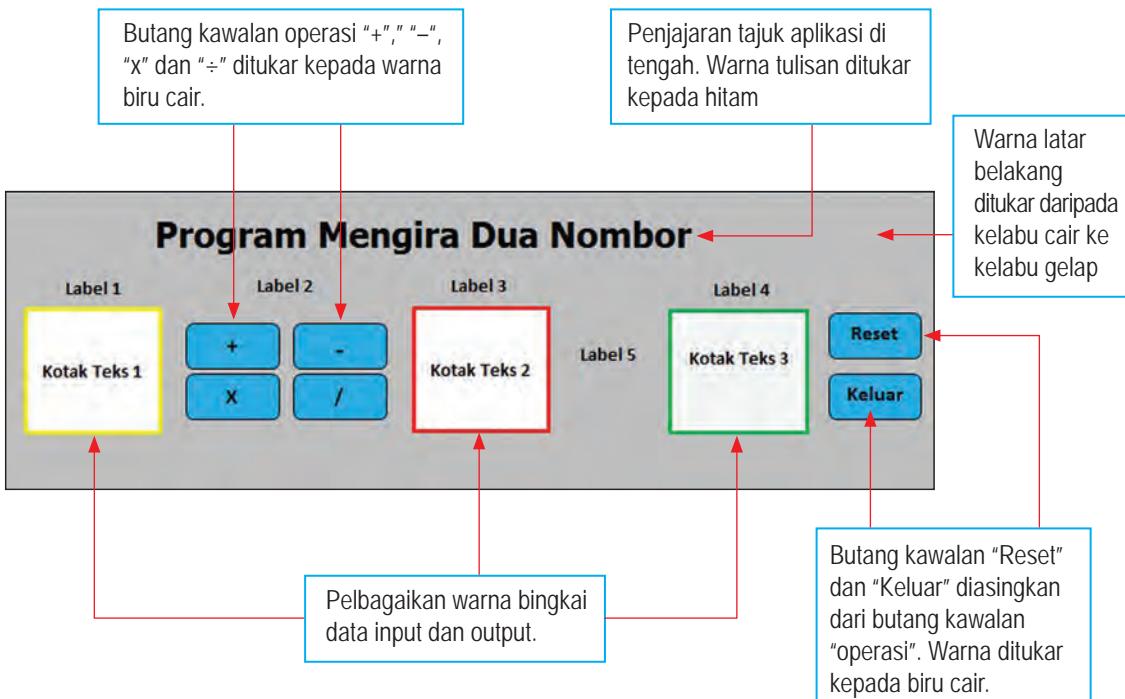
Kategori	Reka Bentuk Alternatif 1	Reka Bentuk Alternatif 2	Jumlah
Guru	2	3	5
Murid	4	11	15
Peratus (%)	30	70	100

Melalui hasil dapatan ini, peratus pengguna memilih Reka Bentuk Alternatif 2 lebih tinggi berbanding Reka Bentuk Alternatif 1 iaitu sebanyak 70%. Oleh sebab itu, pereka bentuk akan menggunakan Reka Bentuk Alternatif 2 dan akan ditambah baik selaras dengan komen pengguna. Jadual 3.5 menunjukkan komen yang diberikan oleh pengguna.

Jadual 3.5 Komen Pengguna

Ciri pada reka bentuk	Komen	Responden
Warna latar belakang Warna butang kawalan	Warna kuning (alternatif 1) dan kelabu cair (alternatif 2) tidak sesuai digunakan. Tukar kepada warna kelabu gelap. Warna butang kawalan tukar kepada biru cair.	Responden 4, 5, 7, 20
Warna bingkai Kotak Tek 1, Kotak Tek 2 dan Kotak Tek 3	Pelbagaiakan warna dalam paparan skrin supaya nampak ceria dan bersesuaian dengan murid darjah 3.	Responden 19,10,3
Butang kawalan	Butang kawalan “Reset” dan “Keluar” perlu diasingkan daripada butang kawalan operasi.	Responden 19,10,3
Tajuk “Program Mengira Dua Nombor”	Penjajaran tajuk perlu diletakkan di bahagian tengah sebelah atas “antara muka”. Warna tulisan tukar kepada hitam.	Responden 17

Melalui komen yang telah diberikan oleh pengguna, perekabentuk akan menambah baik Reka Bentuk Alternatif 2. Hasilnya adalah seperti Rajah 3.6 di bawah.



Rajah 3.6 Penambahbaikan ke atas Prototaip Reka Bentuk Alternatif 2

4 Membuat Penilaian ke atas Reka Bentuk Produk

Selepas menghasilkan prototaip reka bentuk, sekali lagi prototaip yang telah ditambah baik akan dinilai oleh pengguna. Penilaian ini merupakan penilaian akhir di mana produk telah diimplementasikan dengan menggunakan perisian komputer. Rajah 3.7 di bawah adalah paparan reka bentuk skrin yang telah siap dihasilkan.



Rajah 3.7 Contoh prototaip paparan reka bentuk skrin yang telah siap ditambah baik

Selepas prototaip paparan dan reka bentuk skrin dihasilkan, instrumen penilaian paparan dan reka bentuk skrin perlu dibina. Instrumen penilaian tersebut mestilah menepati kriteria yang diperlukan oleh pengguna semasa fasa analisis dan tinjauan masalah dilaksanakan. Berikut ialah contoh instrumen penilaian yang dibuat berdasarkan “Program Mengira Dua Nombor”. Untuk membuat penilaian akhir ke atas reka bentuk produk ini, lima orang guru dan 20 orang murid telah dipilih untuk menjadi pengguna dan menilai produk.

Langkah-langkah untuk membina instrumen penilaian:

- 1 Tentukan sasaran pengguna yang akan diberikan percubaan untuk menilai paparan skrin.
- 2 Bina soalan soal selidik dengan memfokuskan kepada perkara yang akan dibincangkan. Soalan mestilah di dalam skop dan fungsi yang ada dalam paparan skrin sahaja.
- 3 Tentukan jenis maklum balas yang boleh diberikan oleh pengguna. Contoh: Jawapan “Ya” atau “Tidak”. (Perek bentuk juga boleh menggunakan skala instrumen yang lain seperti *Skala Likert*)

Rajah 3.8 merupakan instrumen penilaian kualitatif bagi reka bentuk paparan skrin yang telah dihasilkan melalui pemerhatian mereka terhadap reka bentuk skrin.

Tajuk Projek	Program Mengira Dua Nombor		
Objektif Projek	Membolehkan murid darjah tiga mengira dua nombor dengan pantas secara automatik		
Sasaran	Murid darjah tiga		
Kategori pengguna	Guru/Murid		
Tandakan (✓) sekiranya jawapan anda Ya dan tandakan (✗) sekiranya jawapan anda Tidak.			
Kategori	Kriteria	Ya (✓)	Tidak (✗)
Reka Bentuk Skrin	1. Reka bentuk skrin mudah dan ringkas.		
	2. Pemilihan warna dan ikon yang bersesuaian dan menarik.		
	3. Sistem navigasi yang disediakan mudah dikenal pasti dan mesra pengguna.		
	4. Saiz paparan adalah sesuai.		
	5. Tidak mengandungi kesalahan ejaan.		
	6. Tidak menyebabkan murid sesat dalam penerokaan.		
	7. Laras bahasa yang digunakan mudah difahami.		
	8. Tiada gangguan teknikal semasa menggunakan program ini.		
Interaktiviti pengguna	1. Murid boleh mengawal sepenuhnya butang kawalan operasi matematik.		
	2. Butang kawalan operasi matematik adalah mesra pengguna.		
	3. Butang kawalan operasi menepati setiap pengiraan matematik.		
	4. Butang “Keluar” berfungsi dengan baik.		
	5. Butang “Reset” berfungsi dengan baik.		

Rajah 3.8 Contoh Instrumen Penilaian Kuantitatif Prototaip “Program Mengira Dua Nombor”

3.2.2 Menghasilkan Prototaip Paparan dan Reka Bentuk Skrin

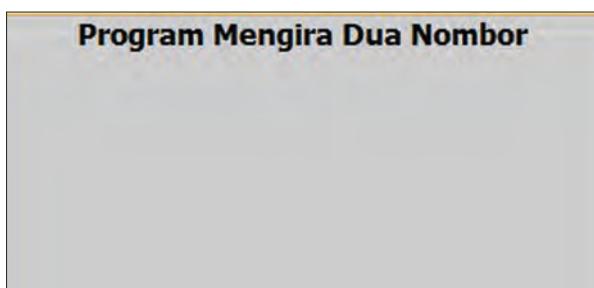
Pernahkah anda mendengar perkataan prototaip? Tahukah anda akan maksudnya? Prototaip merujuk gambaran awal kepada sesuatu produk yang akan dibangunkan yang mana akan melalui proses penilaian dan penambahbaikan secara berulang kali sehingga memperoleh prototaip yang boleh diterima oleh pengguna yang disasarkan. Prototaip tidak semestinya dihasilkan menggunakan perisian tetapi juga boleh dihasilkan menggunakan kertas, gambaran secara elektronik, video simulasi dan kadbod.

Prototaip sebenarnya membolehkan interaksi antara manusia dan komputer, untuk memperolehi pengalaman menggunakan produk tersebut dalam persekitaran yang sebenar dan untuk mengkaji kegunaan produk secara imaginasi.

Untuk menghasilkan prototaip paparan dan reka bentuk skrin bagi “Program Mengira Dua Nombor”, perisian *Java Netbeans* telah digunakan. *Netbeans* ialah salah satu platform bagi bahasa pengaturcaraan *Java* yang membolehkan pengguna membina prototaip paparan skrin dengan baik.

Langkah-langkah yang perlu diikuti untuk menghasilkan prototaip paparan dan reka bentuk skrin adalah seperti yang berikut.

- 1 Selepas memulakan projek baharu, pilih ‘jForm’ dan lakarkan ‘Swing Containers Panel’ seperti Rajah 3.9. Klik *border* pada tetingkap ‘Properties’. Taip nama “Program Mengira Dua Nombor” pada “Title” dan jadikan ‘Justification’ kepada ‘Center’. Tukar warna latar belakang ‘jForm’ kepada warna kelabu pada ruangan tetingkap ‘Properties’. Rujuk Rajah 3.10 dan Rajah 3.11.

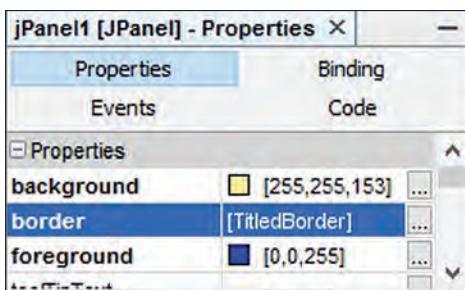


Rajah 3.9 jForm

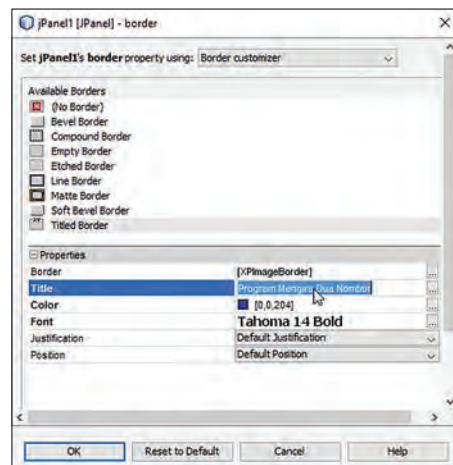


Paparan prototaip dan reka bentuk skrin merupakan teknik untuk menghasilkan paparan dan reka bentuk skrin yang sebenar. Prototaip mempunyai beberapa tujuan yang tertentu iaitu,

- sebagai satu permulaan untuk membina sesuatu paparan dan reka bentuk skrin.
- sebagai artifak yang dianalisis untuk membolehkan pereka bentuk mengkaji masalah pada prototaip tersebut.
- sebagai artifak untuk pereka bentuk menyelesaikan masalah yang terdapat pada prototaip tersebut.
- sebagai contoh yang membolehkan pereka bentuk untuk menghasilkan paparan yang lebih baik.

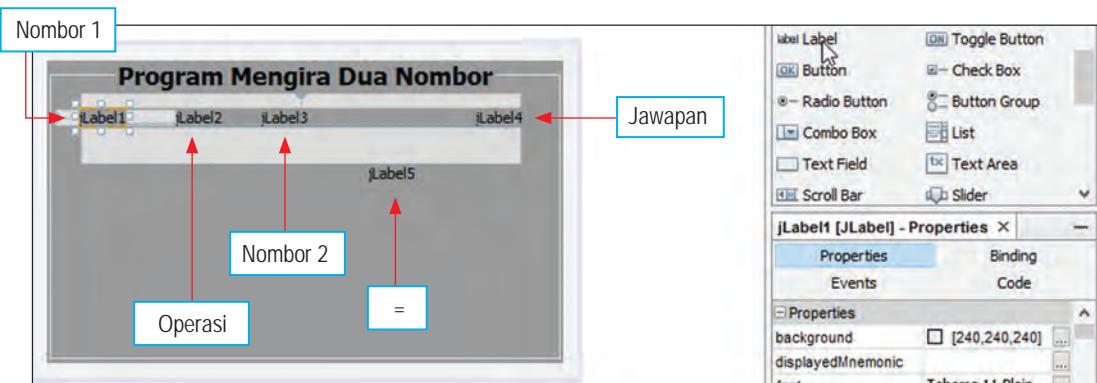


Rajah 3.10 JPanel Properties



Rajah 3.11 Menaip "Program Mengira Dua Nombor" pada Title

- 2** Tarik dan lepas (*drag and drop*) *Swing Control* 'jLabel' seperti Rajah 3.12. Susun atur label seperti lakaran papan cerita. Tukar nama 'jLabel1', 'jLabel2', 'jLabel3', 'jLabel4' dan 'jLabel5' pada ruangan teks di tetingkap 'Properties' kepada Nombor 1, Operasi, Nombor 2, Jawapan dan simbol "=" masing-masing.

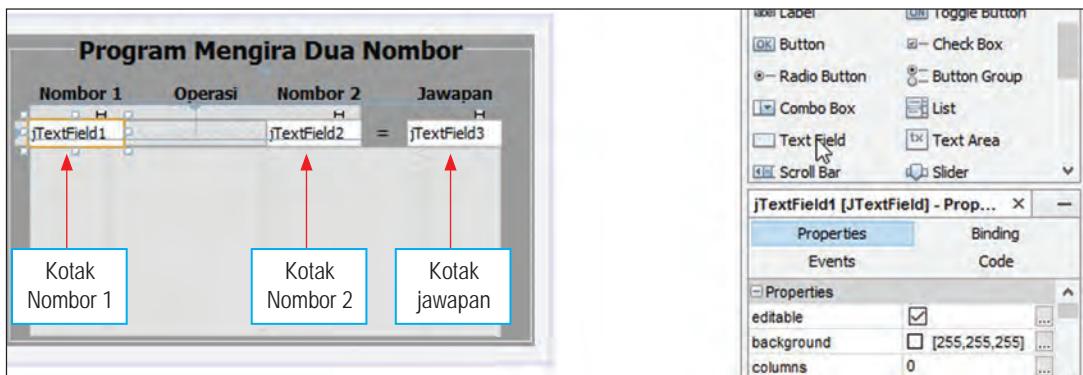


Rajah 3.12 Melakar 'Label' di atas 'jForm'

- 3** Tarik dan lepas (*drag and drop*) *Swing Control* 'jTextField' seperti Rajah 3.13. Susun atur medan teks seperti lakaran papan cerita. Kosongkan nama jTextField1, jTextField2 dan jTextField3 pada ruangan 'text' di tetingkap 'Properties'. Untuk jTextField3, klik kotak 'editable' pada tetingkap 'Properties'. Ini untuk menjadikan data di dalam kotak jawapan tidak boleh diubah.

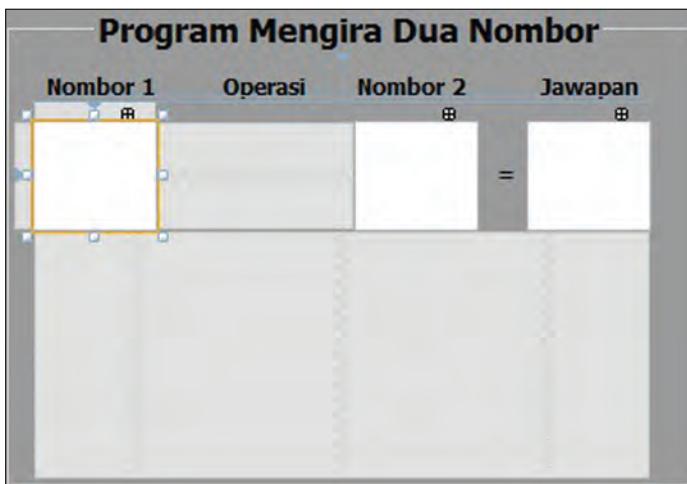


JLabel selalunya komponen yang digunakan untuk menampilkan teks yang pendek. secara praktikalnya, JLabel boleh digunakan untuk menampilkan ikon iaitu gambar yang berukuran kecil serta ikon bersama dengan teks.



Rajah 3.13 Melakar medan teks 'jTextField' di atas 'jForm'

- 4 Sesuaikan saiz medan teks seperti yang telah ditetapkan dalam papan cerita.



Rajah 3.14 Menyusun atur 'Label' dan "jTextBox" di atas 'jForm'

- 5 Tarik dan lepas (*drag and drop*) *Swing Control* 'jButton' seperti Rajah 3.15. Susun atur butang kawalan ('jButton') seperti lakaran prototaip. Tukar label 'jButton1', 'jButton2', 'jButton3', 'jButton4', 'jButton5' dan 'jButton6' di bahagian 'text' pada tetingkap 'Properties' kepada simbol-simbol operasi seperti yang dilakarkan pada papan cerita. Warnakan latar belakang butang kawalan dengan warna biru.

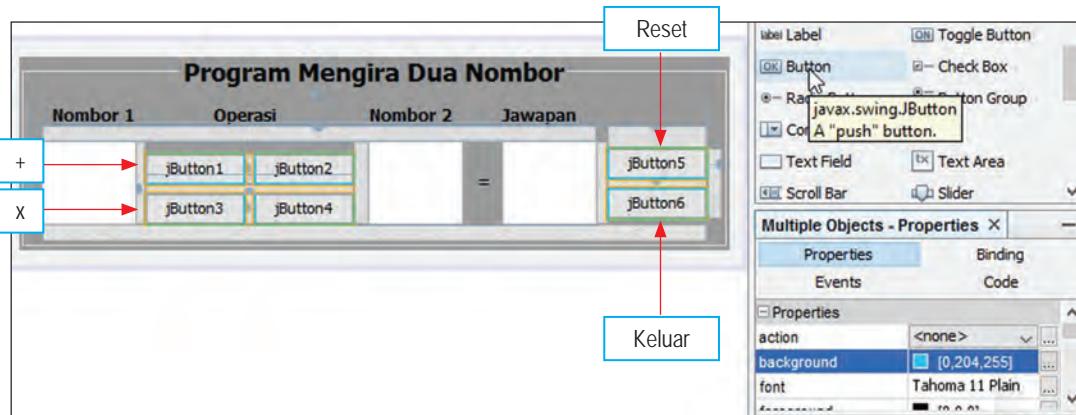
Tahukah Anda ?

Paparan maklumat merupakan salah satu gaya interaksi. Terdapat dua jenis paparan maklumat iaitu maklumat statik dan maklumat dinamik. Maklumat dinamik ialah maklumat yang berubah semasa sesi interaksi dan perubahan tersebut perlu dipaparkan kepada pengguna sistem.

Maklumat statik ialah maklumat yang tidak berubah semasa sesi interaksi.

Tahukah Anda ?

Terdapat empat jenis jButton iaitu jButton(), jButton(ikon), jButton(String, teks) dan jButton(String, teks dan ikon).



Rajah 3.15 Melakar butang kawalan 'jButton' di atas 'jForm'

Saiz teks, kotak medan, butang kawalan perlulah diselaraskan mengikut keperluan selepas penilaian oleh pengguna. Rujuk Rajah 3.7 di muka surat 333 yang menunjukkan prototaip "Program Mengira Dua Nombor" yang telah melalui penambahbaikan.

Tahukah Anda?

Penggunaan warna sangat penting untuk mereka bentuk paparan skrin. Warna yang tidak digunakan dengan betul akan menghasilkan "antara muka" yang tidak menarik. Pereka bentuk skrin dinasihatkan untuk tidak menggunakan warna yang berlebihan pada sesuatu paparan dan tidak menghubungkan maksud dengan warna misalnya merah bermaksud panas. Hal ini akan memberikan kesan kepada pengguna yang mempunyai masalah penglihatan.



46

Menghasilkan Prototaip

Kerja Individu

- 1 Anda telah ditugaskan oleh guru untuk membina sebuah sistem penghitungan markah dan purata markah murid dalam kelas anda secara automatik. Program tersebut mestilah dapat membaca input daripada pengguna dan memaparkan output kepada pengguna. Bagi tujuan itu, anda perlu menjalankan analisis terhadap permasalahan tersebut. Bina satu instrumen penilaian. Agihkan kepada murid yang telah dikenal pasti sebagai responden.
- 2 Anda perlu membina sebuah prototaip paparan bagi program yang dapat menukar unit ukuran daripada kaki kepada meter. Prototaip paparan tersebut mestilah mempunyai ruang untuk menerima input daripada pengguna dan butang kawalan untuk menukar unit ukuran daripada kaki ke meter. Setelah prototaip paparan tersebut dihasilkan, bincang bersama-sama rakan anda untuk menentukan sama ada prototaip paparan tersebut menepati instrumen penilaian atau tidak.
- 3 Hasilkan prototaip paparan untuk program menghitung isi padu sebuah piramid. Paparan tersebut mestilah boleh menerima input daripada pengguna dan memaparkan jawapan selepas menekan butang kawalan kira.

3.2.3

Membincangkan Laporan daripada Penilaian Kuantitatif terhadap Paparan dan Reka Bentuk Skrin

Penilaian kuantitatif ialah penilaian yang dilakukan untuk mengukur keberkesanannya sesuatu produk secara statistik. Penilaian kuantitatif boleh diukur dan dinyatakan dalam bentuk nombor atau peratusan mengikut kesesuaian gaya persembahan. Penilaian kuantitatif boleh dianalisis daripada data yang diperoleh daripada jumlah pengguna dan tahap kepuasan mereka seperti yang ditentukan oleh kaji selidik atau temu bual. Hasil daripada dapatan kaji selidik ini telah diubah ke dalam bentuk yang lebih spesifik menggunakan peratusan. Rujuk Jadual 3.5.

Jadual 3.5 Penilaian Kuantitatif terhadap paparan dan reka bentuk skrin yang diperolehi daripada maklum balas 25 orang pengguna

No	Item	Jumlah		PERATUS (%)	
		Ya (✓)	Tidak (✗)	Ya (✓)	Tidak (✗)
Reka Bentuk Skrin	1. Reka bentuk skrin mudah dan ringkas.	23	2	92	8
	2. Pemilihan warna dan ikon adalah bersesuaian dan menarik.	15	10	60	40
	3. Sistem navigasi yang disediakan mudah dikenali pasti dan mesra pengguna.	23	2	92	8
	4. Saiz paparan adalah sesuai.	24	1	96	4
	5. Tidak mengandungi kesalahan ejaan.	25	0	100	0
	6. Tidak menyebabkan murid sesat dalam penerokaan.	25	0	100	0
	7. Laras bahasa yang digunakan mudah difahami.	25	0	100	0
	8. Tiada gangguan teknikal semasa saya menggunakan program ini.	24	1	96	4
Interaktiviti pengguna	1. Murid boleh mengawal sepenuhnya butang kawalan operasi matematik.	25	0	100	0
	2. Butang kawalan operasi matematik mesra pengguna.	25	0	100	0
	3. Butang kawalan operasi menepati setiap pengiraan matematik.	25	0	100	0
	4. Butang "Keluar" berfungsi dengan baik.	25	0	100	0
	5. Butang "Reset" berfungsi dengan baik.	25	0	100	0
Purata				95.1	4.9

Tahukah Anda?

Penilaian terhadap paparan dan reka bentuk bagi “antara muka grafik” yang telah dihasilkan adalah sangat penting. Sebenarnya terdapat dua jenis penilaian iaitu penilaian kuantitatif dan penilaian heuristik. Apakah penilaian heuristik? Penilaian heuristik ialah peraturan yang dibangunkan dalam sesuatu tempoh masa hasil daripada “trial and error” sesuatu produk. Penilaian heuristik yang sering digunakan dalam reka bentuk “antara muka bergrafik” ialah Jacob Nielsen’s ten Heuristics’.

Analisis bagi Jadual 3.5 diperoleh daripada borang soal selidik yang telah diedarkan kepada 25 orang pengguna yang menggunakan “Program Mengira Dua Nombor”. Daripada jadual tersebut, didapati pengguna yang menyukai reka bentuk skrin tersebut ialah sebanyak 92% berbanding 8% pengguna yang tidak menyukai paparan skrin tersebut. Secara keseluruhannya, pengguna yang memberikan jawapan positif iaitu “Ya” kepada paparan dan reka bentuk skrin lebih tinggi iaitu sebanyak 95.1% berbanding jawapan “Tidak”, 4.9%. Lihat carta pai yang ditunjukkan di bawah untuk melihat perbandingan bergrafik bagi maklum balas pengguna ini.



Rajah 3.16 Carta pai bagi maklum balas pengguna terhadap paparan dan reka bentuk skrin Program Mengira Dua Nombor

Berdasarkan analisis penilaian kuantitatif ini, walaupun peratusan pengguna yang memberikan jawapan positif untuk paparan reka bentuk skrin ini adalah tinggi, namun pelbagai cadangan penambahbaikan perlu disediakan supaya “Program Mengira Dua Nombor” ini sentiasa boleh dikemas kini mengikut peredaran masa. Antara penambahbaikan yang boleh dicadangkan ialah:

- 1 Butang operasi ditambah lagi untuk operasi matematik yang lain seperti kuasa dua, punca kuasa dua dan sebagainya. Butang operasi ini ditambah mengikut keperluan tahap pengguna.
- 2 Warna memainkan peranan untuk menarik perhatian pengguna. Penggunaan warna yang lebih harmoni, iaitu kombinasi warna pastel perlu dimasukkan dalam paparan skrin.
- 3 Butang operasi juga boleh dibina menggunakan butang pilihan seperti “drop down menu”, “radio button” dan “check box”.

Sudah sampai masanya untuk anda mereka bentuk paparan dan reka bentuk skrin. Lakukan tugas ini secara berpasangan.

Syarikat Y ingin membangunkan satu sistem untuk mesin jual airnya. Kehendak syarikat ini adalah untuk mencetak resit yang menunjukkan wang yang diberikan, harga air yang dibeli dan juga baki wang.

Bincangkan antara muka bergrafik yang sesuai dan bagaimana penyelesaian di atas dijalankan. Bina "antara muka" yang mesra pengguna. Bangunkan penyelesaian masalah tersebut dalam persekitaran pengaturcaraan Java. Anda boleh merujuk <https://www.youtube.com/watch?v=dH-5UMptnJc> sebagai panduan.

- Set dan tetapkan objektif anda.
- Lukis carta alir untuk aplikasi yang anda ingin bina. Jangan lupa untuk fikirkan apa yang pengguna sasaran anda ingin lakukan secara berterusan dan pastikan agar setiap paparan yang dihasilkan dapat memudahkan pengguna melaksanakan apa yang diingini tanpa melibatkan terlalu banyak interaksi.
- Programkan aplikasi anda menggunakan alatan atau pengaturcaraan Java.
- Uji aplikasi anda dengan kerjasama daripada pengguna sasaran anda. Kumpul dan kembangkan maklum balas yang anda terima. Adakah aplikasi anda menepati objektif yang anda inginkan? Kemudian, cadangkan penambahbaikan. Harus diingat setiap penambahbaikan mestilah selari dengan objektif yang telah anda telah tentukan di peringkat awal perancangan.

Latihan Formatif 3.2

1 Apakah yang dimaksudkan dengan reka bentuk interaksi?

2 Terangkan apakah yang berlaku dalam proses reka bentuk interaksi?

3 Anda dikehendaki mereka bentuk prototaip paparan dan reka bentuk skrin.

Tajuk yang dicadangkan ialah:

Program untuk mengira isi padu silinder. Paparan anda mestilah mempunyai kotak teks untuk pengguna memasukkan data input seperti jejari dan tinggi silinder tersebut. Sediakan juga butang arahan "Kira", "Reset" dan "Keluar".

4 Daripada prototaip paparan skrin yang telah anda hasilkan, bina instrumen penilaian paparan dan reka bentuk skrin. Dalam instrumen penilaian anda, masukkan juga profil pengguna seperti umur dan jantina. Sediakan ruang untuk pengguna memberikan komen bebas. Kemudian, kumpulkan komen pengguna. Buat ujian terhadap prototaip paparan dan reka bentuk skrin yang telah anda hasilkan dengan memberikan pengguna instrumen penilaian. Kumpulkan instrumen penilaian yang telah diisi oleh pengguna.

5 Buat analisis terhadap instrumen penilaian yang telah dijalankan oleh pengguna sasaran anda.

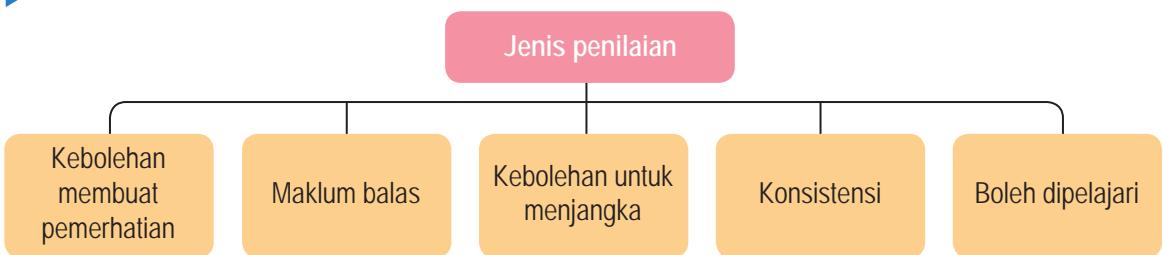
Kumpulkan maklumat yang diperoleh daripada penilaian tersebut. Cadangkan penambahbaikan selepas instrumen penilaian paparan dan reka bentuk skrin dijalankan.

6 Daripada analisis instrumen penilaian yang telah dijalankan ke atas prototaip paparan dan reka bentuk skrin, hasilkan prototaip paparan dan reka bentuk skrin baharu yang telah ditambah baik dengan komen daripada pengguna sasaran anda.

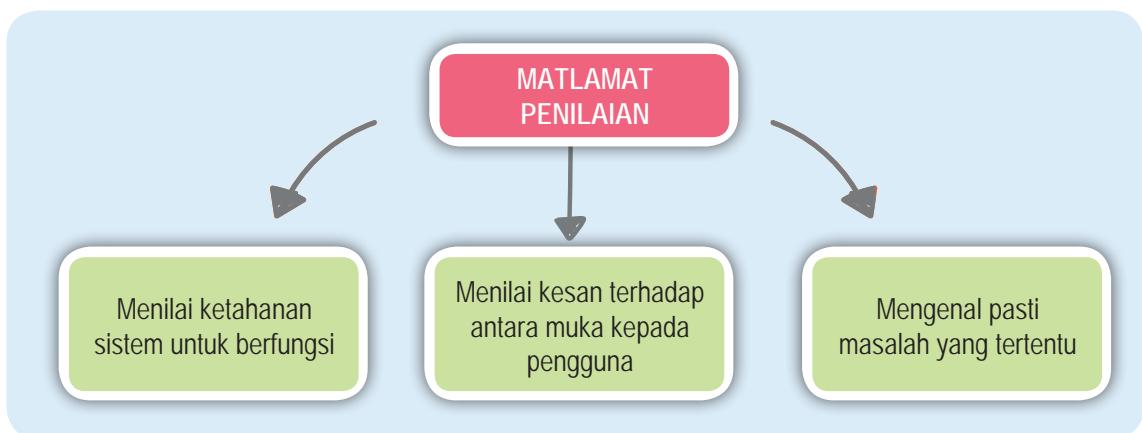


IMBAS SEMULA

Penilaian Produk Interaktif



Proses Reka Bentuk Interaksi



LATIHAN SUMATIF

3

Soalan Objektif

Jawab **semua** soalan.

- 1 Prototaip ialah implementasi dari pada konsep
A terakhir sahaja
B permulaan sahaja
C permulaan dan terakhir
D bukan semua di atas
- 2 Interaksi manusia dan komputer ialah _____ antara manusia dengan komputer.
A perantara media
B hubungan
C ciptaan
D rangkaian
- 3 Pengguna boleh membuat jangkaan terhadap perkara yang akan berlaku dalam aliran proses sesuatu aplikasi terlebih dahulu sebelum mereka menggunakan secara praktikal. Penyataan ini merujuk prinsip
A maklum balas (*feedback*)
B boleh dipelajari (*learnability*)
C kebolehan untuk menjangka (*predictability*)
D konsistensi (*consistency*)
- 4 Antara yang berikut, yang manakah **bukan** keperluan interaksi manusia dengan komputer?
A Mengurangkan kos pembangunan
B Mengurangkan kos selepas jualan
C Mengembangkan aktiviti dan menambahkan pengalaman otak manusia
D Menjadikan komputer sehebat otak manusia
- 5 Sistem yang tidak mematuhi prinsip reka bentuk ini akan membuat paparan antara muka yang pelbagai dengan fungsi yang pelbagai sehingga mengelirukan pengguna. Prinsip yang dimaksudkan ialah
A kebolehgunaan (*usability*)
B konsisten (*consistency*)
C boleh dipelajari (*learnability*)
D kebolehan untuk menjangka (*predictability*)

Soalan Struktur

Jawab **semua** soalan.

- 1 Senaraikan bidang yang berkaitan dengan interaksi manusia dan komputer.
- 2 Lengkapkan pernyataan yang berikut dengan jawapan yang betul.
Stail paparan dan susun atur elemen-elemen _____ yang akan digunakan perlu ditentukan. Reka bentuk interaksi adalah mengenai mereka bentuk keseluruhan sistem yang berhubungan, iaitu _____, antara muka, konteks, persekitaran dan _____.
- 3 Nyatakan **lima** prinsip reka bentuk interaksi.
- 4 Lengkapkan aliran proses reka bentuk interaksi yang berikut.

5 Lengkapkan aliran proses reka bentuk interaksi yang berikut.



Penyelesaian Masalah

Jawab **semua** soalan.

-  1 Sekolah anda memerlukan sebuah sistem pendaftaran murid yang baharu untuk murid tingkatan satu. Sebagai seorang pelajar sains komputer, anda diminta oleh sekolah untuk membangunkan sebuah paparan reka bentuk interaksi yang sesuai berdasarkan maklumat yang disenaraikan di bawah.
 - i. maklumat murid
 - ii. maklumat Ibu bapa
 - iii. maklumat kokurikulum
 - iv. butang ‘seterusnya’
 - v. butang ‘kembali’
 - vi. butang ‘utama’
 - vii. butang lain-lain yang boleh ditambah mengikut kreativiti anda.
-  2 Setiap kali selepas peperiksaan, anda akan melihat keputusan peperiksaan anda menerusi sistem analisis peperiksaan (SAPS). Berdasarkan paparan sistem analisis peperiksaan, senaraikan prinsip yang terdapat di dalam sistem tersebut dan huraikan prinsip yang telah disenaraikan serta cadangkan penambahbaikan untuk sistem itu.
-  3 Pengurusan masa sangat penting untuk manusia. Untuk menguruskan masa, salah satu produk yang diperlukan oleh manusia ialah jam. Jam bukan sahaja dapat memberitahu anda masa dan tarikh tetapi jam boleh mengingatkan anda mengenai sesuatu perkara yang perlu anda lakukan. Paparan masa pula boleh ditunjukkan melalui sistem 12 jam atau sistem 24 jam, bergantung kepada keinginan anda sebagai seorang pengguna. Lakarkan satu reka bentuk yang inovatif bagi paparan masa yang anda inginkan mengikut keperluan anda. Paparan masa anda mungkin berbentuk jam tangan, jam penggera, jam dinding dan sebagainya. Berikan dua lakaran alternatif untuk produk anda.

GLOSARI

Algoritma

Satu set arahan untuk menyelesaikan masalah

Atribut

Ciri-ciri yang menghuraikan sesuatu entiti

Bait

Satu unit yang terdiri daripada 8 bit

Carta alir

Carta yang terdiri daripada nod yang diwakili simbol dan anak panah yang menunjukkan sesuatu proses

Dokumentasi

Proses mengutip dan mengumpulkan data, maklumat serta ringkasan

Entiti

Objek yang unik dan boleh dikenalpastikan dalam persekitaran yang tertentu seperti orang, tempat, benda atau peristiwa yang dikutip dan disimpan sebagai data

Inden

Cara penulisan atur cara yang memudahkan pembacaan

Input

Data yang dimasukkan ke dalam komputer untuk diproses.

Integrasi data

Kepastian dan pengekalan kualiti data yang sah, tepat, stabil dan terkawal melalui keseluruhan kitar hayat data

Java

Bahasa atur cara berorientasikan objek

Kebergantungan fungsi penuh

Berlaku antara dua atribut data apabila atribut-atribut biasa ditentukan sepenuhnya oleh kesemua atribut kunci

Kebergantungan fungsi separa

Berlaku antara dua atribut data apabila atribut-atribut biasa ditentukan oleh salah satu daripada atribut kunci dalam hubungan

Kebergantungan fungsi transitif

Berlaku antara dua atribut data apabila atribut biasa bergantung kepada atribut biasa

Kelewahan

Pertindihan data yang berpunca daripada salinan data tetapi di lokasi berlainan

Ketekalan

Konsistensi atau keselarasan data

Modularisasi

Proses menghimpunkan kod komputer ke dalam modul

Nombor nyata

Nombor yang terdiri daripada semua nombor yang mempunyai titik perpuluhan atau bahagian pecahan

Output

Hasil pemprosesan komputer yang dikeluarkan dari unit sistem

Pemboleh ubah sejagat

Pemboleh ubah yang boleh diakses daripada semua modul

Pemboleh ubah setempat

Pemboleh ubah yang hanya berfungsi dalam subatur cara yang diisytiharkan sahaja

Pengkompilan

Proses menukar kod pengaturcaraan kepada kod boleh laksana

Peniskalaan

Memberikan tumpuan kepada maklumat yang penting sahaja dan mengabaikan bukti yang tidak relevan

Pseudokod

Penyataan-penyataan komputer yang ditulis dalam bentuk bahasa pertuturan manusia dan mempunyai nombor turutan mewakili algoritma penyelesaian masalah

Ralat sintaks

Ralat yang berlaku akibat kesalahan tatabahasa dan penggunaan objek atau aksara yang tidak dikenali

Sintaks

Peraturan yang diperlukan oleh komputer untuk melaksanakan arahan dengan tepat

Skema hubungan

Perhubungan antara dua atau lebih entiti

Tatasusunan

Pemboleh ubah yang membolehkan koleksi beberapa nilai data dalam satu-satu masa disimpan dalam ruang memori berindeks.



SENARAI RUJUKAN

- Barnes, D. J. dan Kölling, M., 2016. Objects First with Java: A Practical Introduction using BlueJ. Edisi 6. England: Pearson Education Limited.
- Chua, H. C., 2016. Java Programming Tutorial: Applets & Web Start Applications – Java Rich Internet Applications (RIA), Yet Another Insignificant Programming Notes,. Diakses pada Mei 2016 daripada https://www.ntu.edu.sg/home/ehchua/programming/java/J4c_AppletWebstart.html
- Chua, H. C., 2016. Java Programming Tutorial: Exercises on Java Basics, Yet Another Insignificant Programming Notes. Diakses pada April 2016 daripada https://www3.ntu.edu.sg/home/ehchua/programming/java/J2a_BasicsExercises.html
- Dayang dan Rosli, 2001. Bab 2 – Penyelesaian Masalah & Pengenalan Kepada Pengaturcaraan. Diakses pada Januari 2016 daripada <http://comp.utm.my/dayang/files/2014/03/Bab2.pdf>
- Deitel, P. J. dan Deitel, H. M., 2014. Java How To Program (Early Objects). Edisi 10. England: Pearson Education Limited.
- Germain, H. J. de St., 2016. Problem Solving, Jim's Computer Science Topics Area – Computer Science General Topics. Diakses pada Mac 2016 daripada http://www.cs.utah.edu/~germain/PPS/Topics/problem_solving.html
- Gibbs, I., 2014. The Five Principles of Interaction Design. Diakses pada April daripada <http://info.obsglobal.com/blog/2014/08/the-five-principles-of-interaction-design>
- InstanceOfJava team., 2015. Data Types in Java with Example Programs. Diakses pada Jun 2016 daripada <http://www.instanceofjava.com/2016/03/data-types-in-java-examples.html>
- Kaneria, D., 2014. Introduction to Data Structures and Algorithm. Diakses pada Februari 2016 daripada <http://www.slideshare.net/DhavalKaneria/introduction-to-data-structures-and-algorithm-35441665>
- NetBean Community, 2016. NetBeans IDE. Diakses pada Jun 2016 daripada <https://netbeans.org/>
- Sedgewick, R. dan Wayne, K., 2016. 4.1 Analysis of Algorithms, Introduction to Programming in Java <http://introcs.cs.princeton.edu/java/41analysis/>.
- Sprankle, M. dan Hubbard, J., 2012. Problem Solving and Programming Concepts, Edisi 9. England: Pearson Education Limited.
- Tutorials Point team, 2016. Java – Basic Datatypes', Tutorials Point. Diakses pada Februari 2016 daripada http://www.tutorialspoint.com/java/java_basic_datatypes.htm
- Tutorials Point team, 2016. Learn Java Programming: Java – Tutorial. Diakses pada April 2016 daripada <http://www.tutorialspoint.com/java/>
- Wu, C. T., 2004. An Introduction to Object-Oriented Programming with Java. Edisi 5. New York, United States: McGraw-Hill.
- Yew, K. H., 2013. Computer Science for Matriculation Semester 2. Shah Alam, Selangor: Oxford Fajar.

Dengan ini **SAYA BERJANJI** akan menjaga buku ini dengan baik dan bertanggungjawab atas kehilangannya serta mengembalikannya kepada pihak sekolah pada tarikh yang ditetapkan

Skim Pinjaman Buku Teks

Sekolah _____

Tahun	Tingkatan	Nama Penerima	Tarikh Terima

Nombor Perolehan: _____

Tarikh Penerimaan: _____

BUKU INI TIDAK BOLEH DIJUAL