

APLIKASI KEUANGANS

APLIKASI QEUANGANS

Dengan Android

Faris Muhammad Ihsan, Syabriena Putri Veriane
Politeknik Pos Indonesia



Kreatif Industri Nusantara

Penulis:

Rolly Maulana Awangga

ISBN : 978-602-53897-0-2

Editor:

M. Yusril Helmi Setyawan

Penyunting:

Syafrial Fachrie Pane

Khaera Tunnisa

Diana Asri Wijayanti

Desain sampul dan Tata letak:

Deza Martha Akbar

Penerbit:

Kreatif Industri Nusantara

Redaksi:

Jl. Ligar Nyawang No. 2

Bandung 40191

Tel. 022 2045-8529

Email : awangga@kreatif.co.id

Distributor:

Informatics Research Center

Jl. Sariasih No. 54

Bandung 40151

Email : irc@poltekpos.ac.id

Cetakan Pertama, 2019

Hak cipta dilindungi undang-undang

Dilarang memperbanyak karya tulis ini dalam bentuk dan dengan cara
apapun tanpa ijin tertulis dari penerbit

*‘Jika Kamu tidak dapat
menahan lelahnya
belajar, Maka kamu harus
sanggup menahan
perihnya Kebodohan.’
Imam Syafi’i*

CONTRIBUTORS

ROLLY MAULANA AWANGGA, Informatics Research Center., Politeknik Pos Indonesia, Bandung, Indonesia

CONTENTS IN BRIEF

1	Sejarah Android	1
2	Android Studio	23
3	Java	29
4	Java Development Kit	61

DAFTAR ISI

Daftar Gambar	xi
Daftar Tabel	xiii
Foreword	xvii
Kata Pengantar	xix
Acknowledgments	xxi
Acronyms	xxiii
Glossary	xxv
List of Symbols	xxvii
Introduction	xxix
<i>Rolly Maulana Awangga, S.T., M.T.</i>	
1 Sejarah Android	1
1.1 Sejarah Android	2
1.2 Versi Pada Android	4
2 Android Studio	23
	ix

2.1	Android Studio	24
2.1.1	Langkah Download Android Studio	27
2.1.2	Cara Install Android Studio	28
3	Java	29
3.1	Sejarah Java	29
3.2	Sejarah perkembangan	30
3.3	Struktur Program	31
3.3.1	Statement dan Ekspresi pada Java	32
3.3.2	Blok Program Java	32
3.3.3	Penulisan Komentar pada Java	33
3.3.4	String dan Character	34
3.3.5	Case Sensitive	34
3.4	Mengenal Tipe Data Dasar di Java	34
3.5	Variabel	36
3.6	Dasar - dasar Pemrograman	38
3.6.1	Operator	38
3.7	Percabangan pada Java	49
3.7.1	Percabangan IF	50
3.7.2	Percabangan IF/ELSE	51
3.7.3	Percabangan IF/ELSE/IF	53
3.7.4	Percabangan SWITCH/CASE	55
3.7.5	Percabangan dalam percabangan (<i>Nested</i>)	57
4	Java Development Kit	61
4.1	JDK <i>Java Development Kit</i>	61
	Daftar Pustaka	63
	Index	65

DAFTAR GAMBAR

1.1	Logo Android	2
1.2	OEM Android Smartphone	3
1.3	Android Apple Pie	4
1.4	Android Banana Bread	6
1.5	Android Cupcake	7
1.6	Android Donut	8
1.7	Android Eclair	9
1.8	Android Froyo	11
1.9	Android Gingerbread	12
1.10	Android Honeycomb	13
1.11	Android Ice Cream Sandwich	15
1.12	Android Jelly Bean	16
1.13	Android Kitkat	17

1.14	Android Lollipop	19
1.15	Android Marshmallow	19
1.16	Android Nougat	21
1.17	Android Oreo	21
2.1	Android Studio	24
3.1	Java	29
3.2	Hasil Pengisian Nilai Operator Penugasan	40
3.3	Hasil Penambahan Operator Penugasan	41
3.4	Hasil Pengurangan Operator Penugasan	41
3.5	Hasil Perkalian Operator Penugasan	42
3.6	Hasil Pembagian Operator Penugasan	42
3.7	Hasil Sisa Bagi Operator Penugasan	43
3.8	Hasil Operator Logika AND	43
3.9	Hasil Operator Logika OR	44
3.10	Hasil negasi/kebalikan	44
3.11	Hasil operator lebih besar	46
3.12	Hasil operator lebih kecil	46
3.13	Hasil operator sama dengan	46
3.14	Hasil operator tidak sama dengan	47
3.15	Hasil operator lebih besar sama dengan	47
3.16	Hasil operator lebih kecil sama dengan	48
3.17	Hasil Operator Bitwise	49
3.18	Alur logika percabangan IF	50
3.19	Alur logika percabangan if else	51
3.20	Hasil program if else	52
3.21	Flowmap logika IF/ELSE/IF	53
3.22	Hasil program IF/ELSE/IF	54
3.23	Flowmap logika SWITCH/CASE	55
3.24	Hasil program SWITCH/CASE	57
3.25	Flowmap logika percabangan dalam percabangan	57
3.26	Hasil program percabangan dalam percabangan	59

DAFTAR TABEL

3.1	Jangkauan nilai tipe data primitif	36
3.2	Operator Aritmatika	39
3.3	Operator Penugasan	40
3.4	Operator Logika	43
3.5	Tabel Kebenaran	45
3.6	Operator Perbandingan	45
3.7	Operator Bitwise	48

Listings

FOREWORD

Sepatah kata dari Kaprodi, Kabag Kemahasiswaan dan Mahasiswa

KATA PENGANTAR

Buku ini diciptakan bagi yang awam dengan git sekalipun.

R. M. AWANGGA

*Bandung, Jawa Barat
Februari, 2019*

ACKNOWLEDGMENTS

Terima kasih atas semua masukan dari para mahasiswa agar bisa membuat buku ini lebih baik dan lebih mudah dimengerti.

Terima kasih ini juga ditujukan khusus untuk team IRC yang telah fokus untuk belajar dan memahami bagaimana buku ini mendampingi proses Intership.

R. M. A.

ACRONYMS

3GP	3gpp
3GPP	3rd Generation Partnership Project
A2DP	Advance Audio Distribution Profile
API	Application Programing Interface
ASF	Apache Software Foundation
AVRCP	Audio / Video Remote Control Profile
C2DM	Android Cloud to Device Messaging
CDMA	Code-Vision Multiple Access
E-MAIL	Electronic Mail
EVDO	Evolution-Data Optimized
G-MAIL	Google Mail
HP	Handphone
HTML	Hypertext Markup Language
IMAP	Internet Message Access Protocol
JIT	Just-In-Time Manufacturing

LED	Light Emitting Diode
MMS	Multimedia Messaging Service
MP4	MPEG4
MPEG	Moving Picture Expert Group
OEM	Original Equipment Manufacturer
OHA	Open Handset Alliance
OS	Operating System
POP3	Post Office Protocol Version 3
SMS	Short Message Service
SMTP	Simple Mail Transfer Protocol
UI	User Interface
USB	Universal Serial Bus
WI-FI	Wireless Fidelity
WVGA	Wide VGA
VGA	Video Graphics Array
XHTML	Extensible Hypertext Markup Language

GLOSSARY

git	Merupakan manajemen sumber kode yang dibuat oleh linus torvald.
bash	Merupakan bahasa sistem operasi berbasiskan *NIX.
linux	Sistem operasi berbasis sumber kode terbuka yang dibuat oleh Linus Torvald

SYMBOLS

- A Amplitude
- $\&$ Propositional logic symbol
- a Filter Coefficient

- \mathcal{B} Number of Beats

INTRODUCTION

ROLLY MAULANA AWANGGA, S.T., M.T.

Informatics Research Center
Bandung, Jawa Barat, Indonesia

Pada era disruptif saat ini. git merupakan sebuah kebutuhan dalam sebuah organisasi pengembangan perangkat lunak. Buku ini diharapkan bisa menjadi penghantar para programmer, analis, IT Operation dan Project Manajer. Dalam melakukan implementasi git pada diri dan organisasinya.

Rumusnya cuman sebagai contoh aja biar keren[?] [?].

$$ABCDEF\alpha\beta\Gamma\Delta\sum_{def}^{abc} \tag{I.1}$$

BAB 1

SEJARAH ANDROID

1.1 Sejarah Android



Gambar 1.1 Logo Android

Android adalah sistem operasi dengan basis linux yang dirancang untuk perangkat yang bergerak atau layar sentuh *touchscreen* seperti telepon pintar *smartphone* dan *tablet*. Android merupakan sistem operasi *open source* (aplikasi yang tidak dipegang hanya untuk seorang saja tetapi orang lain bisa menggunakan *sourcecode* tersebut). Android ini menggunakan lisensi dari Google yang kodenya tersebut berada dibawah Lisensi *Apache*. Lisensi *Apache* merupakan lisensi yang bebas untuk *software* yang ditulis oleh *Apache Software Foundation* (ASF).

Android berdiri pada bulan Oktober 1980, tepatnya di Palo Alto, California. Android ini didirikan oleh Andy Rubin, Rich Muner, Nick Sears, dan Chris White. Android dikembangkan oleh perusahaan dengan nama *Ancroid.inc*. Pada tahun 2005 perusahaan tersebut mendapatkan dukungan finansial dari *google.inc*

Pada awalnya Android tidak dibuat untuk ponsel, namun android pertama kali dibuat untuk kamera digital. Tetapi, dengan melihat peluang yang lebih besar jika Android digunakan untuk perangkat *mobile*, maka Android digunakan pada perangkat *mobile* yang ditujukan untuk menyaingi *Symbian* dan *Windows Mobile*.

Secara resmi, Android dirilis pada tahun 2007 bersamaan dengan berdirinya *Open Handset Alliance*. *Open Handset Alliance* merupakan *Open Source Developer* bagi perangkat *mobile* atau seluler. Tahun 2008, *Handphone* pertama yang menggunakan sistem operasi android kemudian dirilis, *handphone* ini bernama *HTC Dream*. Dua tahun setelah perilisian *Handphone* ini, Google menyusul dengan merilis *Smartphone* dengan seri *Nexus One* yang proses pembuatannya dibantu oleh *HTC*. Dengan dirilisnya *Handphone* tersebut, memancing kemunculan-kemunculan berbagai brand dari *OEM* yang bermacam-macam. Dimulai dari *Samsung*, *Lenovo*, *HTC*, *ASUS*, *LG* dan masih banyak lagi.



Gambar 1.2 OEM Android Smartphone

Selain berfokus pada *smartphone*, Google juga banyak mengembangkan aplikasi Android untuk perangkat lainnya. Contohnya Google mengembangkan Android TV yang digunakan untuk televisi, Android Auto yang digunakan pada, dan Android Wear pada jam tangan. Aplikasi android tersebut memiliki *Interface* yang berbeda beda sesuai dengan kebutuhan dan fungsionalitasnya masing-masing.

Open Source Code dan lisensi yang diguakan pada Android tentunya dapat membuat Sistem Operasi ini dapat diubah-ubah dan dimodifikasi dengan bebas yang kemudian dapat di distribusikan oleh para *developer* Android itu sendiri. Selain mudah untuk digunakan, Android memiliki *Developer Community* (Komunitas Pengembang) sendiri yang dapat memperluas fungsionalitas dari perangkat yang umumnya digunakan menggunakan bahasa pemrograman *Java*. Selain *java*, Android ini juga dapat menggunakan bahasa pemrograman *Kotlin*. Lebih dari Satu juta aplikasi kemudian tersedia untuk android, dan miliaran aplikasi telah melakukan *download* dari *Google Play* (toko utama yang berisi aplikasi-aplikasi dari Android)

Dimulai sejak tahun 2008, Android melakukan pembaruan untuk meningkatkan kinerja aplikasinya secara bertahap dengan cara menambahkan fitur-fitur yang baru dan memperbaiki *error* dan *bug* yang terdapat pada produk dengan versi yang sebelumnya. Setiap versi dari android biasanya disusun dengan nama alfabetis dan nama yang digunakan adalah nama makanan-makanan yang ringan atau cemilan. Sebagai contoh pada android versi 7.0 yang diberi nama Android *Nougat*, Kemudian Android 8.0 yang diberi nama *Oreo* dan seterusnya.

1.2 Versi Pada Android

Pada awal kemunculan Android, Android telah mengeluarkan banyak versi. Setiap versi dari android ini tentunya memiliki fitur-fiturnya masing-masing sesuai dengan perkembangan zaman. Hal ini tentunya sebagai cara agar mengalahkan pesaingnya yang menggunakan OS yang lainnya seperti Apple iOS, Windows, Blackberry, Symbian dan lainnya.

1. Android 1.0 **Apple Pie**



Gambar 1.3 Android Apple Pie

Android versi pertama ini merupakan android dengan versi 1.0 yang diberi nama *Android Apple Pie* yang dirilis oleh android pada 23 September 2008 dan hanya memiliki fitur yang terbatas. Fitur fitur tersebut adalah:

- (a) *Play Store*
- (b) Kamera
- (c) *Web Browser*
- (d) *G-Mail Synchronization*
- (e) Kontak
- (f) Google Agenda
- (g) *Google Maps*

Selain itu, Android versi ini juga sudah mendukung fasilitas youtube. Setidaknya Google dan OHA telah merilis 2 versi saat sebelum Android beta yang dirilis pada bulan November 2007. Pada Android Versi Alpha memiliki sebutan atau *codename Astro Boy, Bender*, dan R2-D2.

KELEBIHAN

- (a) **Android Market**
Android market merupakan aplikasi untuk *mendownload* dan *mengupdate* aplikasi yang terinstall melalui toko resmi dari Android.
- (b) **Web Browser**
Android *Web Browser* merupakan aplikasi untuk *searching website*, menampilkan halaman *Web HTML* dan *XHTML* dan dapat digunakan untuk melihat halaman web dengan *fullscreen* dan dapat juga diperbesar. untuk menampilkan, memperbesar dan melihat dalam layar penuh halaman *Web HTML* dan *XHTML*
- (c) **Kamera**
- (d) Memungkinkan pengelompokan ikon-ikon aplikasi ke dalam satu folder pada bagian layar utama (*homescreen*).
- (e) Dapat memiliki dan mengakses *E-Mail*, mendukung fasilitas POP3, IMAP4, dan SMTP
- (f) Sinkronisasi *G-mail* dengan menggunakan aplikasi *G-mail*.
- (g) Sinkronisasi *Google Contacts* dengan menggunakan aplikasi *People*.
- (h) Sinkronisasi *Google Calendar* dengan menggunakan aplikasi *Calendar*.
- (i) **Aplikasi Google Maps**
Aplikasi *Google Maps* ini menyediakan informasi mengenai Latitude, der-dapat fitur *Street View*, dapat melihat melihat peta dan tampilan melalui citra satelit, menemukan lokasi yang akan dituju dan dapat memberi petunjuk arah saat mengemudi kendaraan maupun saat berjalan-jalan.
- (j) **Google Sync**
Fitur ini dapat memungkinkan pengelolaan sinkronisasi pada aplikasi *Gmail*, *People*, dan *Calendar*.
- (k) **Google Search**
Fitur ini dapat memungkinkan pengguna untuk *Searching* sesuatu menggunakan *website*.
- (l) **Google Talk**
Google Talk merupakan sebuah aplikasi pesan instan yang diproduksi oleh google
- (m) Pesan instan, pesan teks (SMS), dan MMS.
- (n) **Media Player**
Media Player ini digunakan untuk mengelola, mengimpor, dan memutar file yang mendukung pada berkas penyimpanan. Tetapi, pada versi ini belum menyediakan dukungan *Video* dan *Bluetooth Stereo*
- (o) **Notifikasi**
Notifikasi ini merupakan fitur yang akan muncul pada status bar, dengan diberikan pilihan pengaturan untuk mengatur *Ringtone*, cahaya *LED* yang dikeluarkan maupun nada getar.

(p) *Voice Dialer*

Voice Dialer ini memberikan akses kepada pengguna untuk memanggil kontak tanpa harus mengetikkan nama ataupun nomor telepon orang yang akan dituju.

(q) *Wallpaper*

Fitur ini dapat digunakan pengguna untuk mengatur gambar *Wallpaper* pada *Homescreen* perangkat android pengguna.

(r) *Youtube Video Player*

(s) Fitur Pendukung Lainnya seperti:

- i. Jam Alarm
- ii. Kalkulator
- iii. Panggilan
- iv. *Homescreen Launcher*
- v. Galeri
- vi. Pengaturan

(t) Wi-Fi

(u) Bluetooth

KEKURANGAN

- (a) Versi Android ini pada awalnya belum memiliki nama yang cocok sehingga tidak diberi nama dan hal tersebut dapat membuat bingung masyarakat karena tidak akan mudah untuk diingat.

2. Android 1.1 **Banana Bread**



Gambar 1.4 Android Banana Bread

Pada Februari 2009, Android meng-*upgrade* dari versi sebelumnya (versi 1.0) ke versi 1.1 yang bernama Android *Banana Bread*. Fitur pada android ini tidak jauh bedanya dengan versi sebelumnya. *Smartphone* pertama yang menggunakan versi ini adalah *HTC*. Android 1.1 *Banana Bread* ini memiliki nama

lain yaitu "*Petit Four*". Nama ini tentunya bukan nama resmi yang dikeluarkan oleh pihak Android. Versi ini merupakan versi yang berkembang dan memperbaiki beberapa bug (*Error*) yang ada pada Android sebelumnya. Versi ini juga mengubah *API* dari android yang sebelumnya. Selain itu, Android ini menambahkan fitur fitur baru yaitu:

- (a) *Maps* dan pencarian lokasi bisnis sudah terdapat rincian tempat.
- (b) Tombol Panggilan yang dapat di sembunyikan atau di tampilkan.
- (c) Dapat menyimpan lampiran dalam pesan
- (d) Mendapat dukungan *Marquee* pada ruang sistem.

3. Android 1.5 Cupcake



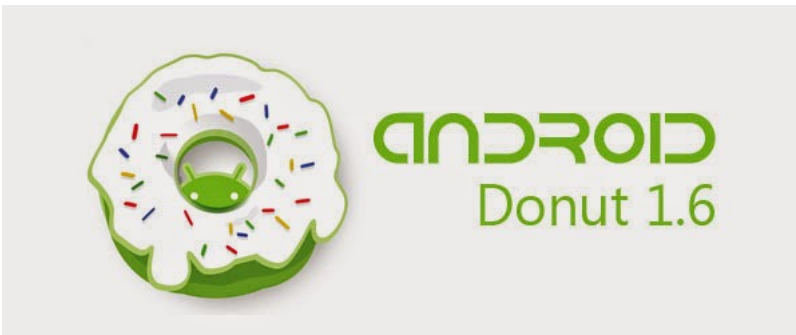
Gambar 1.5 Android Cupcake

Android Versi ini diluncurkan pada bulan April 2009 dan tidak memiliki perbedaan yang signifikan dengan versi Android yang terdahulu. Meski dengan sedikit perbedaan, android ini mendapatkan fitur tambahan seperti *Support Bluetooth A2DP, AVRCP, Soft Keyboard* dengan *Text Suggestion, Record* ataupun *Watch Videos*. Android ini merupakan versi Android yang mulai menggunakan nama makanan cemilan yaitu "*Cupcake*". Pada Android versi ini, terdapat pembaruan, penambahan fitur, dan Perubahan *User Interfaces*, beberapa perubahannya yaitu:

- (a) *Third Party Virtual Keyboard* dengan prediksi teks dan *User Dictionary*(Kamus pengguna)
- (b) Mendapat dukungan *Widget*
- (c) Sudah memiliki kemampuan *record video* dan memutar video dengan format *MPRG-4* (.mp4) dan .3gp
- (d) Memiliki fasilitas *pairing bluetooth*
- (e) Mendukung *bluetooth stereo A2DP dan AVRCP*

- (f) Penambahan fitur *Copy and Paste* pada *Web Browser*
- (g) Fitur Menambahkan foto pada kontak telepon
- (h) Pada bagian panggilan, tanggal dan waktu ditampilkan di bagian log panggilan
- (i) Dapat memanggil kontak melalui log panggilan
- (j) Animasi saat terjadi transisi layar
- (k) Fitur *Auto Rotate* (Putar Otomatis)
- (l) Update Animasi saat *Booting* OS
- (m) Dapat *Upload* video ke youtube
- (n) Dapat *Upload* foto ke picasa

4. Android 1.6 **Donut**



Gambar 1.6 Android Donut

Android Versi 1.6 atau dikenal dengan Android *Donut* merupakan versi android yang dirilis pada tanggal 15 September 2009, dan terdapat fitur-fitur tambahan dari versi yang sebelumnya. Android ini memiliki sedikit kesalahan (*bug* atau *error*) pada sistemnya. Selain itu, Android ini memiliki fitur yang cukup banyak yang ditambahkan oleh Google sehingga Android ini terbilang Android dengan versi yang cukup sempurna pada zamannya. Fitur-fitur tambahan pada Android versi ini adalah sebagai berikut:

- (a) Terdapat fitur pencarian suara dan teks dalam *history*, *bookmark*, kontak dan web.
- (b) Fitur untuk menyertakan konten pada *Developer* pada hasil pencarian.
- (c) *Voice machine* yang dapat mengucapkan berbagai bahasa dan membuat Android tertentu dapat mengucapkan teks.
- (d) Pencarian menjadi lebih mudah

- (e) Kemampuan untuk melihat sedikit cuplikan aplikasi di *Android Market*
- (f) Galeri, Kamera dan *Video recorder* yang terintegrasi
- (g) Akses Kamera lebih cepat
- (h) Dapat memilih *multifiles* untuk menghapus foto dalam sekala banyak.
- (i) *Update* pada teknologi *CDMA/EVDO*, *802.1x*, *VPN* dan mesin pengucap teks
- (j) Mendukung resolusi layar *WVGA*
- (k) Peningkatan pencarian dan kamera
- (l) Memperluas kerangka kerja, Gestur dan penambahan *tools GestureBuilder Developer*.

KEKURANGAN

- (a) Hanya aplikasi tertentu yang dapat diinstall disini.
- (b) Tidak ada *equalizer* pada *Music Playernya*.
- (c) Android market tidak terintegrasi
- (d) Keypad yang lambat
- (e) *Touch Responsiveness* kurang.

5. Android 2.0 **Eclair**



Gambar 1.7 Android Eclair

Android versi 2.0 ini bernama android *eclair* yang dirilis pada tanggal 26 Oktober 2009. Android versi ini memiliki beberapa jenis *API level* yaitu:

- (a) Android Eclair 2.0 (API level 5)
Android versi 2.0 ini dirilis pada tanggal 26 Oktober 2009. Pada Android versi ini memiliki beberapa fitur tambahan dari versi-versi sebelumnya. Perubahan fitur pada android versi ini adalah:
 - i. Bluetooth 2.1

- ii. *Multi-touch system*
 - iii. *Live wallpaper*
 - iv. *Flash kamera, Digital Zoom, Skin mode, macro focus*
 - v. *HTML*
 - vi. *Digital Zoom*
 - vii. *Support Microsoft Exchange*
 - viii. Update pada UI *Web Browser* dengan fitur *bookmark, double tap zoom, support HTML5*
 - ix. Multi akun pada saat sinkronisasi menggunakan *E-Mail* dan kontak
 - x. *E-Mail* untuk *Microsoft Exchange* dengan kemampuan untuk mencari *E-Mail* dari beberapa akun dalam satu halaman
 - xi. Dapat memilih foto kontak
 - xii. Opsi untuk memanggil, mengirim sms dan *E-Mail* pada kontak
 - xiii. Mampu mencari SMS dan MMS yang tersimpan. Pesan yang terlalu lama akan terhapus dengan sendirinya jika sudah mencapai batas maksimum.
 - xiv. Kecepatan *keyboard virtual* lebih cepat dan *support* kamus yang mempelajari kata dan nama kontak
 - xv. Menyempurnakan UI dari kalender, menampilkan notifikasi kalender.
 - xvi. Kecepatan *Software* lebih optimal
 - xvii. Resolusi layar lebih beragam
 - xviii. *Update google maps* ke versi 3.1.2
- (b) Android Eclair 2.0.1 (API level 6)
- Android ini dirilis pada tanggal 3 Desember 2009. Versi ini merupakan *update* dari versi sebelumnya (Android Eclair 2.0). *Update* yang terdapat pada versi ini adalah:
- i. Perubahan minor pada API nya
 - ii. Perbaikan pada bug yang terjadi di versi sebelumnya
 - iii. Perubahan *workflow*(Kerangka kerja) pada sistemnya
- (c) Android Eclair 2.1 (API level 7)
- Android ini dirilis pada tanggal 12 Januari 2010. Versi ini merupakan *update* dari versi sebelumnya (Android Eclair 2.0.1). *Update* yang terdapat pada versi ini adalah:
- i. Perubahan Minor pada API nya
 - ii. Perbaikan pada bug yang terjadi di versi sebelumnya

6. Android 2.2.9 **Froyo**



ANDROID 2.2 FROYO

Gambar 1.8 Android Froyo

Android 2.2.9 atau dikenal dengan nama android Froyo, diluncurkan pada Mei 2010. Versi ini dirilis oleh perusahaan besar yaitu Google. Versi ini merupakan versi penyempurna dari versi versi yang sebelumnya. Versi ini dibentuk dengan tujuan untuk meningkatkan kinerja dari sistem Android. Fitur fitur yang terdapat pada versi android ini adalah:

- (a) Peningkatan kecepatan sistem
- (b) Pengimplementasian JIT
- (c) Integrasi mesin dari JavaScript V8 Chrome kedalam *Web Browser*
- (d) Dukungan *Android Cloud to Device Messaging (C2DM)*
- (e) Meningkatkan *Microsoft Exchange Support*, keamanan, pencarian otomatis, GAL, sinkronisasi pada kalender dan pembersihan dari jarak jauh
- (f) Memberikan fitur *Shortcut* untuk launcher terutama *shortcut* pada telepon dan *web browser*.
- (g) USB Tethering
- (h) Fitur untuk mengaktifkan dan menonaktifkan paket data jaringan seluler
- (i) Menambahkan otomatis *update* pada aplikasi *Market*
- (j) dapat membagikan kontak dan panggilan suara melalui bluetooth
- (k) Mendapat dukungan *Bluetooth enable car* dan *desk docks*
- (l) Mendukung *Password alphanumeric*
- (m) Aplikasi untuk mengontrol *space* pada memori atau storage
- (n) Mendukung upload file melalui *web browser*
- (o) Mendukung animasi GIF
- (p) Mendapat dukungan *Adobe Flash*
- (q) Mendukung tampilan PPI (maksimal hingga 320 ppi), misalnya layar 4 inch dengan resolusi 720p

(r) *Zoom* pada galeri

7. Android 2.3 **Gingerbread**



Gambar 1.9 Android Gingerbread

Android 2.3 atau biasa disebut Android *Gingerbread* merupakan versi android yang dikeluarkan pada bulan Desember 2010. Secara fitur, Android ini sudah tergolong sempurna dan ditambah lagi pada android 2.3 ini telah diadopsi oleh sebuah perusahaan yang membuat *Smartphone* yang sangat terkenal dan terpopuler dengan merek Samsung dengan menanamkan OS ini ke dalam *Smartphonenya*. Smartphone yang digunakan adalah smartphone seri *Nexus* dengan menggunakan fitur-fitur seperti:

- (a) Memperbarui desain UI nya dengan mempercepat kerja UI dan menyederhanakannya.
- (b) Resolusi dan layar dibuat menjadi besar (WXGA dan tingkat yang lebih tinggi)
- (c) Dukungan telepon internet SIP VoIP
- (d) Input untuk teks yang lebih cepat dan lebih intuitif pada *Keyboardnya* Cara tersebut ditingkatkan dengan meningkatkan akurasi, *Text Suggestion* dan *input* dengan menggunakan suara
- (e) *Upgrade* fungsi *Copy and Paste* yang memungkinkan pengguna untuk memilih kata dengan menekan layar.
- (f) Dukungan untuk NEar Field Communication (NFC), memungkinkan pengguna untuk membaca NFC yang ada pada poster, stiker dan juga iklan.
- (g) Dukungan bagi Near Field Communication (NFC), memungkinkan pengguna untuk membaca tag NFC yang tertanam dalam poster, stiker, atau iklan
- (h) Efek audio baru seperti reverb, equalizer, virtualisasi penyuar kuping, dan bass boost
- (i) Download Manager baru, memudahkan pengguna untuk mengakses berkas yang diunduh dari penjelajah web, surel, ataupun dari aplikasi lainnya

- (j) Dukungan multi kamera pada perangkat, termasuk kamera depan, jika tersedia
 - (k) Dukungan bagi pemutar video WebM/VP8, dan audio AAC
 - (l) Peningkatan manajemen daya dengan peran lebih aktif dalam mengelola aplikasi yang beroperasi terlalu lama
 - (m) Peningkatan dukungan bagi pengembangan kode asli
 - (n) Peralihan dari YAFFS ke ext4 pada perangkat yang lebih baru
 - (o) Peningkatan kualitas audio, grafis, dan masukan bagi pengembang permainan
 - (p) Dukungan sensor yang lebih banyak (seperti giroskop dan barometer)
8. Android 3.0 - 3.2 **Honeycomb**



Gambar 1.10 Android Honeycomb

Honeycomb adalah salah satu sistem operasi Android versi terbaru yang dirilis pada bulan Februari 2011 silam. Namun, versi ini lebih ditujukan untuk perangkat Tablet yang mana pada tahun itu sangat laris atau laku dipasaran. Beberapa fitur dan perbaikan pada Android Honeycomb, yaitu :

- (a) Support Multi core
- (b) Support Tablet lebih baik
- (c) Updated 3D UI
- (d) Layar Utama (homescreens) yang dapat diatur
- (e) Melihat aplikasi yang barusan dibuka
- (f) Menyempurnakan layout keyboard
- (g) Transport protocol untuk Media atau Picture video chat Google Talk

- (h) Google eBooks
- (i) Private browsing
- (j) System-wide Clipboard
- (k) HTTP Live streaming

Update 3.1:

- (a) Peningkatan UI
- (b) Open Accessory API
- (c) USB host API
- (d) Support mouse, joysticks dan gamepad
- (e) Widget Home screen yang bisa di atur size atau ukurannya
- (f) Notifikasi MTP
- (g) RTP API untuk audio

Update 3.2:

- (a) Optimise pada berbagai tablets
- (b) Mode kompatibilitas display (zoom for fixed sized apps)
- (c) Sinkronisasi Media dari SD card

Update 3.2.1:

- (a) Update Android Market merupakan automatic updates yang lebih mudah
- (b) Update Google Books
- (c) Peningkatan kinerja Wi-Fi
- (d) Perbaikan prediksi tulisan tangan dengan huruf Chinese

Update 3.2.2:

- (a) Perbaikan kecil

Update 3.2.4:

- (a) Update tambahan Pay as you go bagi tablet

Update 3.2.6

- (a) Perbaikan kecil

9. Android 4.0 **Ice Cream Sandwich**

Puncak kesempurnaan Android yakni ketika pada versi ini, dimana Ice Cream Sandwich dirilis pada bulan Oktober 2011 silam. Dan operasi sistem ini mulai bekerja dengan baik di semua jenis smartphone apapun. Selain bertambahnya



Gambar 1.11 Android Ice Cream Sandwich

berbagai fitur yang menarik, Ice Cream Sandwich juga merupakan versi yang paling banyak disukai pada saat itu. Bahkan, Android Ice Cream Sandwich juga sudah dilengkapi dengan fitur ekstra multitasking serta notifikasi yang lebih banyak. Pembaruan pada versi ini antara lain:

- (a) Tombol lunak tablet Android 3.x tersedia bagi penggunaan di telepon pintar
- (b) Pemisahan widget di tab baru, terletak pada layar yang bersebelahan dengan aplikasi
- (c) Pembuatan folder yang lebih mudah, dengan gaya drag-and-drop
- (d) Launcher yang bisa dikustomisasi
- (e) Peningkatan fitur pesan suara visual, dengan kemampuan untuk mempercepat atau memperlambat kecepatan pesan suara
- (f) Fungsi "cubit" untuk memperbesar pada kalender
- (g) Pengintegrasian fungsi cuplikan layar (screenshot) dengan menekan dan menahan tombol daya dan volume-turun secara bersamaan
- (h) Perbaikan kesalahan koreksi pada papan ketik
- (i) Kemampuan untuk mengakses aplikasi secara langsung dari layar kunci (lock screen)
- (j) Perbaikan fungsi salin dan tempel
- (k) Integrasi suara yang lebih baik dan berkesinambungan
- (l) Mode buka kunci identifikasi wajah, fitur yang memungkinkan pengguna untuk membuka perangkat menggunakan perangkat lunak pengenalan wajah
- (m) Penambahan penjelajah web bawaan Chrome, mampu membuka halaman hingga 16 tab
- (n) Sinkronisasi otomatis pada penjelajah web dengan bookmark Chrome pengguna
- (o) Penambahan jenis huruf baru, Roboto
- (p) Penggunaan data bisa dibatasi, pengguna akan diperingatkan jika penggunaan data sudah mendekati batas tertentu, dan menonaktifkan data yang digunakan ketika batas tersebut terlampaui

- (q) Kemampuan untuk mematikan aplikasi yang menggunakan data di latar belakang
- (r) Peningkatan fungsi aplikasi kamera dengan fitur-fitur seperti zero shutter lag, time lapse settings, mode panorama, dan kemampuan untuk memperbesar saat merekam video
- (s) Penambahan aplikasi pengedit foto bawaan
- (t) Tata letak galeri yang baru, bisa dikelola berdasarkan lokasi dan orang
- (u) Pemutakhiran aplikasi "People" dengan integrasi pada jejaring sosial
- (v) Android Beam, fitur komunikasi area dekat yang memungkinkan dilakukannya pertukaran jarak pendek bookmark web, info kontak, arah, video YouTube, dan data lainnya
- (w) Dukungan format gambar WebP
- (x) Merekam video 1080p bagi perangkat Android tertentu
- (y) Modul kernel Android VPN Framework (AVF) dan TUN (bukan TAP). Sebelum versi 4.0, perangkat lunak VPN membutuhkan rooting.

10. Android 4.1.2 **Jelly Bean**



Gambar 1.12 Android Jelly Bean

Jelly Bean dirilis pada 9 Juli 2012 lewat konferensi I/O Google. Versi ini adalah salah satu versi Android yang kerap mendapatkan update fitur-fitur yang bermanfaat dan menarik, beberapa contohnya semacam memperbaiki rotasi layar, seperti Support resolusi video 4K, Support penulisan huruf Hebrew dan Arabic dari kanan ke kiri, peningkatan kinerja, dan sistem keamanan serta masih banyak lainnya. Fitur yang terdapat pada versi ini adalah :

- (a) Antarmuka pengguna yang lebih halus:
- (b) Waktu vsync pada animasi UI dikelola oleh kerangka kerja Android, termasuk reaksi aplikasi, efek sentuh, komposisi layar, dan penyegaran tampilan

- (c) Triple buffering pada grafis
- (d) Peningkatan aksesibilitas
- (e) Teks dua bahasa dan dukungan bahasa lainnya
- (f) Papan ketik yang bisa dimodifikasi oleh pengguna
- (g) Perluasan notifikasi
- (h) Kemampuan untuk mematikan notifikasi pada aplikasi tertentu
- (i) Shortcut dan widget secara otomatis bisa disusun ulang atau diatur ukurannya
- (j) Transfer data Bluetooth bagi Android Beam
- (k) Diktasi suara luring
- (l) Tablet dengan layar kecil bisa menyesuaikan tata letak antarmuka dan layar depan seperti pada telepon pintar
- (m) Peningkatan pencarian suara
- (n) Peningkatan aplikasi kamera
- (o) Google Wallet (pada Nexus 7)
- (p) Foto kontak Google+ resolusi tinggi
- (q) Aplikasi pencarian Google Now
- (r) Audio multi-saluran
- (s) Audio USB (bagi suara eksternal DACs)
- (t) Audio chaining
- (u) Penjelajah web bawaan Android diganti dengan Google Chrome pada perangkat Android pra-instal
- (v) Kemampuan untuk menambahkan widget aplikasi tanpa akses root

11. Android 4.4 **Kitkat**



Gambar 1.13 Android Kitkat

Android versi inilah yang saat ini banyak dipakai oleh mayoritas masyarakat Indonesia. Kitkat dirilis pada tahun 2013 lalu. pada versi ini, Android banyak

mendapatkan pembaharuan/update fitur. Seperti, terdapatnya fitur Screen recording, untuk merekam kegiatan yang terjadi pada layar smartphone, Peningkatan akses notifikasi, New Translucent system UI, System wide settings untuk closed captioning, dan Peningkatan kinerja serta lain sebagainya. Fitur yang terdapat pada versi ini adalah :

- (a) Pembaruan antarmuka dengan bar status dan navigasi transparan pada layar depan.
- (b) Optimasi kinerja pada perangkat dengan spesifikasi yang lebih rendah
- (c) Kerangka kerja pencetakan
- (d) NFC Host Card Emulation sebagai emulator kartu pintar
- (e) WebViews berbasis Chromium
- (f) Perluasan fungsionalitas bagi layanan pendengar notifikasi
- (g) API umum untuk mengembangkan dan mengelola klien pesan teks, kemampuan untuk menentukan aplikasi SMS standar.
- (h) Kerangka kerja baru untuk transisi UI
- (i) Kerangka kerja akses penyimpanan untuk mengambil konten dan dokumen dari sumber lain
- (j) Sensor batching, Step Detector, dan Counter API
- (k) Peningkatan tampilan mode layar penuh, tombol perangkat lunak dan status bar bisa diakses dari tepi dengan cara menggesek
- (l) Penyeimbang audio, pemantauan audio, dan peningkatan suara audio
- (m) Perekam aktivitas layar yang terintegrasi
- (n) Inframerah
- (o) Peningkatan aksesibilitas API
- (p) Mesin virtual eksperimental baru, ART
- (q) Dukungan Bluetooth Message Access Profile (MAP)

12. Android 5.0 Lollipop

Dirilis pada tahun 2014, Android Lollipop lebih banyak menawarkan fitur tambahan untuk menyempurnakan berbagai fitur yang sudah ada. Dan Nexus 6 merupakan salah satu ponsel yang pertama mencicipi Android Lollipop ini. Selain itu, Google juga lebih menyempurnakan pada kinerja dari Android Lollipop sendiri. Fitur yang terdapat pada versi ini adalah :

- (a) Desain antarmuka (tampilan) yang dinamakan "Material Design".
- (b) 64-bit ART compiler
- (c) Project volta, yang berguna untuk meningkatkan daya hidup baterai 30 persen lebih tahan lama.



Gambar 1.14 Android Lollipop

- (d) â€œfactory reset protectionâ€œ. Fitur ini berguna ketika smartphone hilang, ia tidak bisa direset ulang tanpa memasukkan id google dan kata sandi (password).

13. Android 6.0 **Marshmallow**



Gambar 1.15 Android Marshmallow

Android versi 6.0 dirilis pada tahun 2015 silam, yang banyak membawa pembaharuan. Salah satunya yaitu suda support USB Type-C. Selain itu, Android Marshmallow ini juga terdapat fasilitas autentikasi sidik jari dan daya baterai yang lebih baik.

Android Marshmallow memperkenalkan model izin yang didesain ulang: sekarang ada hanya delapan kategori izin, dan aplikasi yang tidak lagi secara otomatis diberikan semua hak akses mereka ditentukan pada waktu instalasi. Sebuah sistem opt-in sekarang digunakan, di mana pengguna akan diminta untuk memberikan atau menolak izin individu (seperti kemampuan untuk mengakses kamera atau mikrofon) untuk aplikasi ketika mereka dibutuhkan. Aplikasi mengingat hibah izin mereka, dan mereka dapat disesuaikan oleh pengguna setiap saat.

Model izin baru akan digunakan hanya oleh aplikasi yang dikompilasi untuk Marshmallow menggunakan kit pengembangan perangkat lunak (SDK) tersebut, sementara semua aplikasi lainnya akan terus menggunakan model izin sebelumnya.

Marshmallow juga memiliki skema manajemen daya baru bernama Doze yang mengurangi tingkat aktivitas aplikasi latar belakang saat perangkat menentukan bahwa itu tidak sedang aktif ditangani oleh pengguna, yang, menurut Google, menggandakan pemakaian baterai perangkat. Hal ini juga memperkenalkan pilihan untuk mengatur ulang semua pengaturan jaringan, tersedia untuk pertama kalinya pada Android, yang membersihkan pengaturan terkait jaringan untuk WI-FI, Bluetooth dan koneksi seluler.

Android Marshmallow memberikan dukungan asli untuk pengenalan sidik jari, memungkinkan penggunaan sidik jari untuk membuka perangkat dan otentikasi Play Store dan pembelian Android Pay; API standar juga tersedia untuk melaksanakan otentikasi berbasis sidik jari dalam aplikasi lain. Android Marshmallow mendukung USB Type-C, termasuk kemampuan untuk menginstruksikan perangkat untuk mengisi daya perangkat lain melalui USB. Marshmallow juga memperkenalkan "pranala yang diverifikasi" yang dapat dikonfigurasi untuk membuka langsung dalam aplikasi tertentu mereka tanpa petunjuk pengguna lanjut.

Versi API Android yang disediakan oleh Marshmallow adalah 23. Alat pengembangan Android Marshmallow tersedia di Pengelola SDK di bawah tingkat API 23.

14. Android 7.0 **Nougat**

Android Nougat versi 7.0 dirilis pada bulan Agustus 2016 yang lebih meningkatkan pada kinerja versi sebelumnya. Selain itu, Android Nougat juga menambah banyak fitur-fitur baru yang diantaranya seperti sudah dapat multitasking, meningkatkan fitur Doze yang dahulu telah dirilis di versi sebelumnya. Inilah beberapa fitur terbaru yang terdapat pada versi Nougat :

- (a) Support Multi window
- (b) Dapat langsung membalas pesan dari menu notifikasi atau jendela.
- (c) Tampilan panel notifikasi serta quick settings yang baru.
- (d) Mode Doze yang lebih baik, (Doze Mode 2.0)
- (e) Menu di antara system settings.

15. Android 8.0 **Oreo**

Android versi Oreo dirilis pada bulan Agustus 2017 lalu. Tentu saja Android Oreo merupakan versi final untuk sekarang ini. Beberapa fiturnya juga turut



Gambar 1.16 Android Nougat



Gambar 1.17 Android Oreo

diluncurkan Google selaku pihak pengelola. Adapun fitur-fiturnya tersebut antara lain yaitu :

- (a) Android O lebih berfokus pada kecepatan dan efisiensi
- (b) Kecepatan Boot up 2X lebih cepat
- (c) Mode Picture in picture lebih flexibel
- (d) Aplikasi yang berjalan di latarbelakang atau background lebih diperketat untuk lebih menghemat battery
- (e) Battery lebih tahan lama
- (f) Emoji yang diperbaharui dan diperbanyak

BAB 2

ANDROID STUDIO

Pertama kali muncul Android Inc merupakan sebuah perusahaan software kecil yang didirikan pada bulan Oktober 2003 di Palo Alto, California, USA. Perusahaan ini dibangun oleh beberapa senior di beberapa perusahaan yang berbasis IT dan Communication, Andy Rubin, Rich Miner, Nick Sears dan Chris White. Rubin menyatakan bahwa, Android Inc Didirikan untuk mewujudkan mobile device yang lebih fleksibel terhadap lokasi dan preferensi pemilik. Sehingga, Android Inc ingin mewujudkan mobile device yang lebih mengerti pemiliknya selain karena OS nya yang open source. Berawal dari konsep inilah Android Inc ternyata menarik minat Google untuk memilikinya. Maka, pada bulan Agustus 2005, Akhirnya Android Inc diakuisisi oleh Google Inc. dan seluruh sahamnya dibeli oleh Google.

Perusahaan milik Andy Rubin, Rich Miner, Nick Sears dan Chris White tetap di Android Inc yang dibeli Google, sehingga akhirnya mereka pun ikut menjadi bagian dari raksasa Google dan sejarah Android. Disini mereka mulai menggunakan platform Linux untuk membuat sistem operasi bagi mobile phone. Dari sinilah akhirnya banyak pengembang sistem maupun software yang mengembangkan maupun merancang sistem Android menggunakan software *Android Studio* yang support dengan Android, Contohnya ialah : Android Studio.



Gambar 2.1 Android Studio

Berikut akan diulas beberapa alasan mengapa memahami Android Studio itu penting untuk dilakukan :

1. Dengan mempelajari Android Studio dapat membantu Anda untuk mempercepat pembuatan aplikasi yang Anda inginkan.
2. Android Studio merupakan sebuah tools yang mudah dipahami dan digunakan.
3. Dalam satu tools ini Anda bisa mendapatkan berbagai manfaat mulai dari pembuatan aplikasi hingga testing aplikasi.
4. Belajar Android Studio maka Anda bisa menghemat waktu kerja untuk dapat lebih produktif.
5. Dapat memperdalam ilmu codingan dengan baik. Karena dalam android studi diberikan beberapa referensi ketika Anda mengetik sintaks. Dengan begitu tentunya Anda akan mencari tahu apa saja kegunaan dari sintaks yang terdapat.
6. Sarana pembelajaran coding dan pembuatan aplikasi yang baik dan praktis hanya dengan Android Studio.

2.1 Android Studio

Pertama kali Android Studio diumumkan di Google I/O Conference pada tahun 2013 dan dirilis ke publik pada tahun 2014. Sebelum lahirnya Android Studio, aplikasi pada Android dikembangkan dengan Eclipse IDE yaitu IDE Java. Setelah

adanya android studio yang open source dapat memudahkan bagi Anda yang ingin membuat aplikasi dengan Android Studio.

Android dapat menyediakan interface untuk Anda dalam membuat aplikasi serta mengelola manajemen filen aplikasi anda. Untuk bahasa programman anda gunakan adalah Java. Dalam Android Studio, anda hanya tinggal menulis, mengedit, menyimpan dan testing project beserta dan file lainnya yang ada dalam project itu hanya dengan android studio.

Tidak hanya itu, keunggulan menggunakan Android Studio juga memberi Anda akses ke Android Software Development Kit (SDK). SDK adalah sebuah ekstensi dari kode Java yang memperbolehkannya untuk berjalan dengan mulus di device Android. Untuk, Java nya dibutuhkan untuk menulis program, Android SDK sangat diperlukan untuk menjalankan programnya di Android. Maka dari itu dengan menggabungkan keduanya, Anda memerlukan Android Studio. Sehingga ketika Anda menemukan bug pada aplikasi Anda, Anda bisa mengetahui bug tersebut dengan menggunakan Android Studio untuk memperbaikinya.

Berikut ini adalah beberapa fitur Android Studio:

1. Environment yang mempermudah Anda untuk mengembangkan aplikasi untuk Android
2. Support dalam mengembangkan aplikasi Android TV dan Android Wear
3. Template untuk menentukan design dan komponen Android
4. Editor layout dengan interface drag-and-drop
5. Refactoring dan perbaikan cepat khusus Android
6. Dukungan build berbasis Gradle
7. Integrasi ProGuard
8. Emulator yang cepat dan berbagai fitur didalamnya
9. Dapat terintegrasi dengan Google Cloud Messaging dan App Engine
10. Dukungan program basic C++ dan NDK

Android Studio adalah Lingkungan Pengembangan Terpadu (Integrated Development Environment/IDE) resmi untuk pengembangan aplikasi Android, yang didasarkan pada IntelliJ IDEA. Selain sebagai editor kode dan fitur developer IntelliJ yang andal, Android Studio menawarkan banyak fitur yang meningkatkan produktivitas Anda dalam membuat aplikasi Android, seperti:

1. Sistem build berbasis Gradle yang fleksibel
2. Emulator yang cepat dan kaya fitur
3. Lingkungan terpadu tempat Anda bisa mengembangkan aplikasi untuk semua perangkat Android

4. Terapkan Perubahan untuk melakukan push pada perubahan kode dan resource ke aplikasi yang sedang berjalan tanpa memulai ulang aplikasi
5. Template kode dan integrasi GitHub untuk membantu Anda membuat fitur aplikasi umum dan mengimpor kode sampel
6. Framework dan fitur pengujian yang lengkap
7. Fitur lint untuk merekam performa, kegunaan, kompatibilitas versi, dan masalah lainnya
8. Dukungan C++ dan NDK
9. Dukungan bawaan untuk Google Cloud Platform, yang memudahkan integrasi Google Cloud Messaging dan App Engine

Setiap project di Android Studio berisi satu atau beberapa modul dengan file kode sumber dan file resource. Jenis modul meliputi:

1. Modul aplikasi Android
2. Modul library
3. Modul Google App Engine

Secara default, Android Studio menampilkan file project Anda dalam tampilan project Android, seperti yang ditunjukkan. Tampilan ini disusun menurut modul untuk memberikan akses cepat ke file sumber utama project Anda. Semua file build terlihat di tingkat teratas di bagian **Gradle Script** dan setiap modul aplikasi berisi folder berikut:

1. manifests: Berisi file AndroidManifest.xml.
2. java: Berisi file kode sumber Java, termasuk kode pengujian JUnit.
3. res: Berisi semua resource non-kode, seperti tata letak XML, string UI, dan gambar bitmap.

Struktur project Android pada disk berbeda dengan representasi tersatukan ini. Untuk melihat struktur file project sebenarnya, pilih **Project** dari menu drop-down Project.

Anda juga dapat menyesuaikan tampilan file project untuk berfokus pada aspek spesifik dari pengembangan aplikasi Anda. Misalnya, memilih tampilan **Problems** pada project Anda akan menampilkan link ke file sumber yang berisi error coding dan sintaks yang dikenali, seperti tag penutup elemen XML yang tidak ada dalam file tata letak.

2.1.1 Langkah Download Android Studio

Cara mendownload Android studio cukup mudah yaitu dengan <https://developer.android.com/studio> yang merupakan laman website resmi Android dan terdapat SDK berbagai macam jenis didalamnya. Tetapi untuk menjalankan Android Studio Anda juga perlu mendownload Java Development Kit dengan <https://www.oracle.com/technetwork/java/javase/downloads-2133151.html>.

Berikut ini adalah syarat instalasi untuk berbagai sistem operasi : Windows OS

1. Microsoft Windows 7/8/10
2. Minimum RAM 2GB, direkomendasikan Anda menggunakan RAM 8GB
3. Minimum space disk tersedia 2GB, tetapi Anda direkomendasikan menyediakan 4GB (500MB untuk IDE, 1,5GB untuk Android SDK, dan emulator sistem gambar)
4. Resolusi minimum 1280 800
5. Java Development Kit 8

MAC OS

1. MAC OS X 10.8.5 atau lebih sampai dengan 10.11.4 (El Capitan)
2. Minimum RAM 2GB, direkomendasikan Anda menggunakan RAM 8GB
3. Minimum space disk tersedia 2GB, tetapi Anda direkomendasikan menyediakan 4GB (500MB untuk IDE, 1,5GB untuk Android SDK, dan emulator sistem gambar)
4. Resolusi minimum 1280 800
5. Java Development Kit 6

LINUX OS

1. Desktop GNOME atau KDE
2. 64-bit distribution yang bisa menjalankan aplikasi 32-bit
3. GNU C Library (glibc) 2.11 atau versi selanjutnya
4. Minimum RAM 2GB
5. Minimum space disk tersedia 2GB, tetapi Anda direkomendasikan menyediakan 4GB
6. Resolusi minimum 1280 800
7. Java Development Kit 8

2.1.2 Cara Install Android Studio

Pertama sebelum anda menginstall Android Studio, Anda harus terlebih dahulu menginstal Java Development Kit-nya. Caranya ialah Anda tinggal membuka installer Java Development Kit yang sudah ada mengunduh sebelumnya, kemudian selanjutnya ikuti langkah yang mereka tunjukkan.

Setelah itu, Anda sudah bisa menginstall Android Studio dengan mengikuti langkah di bawah ini:

1. Buka installer Android Studio yang sudah ada unduh. Kemudian klik Next.
2. Setelah itu, muncul jendela baru yang memberikan Anda beberapa pilihan komponen apa saja yang ingin Anda install beserta versi android nya. Lalu klik Next.
3. Selanjutnya Anda akan melihat License Agreement, pilih I Agree
4. Setelah itu, Anda akan melihat pilihan lokasi penyimpanan file. Anda tidak perlu mengubah directory yang sudah mereka pilih. Anda tinggal klik Default dan file Anda akan disimpan ke directory yang sudah mereka sediakan. Klik Next dan di layar selanjutnya klik Install.
5. Setelah proses instalasinya selesai klik Next. Kalau sudah, Anda akan melihat windows "Completing Android Studio Setup". Anda tidak perlu mengubah pilihan Start Android Studio dan langsung saja klik Finish.
6. Setelah itu, Anda akan melihat jendela baru dengan 2 pilihan. Checklist pilihan kedua jika kalian belum pernah menginstall IDE Android Studio sebelumnya dan pilih OK.
7. Setelah itu Anda akan melihat layar WELCOME dari Android Studio dan klik Next.
8. Pilih Standard dan klik Next
9. Anda kemudian akan melihat jendela SDK Component Setup. Pilih komponen yang ingin Anda install dan klik Next. Pada layar selanjutnya klik Finish.
10. Anda kemudian akan melihat layar Downloading Component.
11. Setelah unduhan Anda selesai, proses instalasi Android Studio telah selesai. Anda tinggal klik Finish. Kemudian Anda akan melihat jendela Welcome to Android Studio.

BAB 3

JAVA

3.1 Sejarah Java



Gambar 3.1 Java

Java merupakan bahasa pemrograman yang dapat berjalan pada perangkat komputer maupun telepon genggam. Bahasa ini dirilis pada tahun 1995 yang dibuat oleh

James Gosling saat berada di Sun Microsystems yang saat ini merupakan bagian dari Oracle. bahasa pemrograman ini mengadopsi penulisan yang terdapat pada C dan C++ yang dikembangkan lagi menjadi sintaksis yang lebih sederhana. Aplikasi Dengan basis Java umumnya dikompilasi ke dalam bytecode Dan dapat dijalankan pada mesin virtual Java (JVM). Java didesain secara Khusus untuk memanfaatkan implementasi dependensi dengan sangat minimal. Dengan fungsionalitasnya untuk berjalan di beberapa platform sistem operasi yang berbeda sehingga Java memiliki slogan Write once, run anywhere (WORA) atau bisa juga Write once, run everywhere (WORE). Java adalah bahasa pemrograman yang dapat digunakan untuk berbagai macam tujuan. bahasa pemrograman concurrent, berbasis kelas, dan berorientasi objek. Bahasa ini dirancang secara khusus untuk mengurangi ketergantungan ketika bahasa ini di terapkan. Hal ini yang dimaksudkan dalam slogannya "Write Once, Run Anywhere (WORA)". Slogan ini memiliki arti bahwa kode yang dijalankan pada suatu platform tidak perlu dikonfirmasi lagi pada platform lainnya. Java merupakan bahasa pemrograman yang paling populer digunakan dan dimanfaatkan dalam pengembangan berbagai jenis perangkat lunak aplikasi ataupun aplikasi lainnya.

Awalnya bahasa pemrograman ini diberi nama Oak, yang Merupakan nama sebuah pohon yang dapat dilihat oleh Gosling dari jendela kantornya. Kemudian dia memberi nama pemrograman ini dengan nama Green, sehingga akhirnya dia mengganti namanya menjadi Java. Nama ini diambil dari kopi jawa yang banyak dikonsumsi oleh pencipta bahasa pemrograman ini.

Ketika World Wide Web menjadi populer, Banyak kelebihan yang membuat bahasa ini dapat digunakan dengan baik dan cocok pada proyek maupun alat untuk adaptasi ke web. Pengembang bahasa ini merancang cara agar program berjalan aman pada halaman web dan hingga akhirnya dipilihlah nama yang baru untuk bahasa yaitu Java. Hingga sekarang Java menjadi bahasa pemrograman yang sangat populer digunakan. Aplikasi yang paling populer digunakan adalah aplikasi web client-server dengan 10 juta pengguna.

3.2 Sejarah perkembangan

Pada awalnya pemrograman Java berawal dari The Green Project yang berjalan untuk 18 bulan. Project ini dimulai pada 1991 hingga musim panas 1992. Proyek tersebut belum menggunakan versi Oak . Proyek ini beranggotakan Patrick Naughton, Mike Sheridan, dan James Gosling, dan sembilan programmer lainnya dari Sun Microsystems.

Proyek ini berlangsung di gedung perkantoran di Sand Hill Road, Menlo Park. Pada saat musim panas 1992 proyek ini ditutup dengan menghasilkan sebuah program Java Oak yang ditujukan sebagai pengendali peralatan dengan teknologi touch-screen seperti sekarang. Teknologi ini diberi nama "*7" (Star Seven).

Setelah era Star Seven, perusahaan TV kabel tertarik untuk menambahkan beberapa orang dari proyek The Green Project. Mereka bekerja pada kantor di 100 Hamilton Avenue, Palo Alto.

Dengan berjalannya waktu, perusahaan ini semakin maju. Jumlah karyawan yang meningkat dengan sangat signifikan dari 13 orang menjadi 17 orang. Pada saat itu, internet menjadi peran yang menjembatani ide di antara mereka. Pada awal tahun 1990, Internet belum terkenal seperti sekarang sehingga yang dipakai hanyalah untuk akademisi dan militer. Pada maret 1995, source code Java versi 1.0a2 dibuka.

Ketika itu, pada pukul 04.00 di ruangan hotel Sheraton Pallace terjadi perpecahan di antara anggota proyek The Green Project. Tiga pimpinan proyek itu, Eric Schmidt, George Paolini, dan Masrc Andreessen dari Sun Microsystems membentuk perusahaan Netscape.

3.3 Struktur Program

Struktur program pada bahasa Java dibagi menjadi 4:

1. Deklarasi Package

Package merupakan sebuah folder yang diisi dengan sekumpulan program Java. Deklarasi package biasanya dilakukan saat membuat aplikasi atau membuat program.

Contoh deklarasi package:

```
1 package com.proyek.qeuangans ;  
2
```

Pada contoh diatas **com.proyek** merupakan domain dari program proyek. Pada saat membuat program, kita boleh untuk tidak mendeklarasikan package, tetapi saat bagian produksi aplikasi kita wajib menuliskan package.

2. Import Library

Bagian ini melakukan import library yang dibutuhkan oleh program. Library adalah kumpgulan dari class dan fungsi yang bisa kita gunakan ketika membuat program.

Contoh Import Library:

```
1 import java.util.Scanner ;  
2
```

Pada contoh diatas, itu berarti kita mengimport class **Scanner** dari sebuah package yang bernama **java.util**

3. Class

Java merupakan bahasa pemrograman yang berbasis OOP (Object Oriented Programming) yang pada setiap program harus dibungkus di dalam class agar dapat dibuat menjadi objek. Class dibuka dengan kurung buka kurawal { dan kurung tutup kurawal }. Dalam blok class dapat diisi dengan method, fungsi - fungsi dan variable.

Contoh Class:

```

1 class NamaProgram {
2     public static void main(String args[]){
3         System.out.println("Halo Dunia!");
4     }
5 }
6

```

Pada contoh program class tersebut terdapat method **main()**. Method main ini akan di eksekusi pertama kali.

4. Method Main

Method main() merupakan blok program yang pertama kali dieksekusi. Method main() pada java wajib dibuat. Tanpa method main ini, maka program tidak bisa dieksekusi. Method main() memiliki parameter args[]. Parameter ini akan menyimpan sebuah nilai dari sebuah argumen di command line.

Contoh method main:

```

1 public static void main(String args[]){
2     System.out.println("Halo Dunia!");
3 }
4

```

Fungsi **String args[]** pada baris program diatas adalah sebagai parameter yang berupa array yang memiliki tipe data string. Parameter - parameter tersebut akan menampung argumen yang akan diberikan ke program.

3.3.1 Statement dan Ekspresi pada Java

Statement dan ekspresi merupakan bagian yang terkecil dalam suatu program. Setiap statement dan ekspresi pada pemrograman java harus diakhiri dengan tanda titik koma (;). Statemen dan ekspresi akan menjadi instruksi yang akan dikerjakan oleh komputer.

Contoh Statement dan Ekspresi:

```

1 System.out.println("Hello World");
2 System.out.println("Apa kabar?");
3 var x = 3;
4 var y = 8;
5 var z = x + y;

```

3.3.2 Blok Program Java

Blok program adalah kumpulan dari beberapa statement dan ekspresi yang dibungkus menjadi satu. Pada blok program ini harus selalu dibuka dengan kurung buka kurawal ({) dan ditutup dengan kurung tutup kurawal (}) Blok program dapat juga berisi blok program yang lain atau blok program didalam blok program.

```

1 // blok program method main
2 public static void main(String args[]){
3     System.out.println("Halo Dunia!");

```

```

4   System.out.println("Halo para pembaca buku ini");
5
6   // blok program percabangan if
7   if( true ){
8       System.out.println('True');
9   }
10
11  // blok program perulangan for
12  for ( int i = 0; i<10; i++){
13      System.out.println("Perulangan ke"+i);
14  }
15 }

```

3.3.3 Penulisan Komentar pada Java

Dalam sebuah program, akan ada bagian yang tidak akan pernah di seksesekusi ketika program dijalankan, bagian tersebut adalah komentar.

Penulisan komentar pada java sama seperti pada bahasa C. Java dapat mendefinisikan beberapa macam komentar yang diantaranya:

- Single-Line comment

Single-Line comment merupakan penulisan komentar yang digunakan hanya untuk satu baris perintah saja. Biasanya komentar ini menggunakan garis miring ganda (*//*). Contoh penggunaan Single-Line comment:

```

1 public static void main(String args[]){
2     // ini adalah komentar satu baris
3 }
4

```

- Multiline comment

Multiline comment biasanya digunakan untuk banyak baris perintah atau beberapa baris perintah. Biasanya komentar ini menggunakan garis miring bintang (*/*...*/*). Contoh penggunaan Multiline comment:

```

1 public static void main(String args[]){
2     /* awal penggunaan multiline comment
3     for ( int i = 0; i<10; i++){
4         System.out.println(" Perulangan ke"+i);
5     }
6     akhir penggunaan multiline comment */
7 }
8 }
9

```

- Documentation comment

Documentation ini digunakan untuk menghasilkan file HTML dari program yang kit buat. Biasanya komentar ini dimulai dengan menggunakan tanda (*/** */*). Contoh penggunaan Documentation comment:

```

1 /**

```



```

2  *
3  *
4  *
5  * @author D4TI2B (1184099 dan 1184094)
6  * @since 2020
7  *
8  *
9  */
10 public static void main(String args[]){
11
12 }
13

```

Komentar biasanya digunakan untuk beberapa hal:

1. Memberi keterangan pada baris kode program
2. Menonaktifkan fungsi atau method tertentu
3. Membuat dokumentasi
4. dll.

3.3.4 String dan Character

String merupakan kumpulan dari Character. Biasanya kita mengetahuinya dengan nama teks. Contoh String:

```
1 "Hello World"
```

Penulisan string pada Java wajib diapit oleh tanda petik ganda seperti pada contoh yang tertulis diatas. Jika diapit dengan tanda petik tunggal, maka dibuat menjadi character. Contoh penggunaan:

```
1 'Hello world'
```

3.3.5 Case Sensitive

Java memiliki sifat Case Sensitive, case sensitive disini berarti perbedaan huruf kapital dan huruf kecil harus diperhatikan karena banyak orang yang sering melakukan kesalahan dalam hal ini. Karena masih banyak orang yang belum bisa membedakan mana variabel yang menggunakan huruf besar dan huruf kecil.

3.4 Mengenal Tipe Data Dasar di Java

Tipe data merupakan bagian penting dalam pemrograman java. Tipe data biasanya akan digunakan dalam bentuk variabel variabel. Java memiliki banyak tipe data dasar atau yang biasa dikenal tipe data primitif pada java. Tipe data dasar atau primitif dibagi menjadi empat bagian:

1. Integer

Integer merupakan tipe data untuk data bilangan bulat. Tipe data integer terbagi lagi menjadi empat bagian yaitu:

- byte
- short
- Int
- Long

Semakin besarnya tipe data maka akan semakin besar pula nilai yang dapat ditampung oleh tipe data tersebut.

2. Floating Point

Floating point merupakan tipe data rasional, tipe data ini dibagi menjadi dua bagian yaitu:

- Float
- Double

3. Char

Char merupakan simbol dari sebuah character. Terdapat kurang lebih 256 simbol yang terdaftar dalam kode ASCII.

ASCII (American Standard Code for Information Interchange) merupakan Kode Standar Amerika untuk Pertukaran Informasi atau sebuah standar internasional dalam pengkodean huruf dan simbol secara universal. [1]

4. Boolean Boolean merupakan tipe data yang paling sederhana dalam pemrograman karena tipe data ini hanya memiliki dua jenis saja yaitu:

- True
- False

Nilai dari boolean ini, sering digunakan untuk mengatur alur dari jalannya program. Boolean lebih sering digunakan untuk percabangan pada program. [2]

Berikut ini jangkauan nilai yang dapat diterapkan pada tipe data primitif diatas:

Type Data	Besar Storage	Nilai Minimal	Nilai Maksimal
Byte	8 bit (1 byte)	-128	127
Short	16 bit (2 byte)	-32768	32767
Int	32 bit (4 byte)	-2147483648	2147483647
Long	64 bit (8 byte)	-9223372036854775808	9223372036854775807
Float	32 bit (4 byte)	3.4 E-38	3.4 E+38
Double	64 bit (8 byte)	1.7 E-308	1.7 E+308
Char	16 bit (2 byte)	\u0000	\uffff

Tabel 3.1 Jangkauan nilai tipe data primitif

Tipe data tersebut dapat dibuat menjadi array. Tanda array adalah dengan adanya kurung siku "[]" pada saat setelah mengetik tipe data yang digunakan.

3.5 Variabel

Variabel adalah tempat untuk menampung sejumlah value pada memori. Jika dianalogikan, Variabel dapat dimisalkan seperti sebuah ruangan.

Berikut ini adalah beberapa para ilmuwan yang memberikan pengertian dari variabel :

1. F.N Kerlinger

Pengertian variabel menurut F.N Kerlinger merupakan suatu konsep yang memiliki macam-macam nilai dari suatu konsep yang dapat di rubah. Sehingga konsep tersebut akan mendapatkan titik kesimpulan yang tepat dan terbaik. [3]

2. Sutrisno Hadi

Variabel merupakan variasi dari objek penelitian, seperti tinggi badan manusia yang divariasikan dengan berat badan maupun usia yang dimiliki. Sehingga menghasilkan nilai kuantitatif dari suatu penelitian yang diterapkan secara real atau nyata.

3. Sugiono

Pengertian Variabel dari Sugiono merupakan segala sesuatu yang diproses melalui informasi tentang suatu hal dari penelitian untuk dipelajari dan mendapatkan hasil dari penelitian tersebut. Yang mana akan ada kesimpulan dari proses penelitiannya.

4. Freddy Rankuti

Freddy Rankuti menerapkan variabel dengan artian suatu konsep yang memiliki nilai bervariasi. Yang mana nilai tersebut dibagi menjadi 4 data yang berbeda. Seperti rasio, skala, ordinal, nominal dan internal.

5. Suharsimi Arikunto
Variabel merupakan objek penelitian yang menjadi perhatian pada suatu titik objek penelitian. Yang nantinya akan mendapatkan nilai dari kesimpulan suatu proses.
6. Bagja Waluya
Konsep yang tidak pernah ketinggalan dalam setiap eksperimen yang dilakukan oleh seseorang. Dari eksperimen tersebut akan menghasilkan suatu data yang berguna sebagai bukti otentik suatu penelitian.
7. Moh. Nazir
Berikutnya, mengenai pengertian variabel menurut Moh. Nazir adalah suatu konsep yang memiliki bermacam-macam nilai yang nyata. Dalam suatu penelitian yang menghasilkan garis besar dari adanya nilai kualitas dan kuantitas.
8. Sugiarto
Menurut Sugiarto variabel adalah suatu karakter yang dapat di observasi dari unit amatan yang merupakan pengenalan atau atribut dari anggota kelompok. Maksud dari variabel ini adalah terjadinya proses variasi antara objek satu dengan objek yang lain. Yang mana aturan masing-masing kelompok memiliki perbedaan variasi.
9. Tri Mutiara
Suatu proses yang berjalan dengan baik hingga mendapat perhatian dengan fokus pada pengaruh nilai yang value. Itulah pengertian variabel menurut Tri Mutiara. Yang mengartikan variabel sebagai cara terbaik mendapatkan hasil penelitian.
10. Bhisma Murti
Definisi variabel menurut bhisma murti adalah adanya fenomena yang memiliki variasi nilai pada sebuah observasi. Yang mana variasi nilai itu dapat di kukur dengan cara kualitatif dan kuantitatif. Sehingga menghasilkan data yang benar dan tepat.
11. Dr Ahmad Watik Pratiknya
Konsep yang memiliki variabilitas dengan penggambaran suatu abstraksi dari fenomena tertentu. Yang mana konsep tersebut berupa data seperti asal kepemilikan ciri yang bervariasi, inilah yang disebut variabel.

Variabel pada java dibagi menjadi dua yaitu:

- Variabel Lokal
Variabel lokal merupakan variabel yang hanya dapat diakses pada lingkup khusus saja. Variabel ini biasanya hanya bisa diakses pada prosedur atau fungsi dimana prosedur tersebut di deklarasikan.
- Variabel Global
Variabel Global merupakan variabel yang dapat diakses melalui semua lingkup

dalam program yang dibuat. Variabel Global ini dapat digunakan oleh semua fungsi dan prosedur.

Untuk mendeklarasikan variabel, Kita harus menulis terlebih dahulu tipe data dari variabelnya, kemudian nama variabel, setelah itu kita wajib menginisiasi variabel tersebut agar tidak terjadi error.

Berikut merupakan standar-standar dalam penulisan variabel:

1. Nama variabel biasanya diawali dengan huruf atau garis bawah. Contoh:

```
1 nama = "Nama Kalian";
2 _nama = "Nama Kalian";
3 namaKu = "Nama Kalian";
4 nama_variabel = "Nama Kalian";
```

2. Karakter selanjutnya pada variabel dapat berupa huruf, garis bawah maupun angka. Contoh:

```
1 __nama = "Nama Kalian"
2 namal = "Nama Kalian"
3 x1 = "Nama Kalian"
```

3. Nama variabel tidak boleh diawali dengan angka. Contoh:

```
1 /* Penggunaan variabel ini salah karena diawali dengan angka */
2 3nama = "Nama Kalian"
```

4. Penulisan variable bersifat Case-Sensitive (huruf besar dan huruf kecil sangat berpengaruh), contoh:

```
1 NAMA = "Nama Kalian"
2 nama = "Nama Kalian"
```

Variable nama dan NAMA pada contoh diatas merupakan variabel yang berbeda, dan memiliki arti yang berbeda.

5. Nama variabel yang digunakan tidak boleh menggunakan kata kunci yang ada pada bahasa pemrograman java (Reserved Word), contoh:

```
1 /* Penggunaan Variable seperti ini salah karena menggunakan
   reserved word pada java. */
2 if = "Nama saya: "
3 else = "Nama anda: "
4 while = "Nama kami: "
```

3.6 Dasar - dasar Pemrograman

3.6.1 Operator

Pada setiap pemrograman, operator merupakan peran penting karena tanpa adanya operator maka program tidak akan bisa berjalan dengan sempurna. Pada java, operator dibagi menjadi 6 jenis operator yaitu:

- Operator Aritmatika
- Operator Penugasan
- Operator Logika
- Operator Pembandingan
- Operator Bitwise
- Operator Ternary

Dari keenam operator tersebut tentunya akan kami jabarkan satu per satu.

1. Operator Aritmatika

Operator aritmatika digunakan untuk melakukan operasi aritmatika.

Nama	Simbol
Penjumlahan	+
Pengurangan	-
Perkalian	*
Pembagian	/
Sisa bagi (Mod/Modulus)	%

Tabel 3.2 Operator Aritmatika

Contoh penggunaan operator aritmatika pada program:

```

1 // Penjumlahan
2 total = angka1 + angka2;
3 System.out.println("Total = " + total);
4
5 // Pengurangan
6 total = angka1 - angka2;
7 System.out.println("Total = " + total);
8
9 // Perkalian
10 total = angka1 * angka2;
11 System.out.println("Total = " * total);
12
13 // Pembagian
14 total = angka1 / angka2;
15 System.out.println("Total = " + total);
16
17 // Modulus (sisa bagi)
18 total = angka1 % angka2;
19 System.out.println("Total = " + total);
20

```

2. Operator Penugasan

Operator penugasan atau biasa disebut *Assignment Operator* digunakan untuk memberikan sebuah tugas pada variable tertentu untuk mengisi nilai. Operator penugasan terdiri dari: Contoh penggunaan operator penugasan:

Nama	Simbol
Pengisian Nilai	=
Pengisian dan Penambahan	+=
Pengisian dan Pengurangan	-=
Pengisian dan Perkalian	*=
Pengisian dan Pembagian	/=
Pengisian dan Sisa bagi	%=

Tabel 3.3 Operator Penugasan

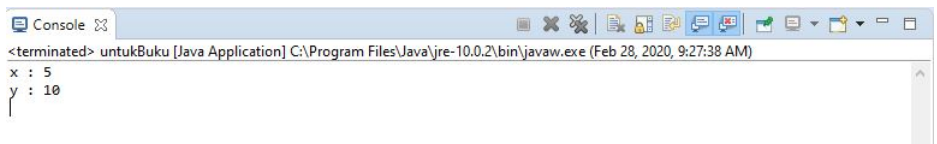
▪ Pengisian Nilai

```

1 int x;
2 int y;
3
4 // Pengisian nilai
5 x = 5;
6 y = 10;
7 System.out.println("x : " + x);
8 System.out.println("y : " + y);
9

```

Hasil ketika program dijalankan:



Gambar 3.2 Hasil Pengisian Nilai Operator Penugasan

▪ Penambahan

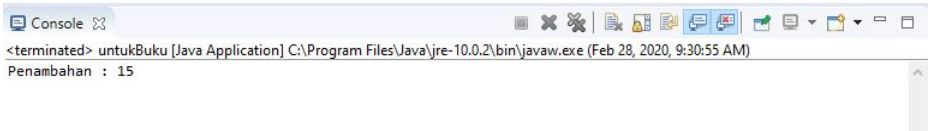
```

1 int x;
2 int y;
3 x = 5;
4 y = 10;
5
6 // penambahan
7 y += x;
8

```

```
9 // Hasil y menjadi 15
10 System.out.println("Penambahan : " + y);
11
```

Hasil ketika program dijalankan:

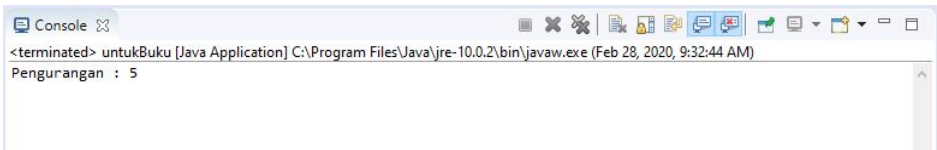


Gambar 3.3 Hasil Penambahan Operator Penugasan

▪ Pengurangan

```
1 int x;
2 int y;
3 x = 5;
4 y = 10;
5
6 // Pengurangan
7 y -= x;
8
9 // Hasil y menjadi 5
10 System.out.println("Pengurangan : " + y);
11
```

Hasil ketika program dijalankan:



Gambar 3.4 Hasil Pengurangan Operator Penugasan

▪ Perkalian

```
1 int x;
2 int y;
3 x = 5;
4 y = 10;
5
6 // Perkalian
7 y *= x;
8
9 // Hasil y menjadi 50
10 System.out.println("Perkalian : " + y);
11
```

Hasil ketika program dijalankan:



Gambar 3.5 Hasil Perkalian Operator Penugasan

▪ Pembagian

```

1 int x;
2 int y;
3 x = 5;
4 y = 10;
5
6 // Pembagian
7 y /= x;
8
9 // Hasil y menjadi 2
10 System.out.println("Pembagian : " + y);

```



Gambar 3.6 Hasil Pembagian Operator Penugasan

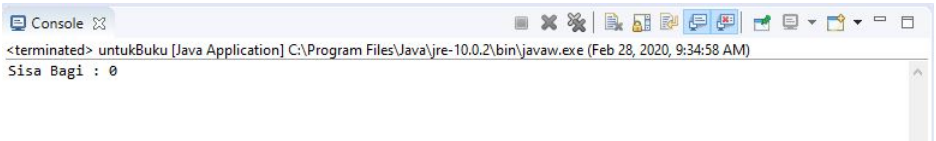
▪ Sisa bagi

```

1 int x;
2 int y;
3 x = 5;
4 y = 10;
5
6 // Pengurangan
7 y %= x;
8
9 // Hasil y menjadi 0
10 System.out.println("Sisa Bagi : " + y);
11

```

Hasil ketika program dijalankan:



Gambar 3.7 Hasil Sisa Bagi Operator Penugasan

3. Operator Logika

Operator Logika merupakan operator yang akan sering kalian temui pada saat membuat sebuah program. Operator logika hanya bernilai **true** atau **false**. Jenis - jenis operator logika:

Nama	Simbol
Logika AND	&&
Logika OR	
Negasi / Kebalikan	!

Tabel 3.4 Operator Logika

Contoh penggunaan operator:

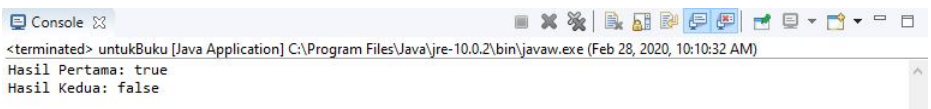
▪ Logika AND

```

1 public static void main(String[] args) {
2     int a = 5;
3     int b = 10;
4     //Hasil Perbandingan a dan b benar keduanya maka Hasil =
      true
5     boolean hasil_satu = (a > 1) && (b > 5);
6     System.out.println("Hasil Pertama: " + hasil_satu);
7
8     //Hasil Perbandingan a dan b hanya benar salah satunya
      saja maka Hasil = false
9     boolean hasil_dua = (a > 1) && (b < 5);
10    System.out.println("Hasil Kedua: " + hasil_dua);
11 }
12

```

Hasil dari program tersebut adalah:



Gambar 3.8 Hasil Operator Logika AND

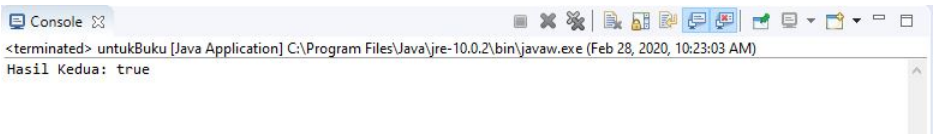
▪ Logika OR

```

1 public static void main(String [] args) {
2     int a = 5;
3     int b = 10;
4
5     //Hasil Perbandingan a dan b hanya benar salah satunya ,
6     //Tertapi jika menggunakan logika OR meskipun hanya benar
7     //salah satu maka akan menghasilkan nilai true
8     boolean hasil_satu = (a > 1) || (b < 5);
9     System.out.println("Hasil Kedua: " + hasil_satu);
10 }

```

Hasil dari program tersebut adalah:



Gambar 3.9 Hasil Operator Logika OR

▪ Negasi / Kebalikan

```

1 public static void main(String [] args) {
2     int a = 5;
3     int b = 10;
4
5     boolean hasil_dua = !((a < 15) ^ (b > 6));
6     System.out.println("Hasil Kedua: " + hasil_dua);
7 }
8

```

Hasil dari program tersebut adalah:



Gambar 3.10 Hasil negasi/kebalikan

Pada penggunaan operator logika ini, kita dapat melihat tabel kebenaran untuk mendapatkan hasil dari logika.

Pernyataan 1	Pernyataan 2	AND	OR
true	true	true	true
true	false	false	true
false	true	false	true
false	false	false	false

Tabel 3.5 Tabel Kebenaran

4. Operator Perbandingan

Operator perbandingan fungsinya adalah membandingkan. Operator ini akan menghasilkan nilai berupa boolean. Boolean berarti hasil ini hanya berupa **true** atau **false**. Jenis - jenis operator logika:

Nama	Simbol
Lebih Besar	>
Lebih Kecil	<
Sama Dengan	==
Tidak Sama Dengan	!=
Lebih Besar Sama Dengan	>=
Lebih Kecil Sama Dengan	<=

Tabel 3.6 Operator Perbandingan

Contoh penggunaan operator:

▪ Lebih Besar

```

1 public static void main(String[] args) {
2     int A = 12;
3     int B = 4;
4     boolean hasil;
5
6     // Apakah nilai A lebih besar dari nilai B?
7     hasil = A > B;
8     System.out.println(hasil);
9 }
10

```

Hasil Eksekusi Program:



Gambar 3.11 Hasil operator lebih besar

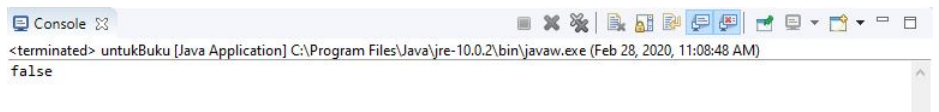
▪ Lebih Kecil

```

1 public static void main(String[] args) {
2     int A = 12;
3     int B = 4;
4     boolean hasil;
5
6     // Apakah nilai A lebih kecil dari nilai B?
7     hasil = A < B;
8     System.out.println(hasil);
9 }
10

```

Hasil Eksekusi Program:



Gambar 3.12 Hasil operator lebih kecil

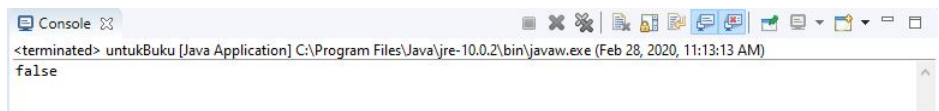
▪ Sama Dengan

```

1 public static void main(String[] args) {
2     int A = 12;
3     int B = 4;
4     boolean hasil;
5
6     // Apakah nilai A sama dengan nilai B?
7     hasil = A == B;
8     System.out.println(hasil);
9 }
10

```

Hasil Eksekusi Program:



Gambar 3.13 Hasil operator sama dengan

▪ Tidak Sama Dengan

```
1 public static void main(String [] args) {  
2     int A = 12;  
3     int B = 4;  
4     boolean hasil;  
5  
6     // Apakah nilai A tidak sama dengan nilai B?  
7     hasil = A != B;  
8     System.out.println(hasil);  
9 }  
10
```

Hasil Eksekusi Program:



Gambar 3.14 Hasil operator tidak sama dengan

▪ Lebih Besar Sama Dengan

```
1 public static void main(String [] args) {  
2     int A = 12;  
3     int B = 4;  
4     boolean hasil;  
5  
6     // Apakah nilai A lebih besar atau sama dengan nilai B?  
7     hasil = A >= B;  
8     System.out.println(hasil);  
9 }  
10
```

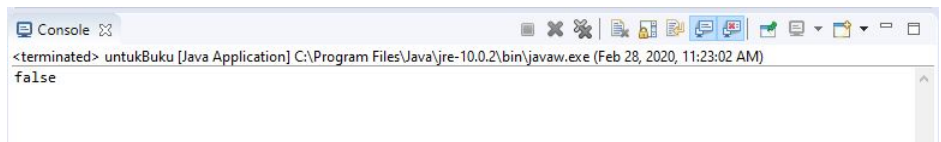
<terminated> untukBuku [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Feb 28, 2020, 11:21:09 AM)
true

Gambar 3.15 Hasil operator lebih besar sama dengan

▪ Lebih Kecil Sama Dengan

```
1 public static void main(String [] args) {  
2     int A = 12;  
3     int B = 4;  
4     boolean hasil;  
5  
6     // Apakah nilai A lebih kecil atau sama dengan nilai B?  
7     hasil = A <= B;  
8     System.out.println(hasil);  
9 }  
10
```

Hasil Eksekusi Program:



Gambar 3.16 Hasil operator lebih kecil sama dengan

5. Operator Bitwise

Operator bitwise fungsinya hampir sama dengan operator logika. Namun perbedaannya adalah operator bitwise digunakan untuk bilangan biner saja (bit). Operator bitwise terdiri dari:

Nama	Simbol
AND	&
OR	
XOR	^
Negasi / Kebalikan	~
Left Shift	<<
Right Shift	>>
Left Shift (Unsigned)	<<<
Right Shift (Unsigned)	>>>

Tabel 3.7 Operator Bitwise

Contoh penggunaan aplikasi bitwise:

```

1 public static void main(String[] args) {
2     int a = 60;    /* 60 = 0011 1100 */
3     int b = 13;    /* 13 = 0000 1101 */
4     int c = 0;
5
6     c = a & b;      /* 12 = 0000 1100 */
7     System.out.println("a & b = " + c);
8
9     c = a | b;      /* 61 = 0011 1101 */
10    System.out.println("a | b = " + c);
11
12    c = a ^ b;      /* 49 = 0011 0001 */
13    System.out.println("a ^ b = " + c);
14
15    c = ~a;         /* -61 = 1100 0011 */
16    System.out.println("~a = " + c);
17
18    c = a << 2;      /* 240 = 1111 0000 */
19    System.out.println("a << 2 = " + c);
20
21    c = a >> 2;      /* 215 = 1111 */

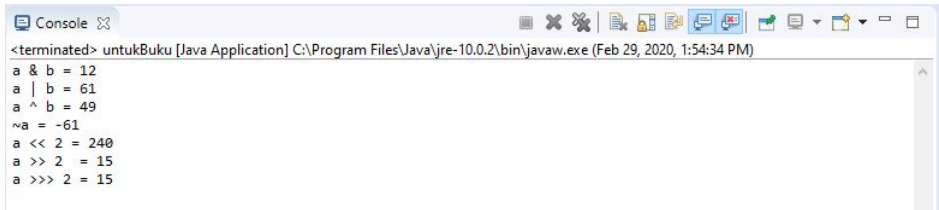
```

```

22     System.out.println("a >> 2 = " + c);
23
24     c = a >>> 2;      /* 215 = 0000 1111 */
25     System.out.println("a >>> 2 = " + c);
26 }
27

```

Hasil yang dikeluarkan adalah:



```

<terminated> untukBuku [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Feb 29, 2020, 1:54:34 PM)
a & b = 12
a | b = 61
a ^ b = 49
~a = -61
a << 2 = 240
a >> 2 = 15
a >>> 2 = 15

```

Gambar 3.17 Hasil Operator Bitwise

3.7 Percabangan pada Java

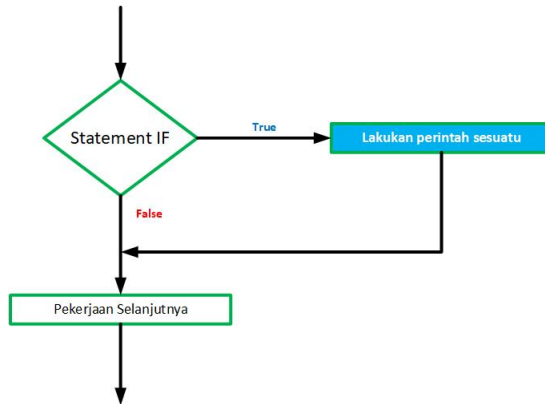
Percabangan merupakan pemilihan keputusan untuk mengeksekusi program berdasarkan kondisi yang ditetapkan. [4]

Percabangan biasa disebut juga dengan *Control Flow* yang merupakan perintah pada program agar program dapat mengambil keputusan. Percabangan akan menggunakan nilai **true** atau **false** dari beberapa kondisi yang terjadi.

Pada dasarnya dalam pemrograman terdapat 3 macam percabangan:

1. Percabangan IF
2. Percabangan IF/ELSE
3. Percabangan IF/ELSE/IF
4. Percabangan CASE

3.7.1 Percabangan IF



Gambar 3.18 Alur logika percabangan IF

Percabangan IF hanya punya satu pilihan dan hanya akan dikerjakan kalau *statementnya* bernilai benar. Jika *statement* tidak bernilai benar, maka percabangan ini tidak bisa dijalankan. Bentuk dari IF adalah:

```

1 if ( statement ) {
2     // perintah jika statement bernilai benar
3     // perintah 1
4     // perintah 2
5     // dan seterusnya
6 }
  
```

Statement hanya akan menghasilkan nilai **True** atau **False**. Untuk pengisian statement dapat digabungkan dengan operator logika, perbandingan, dan berbagai operator yang dapat digunakan dalam program.

Contoh dalam kegiatan sehari - hari adalah ketika kita belanja. Misalkan kita berbelanja diatas nominal 200.000 maka akan mendapat potongan harga.

Jika contoh tersebut dibuat menjadi program maka akan menjadi:

```

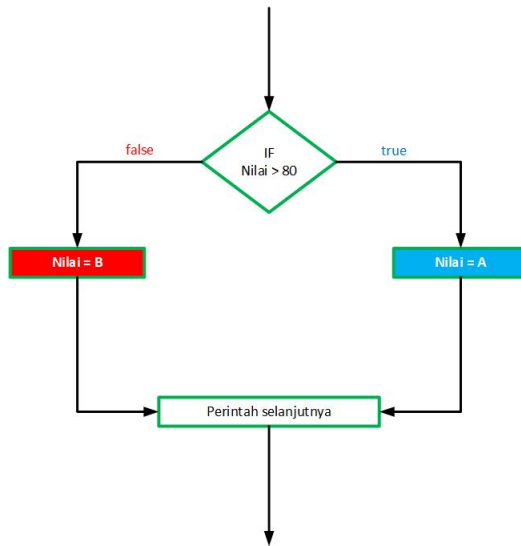
1 public static void main(String[] args) {
2
3     // membuat variabel belanja dan scanner
4     int belanja = 0;
5     Scanner scan = new Scanner(System.in);
6
7     // mengambil input
8     System.out.print("Total Belanjaan: Rp ");
9     belanja = scan.nextInt();
10
11    // cek apakah dia belanja di atas 100000
12    if ( belanja > 100000 ) {
13        System.out.println("Selamat, anda mendapatkan hadiah!");
14    }
15 }
  
```

```

14     }
15
16     System.out.println("Terima kasih...");
17 }

```

3.7.2 Percabangan IF/ELSE



Gambar 3.19 Alur logika percabangan if else

Percabangan IF/ELSE pada dasarnya hampir sama dengan percabangan IF diatas. Namun IF/ELSE memiliki perintah alternatif jika statement yang dijalankan bernilai false.

Struktur dari IF/ELSE adalah:

```

1  if (statement) {
2      // perintah jika statement bernilai benar
3      // perintah 1
4      // perintah 2
5      // dan seterusnya
6  } else {
7      // perintah jika statement bernilai salah
8      // perintah 1
9      // perintah 2
10     // dan seterusnya
11 }

```

Contoh dalam kegiatan sehari - hari adalah ketika dosen akan menentukan nilai hasil ujian akhir semester mahasiswa. Jika nilai mahasiswa > 80 maka nilai = A, dan jika nilai dibawah 80 maka nilai = B

Bentuk Program dari contoh tersebut:

```

1 public static void main(String [] args) {
2
3     // membuat variabel Nilai dan scanner
4     int nilai = 0;
5     Scanner scan = new Scanner(System.in);
6
7     // mengambil input
8     System.out.print("Nilai: ");
9     nilai = scan.nextInt();
10
11    // cek apakah nilai UAS diatas 80?
12    if (nilai > 80) {
13        //jika hasil benar maka:
14        System.out.println("Nilai: A");
15    } else {
16        //(Alternatif) Jika hasil salah maka:
17        System.out.println("Nilai: B");
18    }
19
20 }

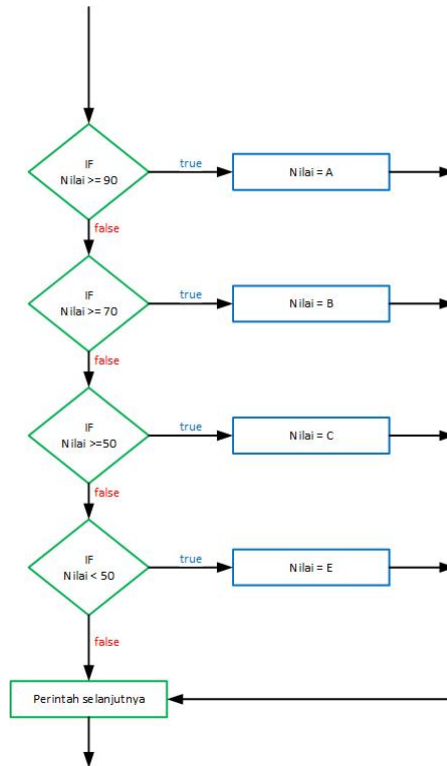
```

Hasil dari program yang dijalankan adalah:



Gambar 3.20 Hasil program if else

3.7.3 Percabangan IF/ELSE/IF



Gambar 3.21 Flowmap logika IF/ELSE/IF

Jika kita lihat pada percabangan IF/ELSE hanya memiliki dua percabangan saja yaitu jika kondisi true dan false. Namun pada percabangan IF/ELSE/IF, percabangan bisa memiliki banyak pilihan ketika kondisi yang dijalankan memiliki nilai false.

Struktur dari IF/ELSE/IF:

```

1  if (statement 0) {
2      // maka kerjakan perintah ini
3      // perintah 1
4      // perintah 2
5      // ....
6  } else if (statement 1) {
7      // kerjakan perintah ini
8      // perintah 1
9      // perintah 2
10     // ....
11 } else if (statement 2) {
12     // kerjakan perintah ini
13     // perintah 1 juga
14     // perintah 2 juga

```

```

15 // ....
16 } else {
17 // kerjakan perintah ini
18 // hanya jika
19 // kondisi statement diatas tidak ada yang benar
20 // ....
21 }

```

Contoh dalam kegiatan sehari-hari adalah ketika dosen akan menentukan nilai grade dari mahasiswa. Jika Nilai ≥ 90 maka Grade = A, jika Nilai ≥ 70 maka Grade = B, jika Nilai ≥ 50 Grade C dan nilai < 50 maka Grade E.

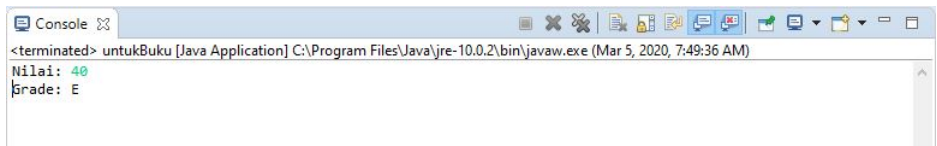
Bentuk contoh tersebut jika diubah menjadi bahasa program:

```

1 public static void main(String[] args) {
2
3     // membuat variabel Nilai dan scanner
4     int nilai = 0;
5     String grade;
6
7     Scanner scan = new Scanner(System.in);
8
9     // mengambil input
10    System.out.print("Nilai: ");
11    nilai = scan.nextInt();
12
13    // IF/ELSE/IF untuk menentukan grade nilai mahasiswa
14    if ( nilai >= 90 ) {
15        grade = "A";
16    } else if ( nilai >= 70 ) {
17        grade = "B";
18    } else if ( nilai >= 50 ) {
19        grade = "C";
20    } else {
21        grade = "E";
22    }
23
24    // Mencetak Grade dan nilai yang sudah diinputkan
25    System.out.println("Grade: " + grade);
26
27 }

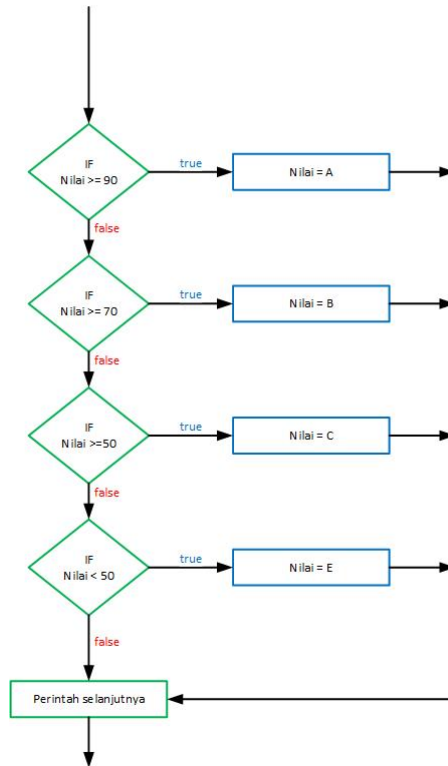
```

Hasil dari program yang dijalankan adalah:



Gambar 3.22 Hasil program IF/ELSE/IF

3.7.4 Percabangan SWITCH/CASE



Gambar 3.23 Flowmap logika SWITCH/CASE

Percabangan SWITCH/CASE secara fungsi hampir sama dengan percabangan IF/ELSE/IF namun pada percabangan ini perlu diingat bahwa kata kunci yang digunakan adalah **switch** dan **case**.

Bentuk program dari SWITCH/CASE adalah:

```

1 switch (variabel){
2     case 1:
3         // Kerjakan perintah ini
4         // perintah ini juga
5         // perintah ini
6         // ...
7         break;
8     case 2:
9         // Kerjakan perintah ini
10        // perintah ini juga
11        // perintah ini
12        // ...
13        break;
14    case 3:

```

```

15         // Kerjakan perintah ini
16         // perintah ini juga
17         // perintah ini
18         // ...
19         break;
20     default:
21         // Kerjakan perintah ini
22         // perintah ini juga
23         // perintah ini
24         // ...
25     break;
26 }

```

Case 1 disini berisi Variabel yang akan dibandingkan sehingga memiliki hasil akhir true tau false.

Contoh dalam kegiatan sehari - hari adalah ketika anda memiliki banyak pilihan seperti pada mesin ATM. Pada mesin ini disediakan beberapa pilihan menu seperti transfer, tarik tunai, cek saldo, dan lain-lain.

Bentuk contoh tersebut jika diubah menjadi bahasa program:

```

1 // membuat variabel dan Scanner
2 String pilihan;
3 Scanner scan = new Scanner(System.in);
4
5 // mengambil input
6 System.out.println("Selamat Datang di atm SNI");
7 System.out.println("1. Cek Saldo");
8 System.out.println("2. Tarik Tunai");
9 System.out.println("3. Setor");
10 System.out.println("Pilih Menu: ");
11 pilihan = scan.nextLine();
12
13 switch(pilihan){
14     case "1":
15         System.out.println("Saldo anda Rp.0,-");
16         break;
17     case "2":
18         System.out.println("Anda Tidak Punya Saldo");
19         break;
20     case "3":
21         System.out.println("Setor trosss");
22         break;
23     default:
24         System.out.println("Masukan pilihan dengan benar!");
25 }

```

Hasil dari program yang dijalankan:

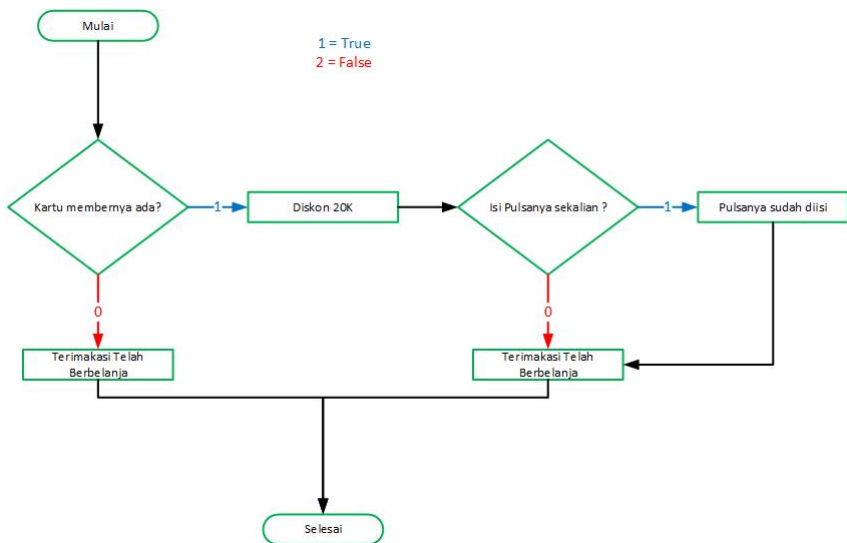
```

Console
<terminated> untukBuku [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Mar 5, 2020, 8:22:58 AM)
Selamat Datang di atm SNI
1. Cek Saldo
2. Tarik Tunai
3. Setor
Pilih Menu:
1
Saldo anda Rp.0,-

```

Gambar 3.24 Hasil program SWITCH/CASE

3.7.5 Percabangan dalam percabangan (*Nested*)



Gambar 3.25 Flowmap logika percabangan dalam percabangan

Pada dasarnya percabangan dalam percabangan ini hampir sama dengan IF/ELSE karena menggunakan perintah **if** dan **else**. Namun ada **if/else** dalam **if/else** sehingga disebut "Percabangan dalam percabangan (*Nested*)"

Struktur percabangan dalam percabangan:

```

1  if (statement) {
2      if (statement) {
3          // Eksekusi perintah ini
4          // Perintah ini
5          // perintah itu
6          // ....
7      } else if (statement2) {
8          // Eksekusi perintah ini
9          // Perintah ini
10         // perintah itu

```



```

11         // ....
12     } else {
13         // Eksekusi perintah ini
14         // Perintah ini
15         // perintah itu
16         // ....
17     }
18
19     } else {
20         if (statement) {
21             // Eksekusi perintah ini
22             // Perintah ini
23             // perintah itu
24             // ....
25         } else {
26             // Eksekusi perintah ini
27             // Perintah ini
28             // perintah itu
29             // ....
30         }
31     }

```

Contoh dalam kegiatan sehari - hari adalah ketika membeli sesuatu di minimarket, kemudian kasir bertanya kartu member. Jika memiliki kartu member maka diskon 20.000 dan kasir akan bertanya lagi apakah akan isi pulsa sekalian? jika ya maka pulsa diisi, jika tidak maka selesai. Sedangkan jika tidak punya kartu member, Maka proses langsung selesai.

Bentuk Contoh tersebut jika diubah menjadi bahasa program:

```

1 // deklarasi variabel dan Scanner
2 String kartu , pulsa;
3 Scanner scan = new Scanner(System.in);
4
5 // mengambil input
6 System.out.print("Apakah ada kartu membernya?: ");
7 kartu = scan.nextLine();
8
9 // proses
10 if (kartu.equalsIgnoreCase("ya")) {
11     System.out.println("Anda mendapat diskon 20.000");
12     System.out.print("Mau isi pulsa sekalian?: ");
13     pulsa = scan.nextLine();
14     if (pulsa.equalsIgnoreCase("ya")) {
15         System.out.println("Pulsa Terisi");
16         System.out.println("Terimakasih telah berbelanja");
17     } else if (pulsa.equalsIgnoreCase("tidak")) {
18         System.out.println("Terimakasih telah berbelanja");
19     } else {
20         System.out.println("Terimakasih telah berbelanja");
21     }
22 }
23 } else {
24     System.out.println("Terimakasih telah berbelanja");
25 }

```

Hasil dari program yang dijalankan:

A screenshot of a Java console window titled 'Console'. The window shows the execution of a Java application. The output text is as follows:

```
<terminated> untukBuku [Java Application] C:\Program Files\Java\jre-10.0.2\bin\javaw.exe (Mar 5, 2020, 9:56:23 AM)
Apakah ada kartu membernya?: ya
Anda mendapat diskon 20.000
Mau isi pulsa sekalian?: ya
Pulsa Terisi
Terimakasih telah berbelanja
```

Gambar 3.26 Hasil program percabangan dalam percabangan

BAB 4

JAVA DEVELOPMENT KIT

4.1 *JDK Java Development Kit*

Java Development Kit (JDK) adalah salah satu dari tiga package inti yang digunakan dalam pemrograman Java, bersama dengan JVM (Java Virtual Machine) dan JRE (Java Runtime Environment)..[5]

DAFTAR PUSTAKA

1. MateriDosen, “Pengertian dan fungsi kode ascii (lengkap),” <http://www.materidosen.com/2016/10/pengertian-dan-fungsi-kode-ascii-lengkap.html>, Oktober 2016, diakses pada 27 Februari 2020.
2. A. Andianto, *Pemrograman Dasar Menggunakan Java dan NetBeans IDE*. Pamekasan: Duta Media, 2017.
3. K. F. N. . L. H. B., *Foundation of Behavioral Research. Fourth Edition*. Harcourt College Publishers, 2000.
4. M. H. Alexander, *Membuat Aplikasi Database dengan Delphi 8.0*. Jakarta: Elex Media Komputindo, 2005.
5. M. Tyson, “What is the jdk? introduction to the java development kit,” <https://www.javaworld.com/article/3296360/what-is-the-jdk-introduction-to-the-java-development-kit.html>, Januari 2020, diakses pada 5 Maret 2020.

Index

disruptif, xxix
modern, xxix