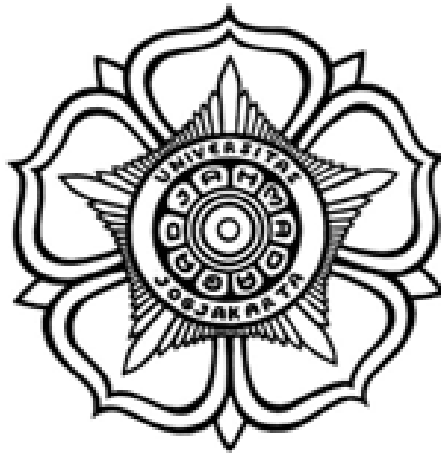


MODUL PRAKTIKUM REKAYASA PERANGKAT LUNAK



**PROGRAM STUDI D3 KOMPUTER DAN SISTEM INFORMASI
SEKOLAH VOKASI
UNIVERSITAS GADJAH MADA
YOGYAKARTA
2013**

BAB I

PENDAHULUAN

Tujuan Pembelajaran

1. Mahasiswa mengenal tools yang akan digunakan pada praktikum RPL.
2. Mahasiswa mengenal macam-macam software yang digunakan untuk perancangan sistem.

Keluaran Pembelajaran (Skill)

1. Mahasiswa mengetahui tools yang digunakan untuk perancangan perangkat lunak.

1.1. Pengenalan Tools

Software atau tools yang akan digunakan pada praktikum RPL adalah Microsoft Visio dan StarUML.

1.2. Microsoft Visio

Microsoft Visio (atau sering disebut Visio) adalah sebuah program aplikasi komputer yang sering digunakan untuk membuat diagram, diagram alir (flowchart), brainstorm, dan skema jaringan yang dirilis oleh Microsoft Corporation. Aplikasi ini menggunakan grafik vektor untuk membuat diagram-diagramnya.



Visio, merupakan software yang mendukung untuk mendeskripsikan data dan fungsi/proses yang terlibat dalam proses rekayasa suatu perangkat lunak. Adapun data dan fungsi yang dimodelkan dalam proses dalam rekayasa perangkat lunak adalah

1. Pemodelan Data
 - a. ERD (Diagram Keterhubungan antar Objek Data)

Entity Relationship Diagram merupakan diagram yang dapat digunakan untuk melakukan aktifitas pemodelan data dan menggambarkan hubungan antara objek data. Dalam ERD terdapat tiga komponen yang digunakan yaitu entitas, atribut dan relasi .

Contoh :

- Entitas (Dosen)
- Atribut : NIP, Nama Dosen, Alamat, No_Telpon, Jabatan

b. Kamus Data

Merupakan catatan yang digunakan untuk menyimpan deskripsi atau atribut dari semua objek data yang didefinisikan.

Contoh : Dosen : { NIP, Nama Dosen, Alamat, No_Telpon, Jabatan }

2. DFD

Mendeskripsikan seluruh fungsi yang terlibat dalam Perangkat Lunak. DFD atau diagram Aliran Data, merupakan gambaran bagaimana data ditransformasikan pada sebuah sistem.



1.3. Star UML

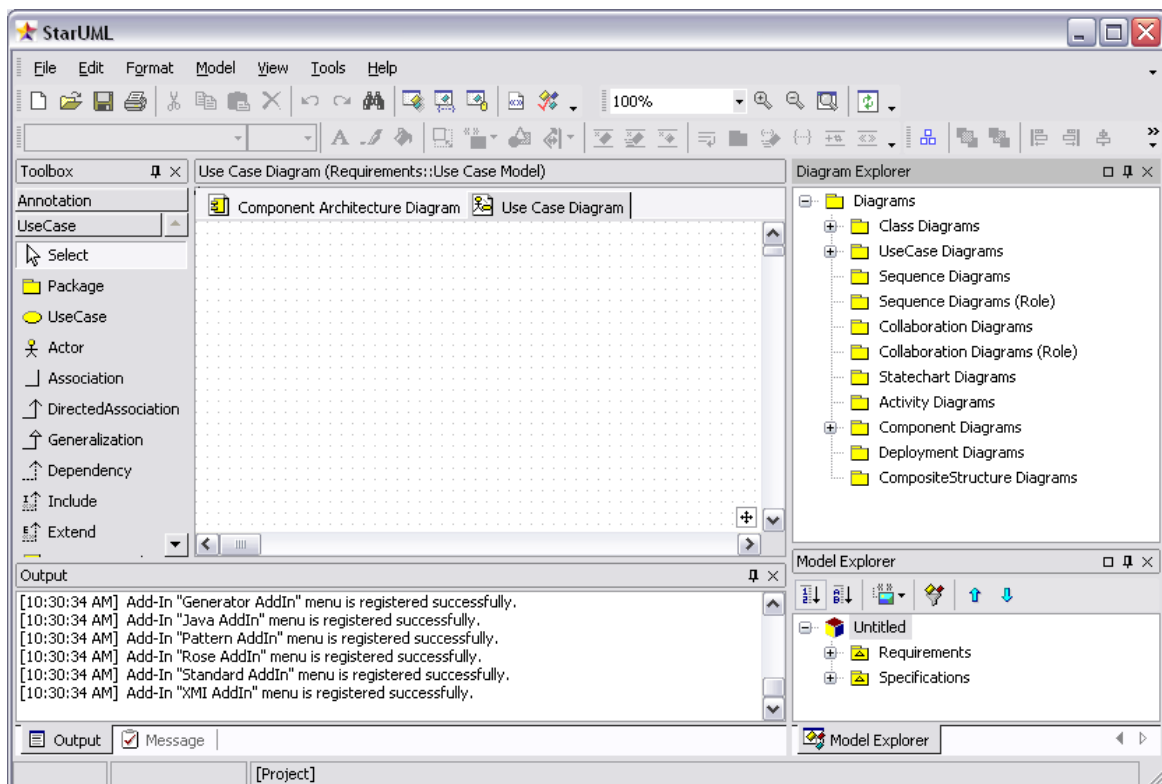
StarUML, adalah salah satu tool pendesainan yang mendukung **Unified Modelling Language (UML)** yang bersifat open source. Hampir semua kegiatan dalam modul ini dilakukan

menggunakan StarUML yang fungsinya bisa dibandingkan dengan Rational Rose, Together, atau UML tool lainnya.



Adapun software atau tools lainnya baik itu tool komersial maupun opensource antara lain :

- Rational Rose (www.rational.com)
- Together (www.togethersoftware.com)
- Object Domain (www.objectdomain.com)
- Jvision (www.object-insight.com)
- Objectteering (www.objectteering.com)
- MagicDraw (www.nomagic.com/magicdrawuml)
- Visual Object Modeller (www.visualobject.com)



BAB II

PENGANTAR RPL

Tujuan Pembelajaran

1. Mahasiswa mampu menjelaskan pengertian perangkat lunak
2. Mahasiswa mampu menjelaskan analisis sistem dalam rekayasa perangkat lunak
3. Mahasiswa mampu menjelaskan objek data, atribut dan hubungan antar objek

Keluaran Pembelajaran (Skill)

1. Agar mahasiswa dapat menjelaskan pengertian perangkat lunak.
2. Agar mahasiswa dapat menyebutkan dan menjelaskan contoh jenis dan aplikasi perangkat lunak
3. Agar praktikan bisa membedakan antara objek data, atribut dan hubungannya.

TEORI

A. Pengertian Perangkat Lunak

Perangkat lunak kini sudah menjadi kekuatan yang menentukan. Perangkat lunak menjadi mesin yang mengendalikan pengambilan keputusan di dalam dunia bisnis, berfungsi sebagai dasar dari semua bentuk pelayanan serta penelitian keilmuan modern. Saat ini perangkat lunak memiliki dua peran. Di satu sisi berfungsi sebagai sebuah produk, dan di sisi lain sebagai kendaraan yang mengantarkan sebuah produk.

Sebenarnya, apa yang dimaksud dengan **Perangkat Lunak**? Perangkat Lunak (*Software*) adalah (1) *Perintah (program komputer) yang bila dieksekusi memberikan fungsi dan unjuk kerja seperti yang diinginkan.* (2) *Struktur data yang memungkinkan program memanipulasi informasi secara proporsional.* (3) *Dokumen yang menggambarkan operasi dan kegunaan program.*

Sedangkan yang dimaksud dengan **Rekayasa Perangkat Lunak** adalah *Suatu disiplin ilmu yang membahas semua aspek produksi perangkat lunak, mulai dari tahap awal requirement*

capturing (analisa kebutuhan pengguna), specification (menentukan spesifikasi dari kebutuhan pengguna), desain, coding, testing sampai pemeliharaan sistem setelah digunakan.

Jadi yang perlu digaris bawahi, Rekayasa Perangkat Lunak merupakan serangkaian proses yang amat panjang untuk membuat atau menciptakan suatu perangkat lunak, bukan merupakan cabang ilmu *Computer Science* yang mempelajari tentang *technical coding*.

B. OBJEK DATA, ATRIBUT DAN RELASI

Model data terdiri dari tiga informasi yang saling tergantung : objek data, atribut yang menggambarkan objek data tersebut, dan hubungan yang menghubungkan objek data yang satu dengan yang lain.

Objek data (entity) adalah representasi dari hampir semua informasi gabungan yang harus dipahami oleh perangkat lunak. Dengan informasi gabungan kita mengartikan sesuatu yang memiliki sejumlah sifat atau atribut yang berbeda.

Atribut adalah karakteristik dari entity atau relationship yang menyediakan detail tentang entity atau relationship tersebut sehingga dapat dibedakan.

Relasi adalah hubungan yang terjadi antara satu atau lebih objek data (entity).

SKENARIO

Tentukan objek, atribut dan relasi pada gambar (2.1)



Gambar 2.1 Pemeriksaan

Step 1 Tentukan objek data dari gambar (2.1)

Objek = Dokter dan Pasien

Step 2 Tentukan atribut yang dimiliki.

Dokter = NIP, nama, alamat, spesialisasi

Pasien = NoRM, nama, alamat, jenis_kelamin, tgl_lahir, gol_darah

Step 3 Tentukan relasi dari kedua objek tersebut

Relasi = Dokter memeriksa Pasien (memeriksa merupakan nama relasi)

LATIHAN

1. Sebutkan contoh-contoh perangkat lunak beserta kegunaanya.
2. Tuliskan objek, atribut dan relasi dari gambar 2.2



Gambar 2.2 Mobil berjalan

3. Seorang pengendara sepeda motor melakukan pelanggaran di jalan raya karena melewati garis marka. Polisi menghentikan laju pengendara dan mengenakan sanksi sesuai dengan undang-undang. Tentukan objek, atribut dan relasi dari deskripsi tersebut.

BAB III

ANALISIS SISTEM

Tujuan Pembelajaran

1. Mahasiswa mampu menjelaskan pengertian analisis sistem
2. Mahasiswa mampu menjelaskan tahapan analisis sistem
3. Mahasiswa mampu menjelaskan analisis kebutuhan sistem dalam rekayasa perangkat lunak

Keluaran Pembelajaran (Skill)

1. Agar mahasiswa dapat menjelaskan tahapan analisa sistem dalam rekayasa perangkat lunak dengan benar
2. Agar mahasiswa dapat menjelaskan tujuan analisa sistem dalam rekayasa perangkat lunak dengan benar
3. Agar mahasiswa dapat menjelaskan analisis kebutuhan sistem dalam rekayasa perangkat lunak dengan benar
4. Agar mahasiswa dapat menyebutkan kebutuhan fungsional dan non fungsional dalam rekayasa perangkat lunak dengan benar

Teori

A. ANALISIS SISTEM.

Analisis sistem adalah kegiatan untuk mendefinisikan kebutuhan terkait sistem yang akan dikembangkan. Orang yang bertugas menganalisis sistem disebut sebagai *system analyst*.

a. Tahapan Analisis Sistem

Dalam melakukan analisis sistem tahapan yang dilalui adalah :

1. IDENTIFY → mengidentifikasi masalah
2. UNDERSTAND → memahami kerja sistem yang ada
3. ANALYZE → menganalisis sistem
4. REPORT → membuat laporan hasil

b. Tujuan dari analisis sistem antara lain :

- Menjabarkan kebutuhan pemakai
- Meletakkan dasar-dasar untuk proses perancangan perangkat lunak
- Mendefinisikan semua kebutuhan pemakai sesuai dengan lingkup kedua belah pihak

B. ANALISIS KEBUTUHAN

a. Analisis Kebutuhan Fungsional (*Functional Requirement*)

Merupakan kegiatan dalam mendeskripsikan fungsionalitas atau layanan yang diharapkan akan diberikan oleh sistem. Persyaratan fungsional untuk sistem perangkat lunak bisa dinyatakan dalam sejumlah cara. Persyaratan tersebut bergantung pada jenis perangkat lunak, user yang menggunakan, dan jenis sistem yang akan dikembangkan.

Contoh :

Terdapat sejumlah persyaratan fungsional untuk sistem perpustakaan (Kotonya dan Sommerville, 1998) bagi mahasiswa dan dosen untuk memesan buku dan dokumen dari perpustakaan lain :

1. User dapat mencari semua atau satu set awal database atau memilih subset darinya
2. Sistem akan menyediakan viewer yang sesuai bagi user untuk membaca dokumen pada penyimpanan (store) dokumen.

Pada prinsipnya spesifikasi persyaratan fungsional untuk sebuah sistem harus lengkap dan konsisten.

b. Analisis Kebutuhan Non Fungsional (*Non Functional Requirement*)

Merupakan persyaratan yang tidak langsung berhubungan dengan fungsi spesifik yang disediakan oleh sistem. Persyaratan ini mungkin berhubungan dengan properti sistem yang muncul belakangan seperti kehandalan, waktu tanggap, dan penempatan pada media penyimpanan. Alternatifnya, persyaratan ini dapat mendefinisikan batasan pada sistem seperti kemampuan piranti I/O dan representasi data yang dipakai pada interface sistem.

Kebutuhan non fungsional ini meliputi kebutuhan :

a. **Development Requirement**

Tools yang digunakan (hardware dan software) untuk pengembangan sistem.

Contoh : eclipse, netbeans, starUML dsb.

b. **Deployment Requirement**

Terkait dengan lingkungan dimana sistem akan digunakan

Contoh: sistem harus mampu berjalan dengan spesifikasi RAM 4GB, OS Ubuntu, dsb.

c. **Performance Requirement**

Terkait dengan ukuran kualitas maupun kuantitas khususnya terkait dengan kecepatan, skalabilitas, dan kapasitas.

Contoh: sistem harus mampu diakses oleh 100org dalam waktu bersamaan.

d. **Dokumentation Requirement**

Terkait dengan dokumen apa saja yang akan disertakan pada produk akhir.

Contoh : dokumen teknis (dokumen perencanaan proyek, analisis , desain, pengujian), *user manual*, dan dokumen pelatihan.

e. **Support Requirement**

Kebutuhan yang terkait dengan dukungan yang diberikan setelah sistem informasi digunakan

Contoh: perlu adanya pelatihan bagi calon pengguna.

SKENARIO

Minimarket milik pak joko menjual mulai peralatan rumah tangga , alat tulis, dan barang kelontong untuk kebutuhan sehari-hari. Pak joko ingin membuat sistem yang bisa digunakan untuk transaksi jual beli. Sistem tersebut akan digunakan oleh Pak Joko sebagai direktur dan karyawannya sebagai kasir. Jika anda diminta membangun sistem tersebut, analisislah kebutuhan sistemnya terlebih dahulu.

Step 1 : Identifikasi objek (entitas) lengkap dengan atribut yang terlibat di dalam sistem tersebut

Objek = direktur, barang, kasir

Step 2 : Tuliskan jawaban anda kedalam model kamus data

Kamus data minimarket :

Direktur = {nama, alamat, no_telp}

Barang = {kode_barang, nama_barang, jenis, stok}

Kasir = {kode_kasir, nama, alamat, no_telp}

Step 3 : Tuliskan kebutuhan fungsional sistem

1. Kasir dapat menggunakan sistem untuk transaksi penjualan
2. Sistem dapat digunakan untuk menyimpan data barang baru
3. Direktur dapat melihat laporan data barang dan data penjualan

Step 4 : Tuliskan kebutuhan non fungsional sistem

1. Sistem ini berbasis desktop
2. Sistem dibangun dengan menggunakan bahasa pemrograman J2SE
3. Sistem dapat berjalan pada komputer dengan spesifikasi minimal RAM 1GB

Latihan :

1. Tentukan objek data dan atribut yang bisa ditemukan dalam apotek. Tulisakan hasil analisis kebutuhan fungsional dan nonfungsional untuk sistem informasi apotek.
2. Analisislah kembali dengan mengembangkan ide-ide baru pada sistem informasi akademik (SIA) di kampus anda. Buatlah kebutuhan fungsional dan non fungsionalnya. Tuliskan teknologi baru yang anda gunakan, jika ada.

BAB III

DIAGRAM ALIR DATA I

Tujuan Pembelajaran

1. Mengenalkan DAD (Diagram Alir Data) sebagai tool perancangan sistem
2. Memahami tahapan-tahapan levelisasi DAD
3. Menggunakan Microsoft Visio untuk menggambar DAD

Alat dan bahan

Microsoft Visio

Keluaran Pembelajaran (Skill)

1. Agar mahasiswa dapat menjelaskan pengertian DAD dan kegunaannya
2. Agar mahasiswa dapat menjelaskan komponen-komponen yang terlibat dalam pembuatan DAD.
3. Agar mahasiswa dapat menciptakan DAD dalam perancangan sistem.
4. Agar mahasiswa dapat menggambar DAD menggunakan Microsoft Visio.


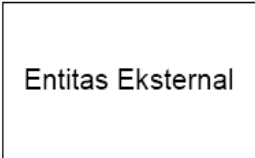
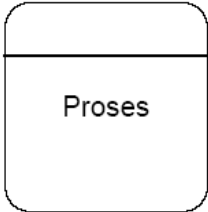
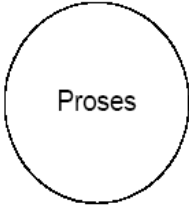
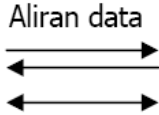
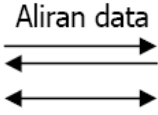
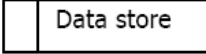
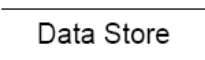
Teori

A. DAD dan Komponen Pembentuk DAD

DAD (Diagram Aliran Data) atau yang juga dikenal dengan sebutan **DFD (Data Flow Diagram)** merupakan alat perancangan sistem yang berorientasi pada alur data dengan konsep dekomposisi dapat digunakan untuk penggambaran analisa maupun rancangan sistem yg mudah dikomunikasikan oleh profesional sistem kepada pemakai maupun pembuat program. Pada umumnya, DAD digunakan untuk merancang sistem yang menggunakan data store dalam mengelola informasi dalam sistem.

Komponen DAD, menurut Yourdon dan DeMarco adalah sebagai berikut :

Tabel 3.1 Komponen DAD

Gane/Sarson	Yourdon/De Marco	Keterangan
		Entitas eksternal, dapat berupa orang/unit terkait yang berinteraksi dengan sistem tetapi diluar sistem
		Orang, unit yang mempergunakan atau melakukan transformasi data. Komponen fisik tidak diidentifikasi.
		Aliran data dengan arah khusus dari sumber ke tujuan
		Penyimpanan data atau tempat data direfer oleh proses.

Keterangan :

- a. **Entitas Luar** : kesatuan diluar sistem yang akan memberikan input atau menerima output dari sistem, dapat berupa orang atau,organisasi,sumber informasi lain atau penerima akhir dari suatu laporan.
- b. **Proses** adalah transformasi input menjadi output, merupakan kegiatan atau pekerjaan yang dilakukan oleh orang atau mesin komputer, dimana aliran data masuk ditransformasikan ke aliran data keluar. Penamaannya sesuai dengan proses yang sedang dilakukan.

Ada beberapa hal yang perlu diperhatikan tentang proses :

1. Proses harus memiliki input dan output.

2. Proses dapat dihubungkan dengan komponen *entitas luar*, *data store* atau *proses* melalui alur data.
3. Sistem/bagian/divisi/departemen yang sedang dianalisis oleh profesional sistem digambarkan dengan komponen proses.
4. Penomoran proses dapat dilihat pada tabel 3.1

Tabel 3.1 Penomoran proses

Nama Level	Nama Diagram	Nomor Proses
0	Konteks	0
1	Diagram level 1	1.0, 2.0, 3.0
2	Diagram rinci 1.0	1.1, 1.2, 1.3
2	Diagram rinci 2.0	2.1, 2.2, 2.3
2	Diagram rinci 3.0	3.1, 3.2, 3.3
3	Diagram rinci 1.1	1.1.1, 1.1.2, 1.1.3
3	Diagram rinci 1.2	1.2.1, 1.2.2, 1.2.3
3	Diagram rinci 1.3	1.3.1, 1.3.2, 1.3.2
dst		

- c. **Aliran data/Arus data** digunakan untuk menjelaskan perpindahan data atau paket data dari satu bagian ke bagian lain.

Aliran data dapat berbentuk sebagai berikut :

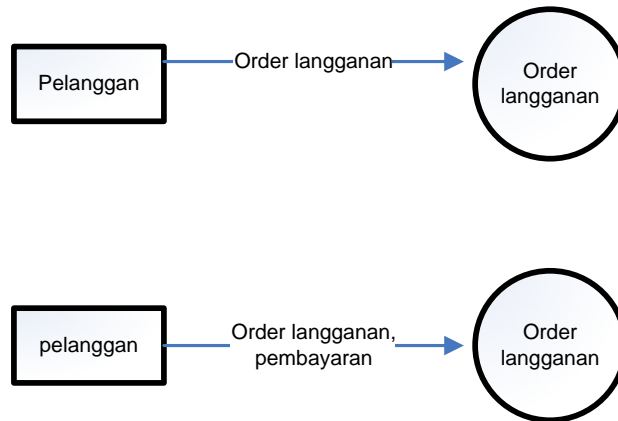
- Formulir atau dokumen yang digunakan perusahaan
- Laporan tercetak yang dihasilkan sistem
- Output dilayar komputer
- Masukan untuk komputer
- Komunikasi ucapan
- Surat atau memo
- Data yang dibaca atau direkam di file
- Suatu isian yang dicatat pada buku agenda
- Transmisi data dari suatu komputer ke komputer lain

Catatan : aliran data tidak dalam bentuk kalimat.

Konsep Arus Data :

1. Packet of Data (Paket Data)

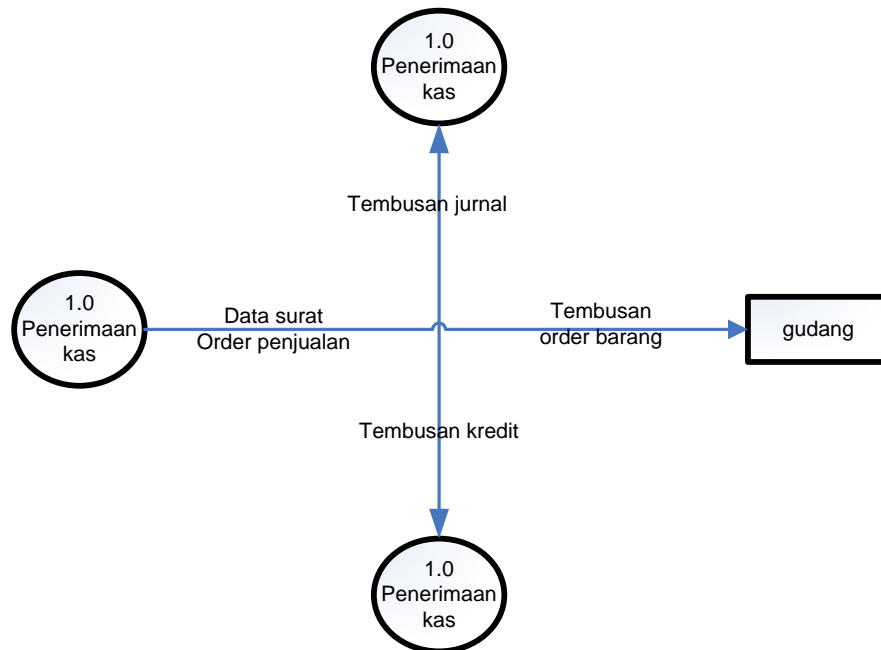
Bila dua data mengalir dari suatu sumber yang sama menuju ke tujuan yang sama, maka harus dianggap sebagai suatu arus data yang tunggal.



Gambar 3.1 Paket Data

2. Diverging Data Flow (Arus Data Menyebar)

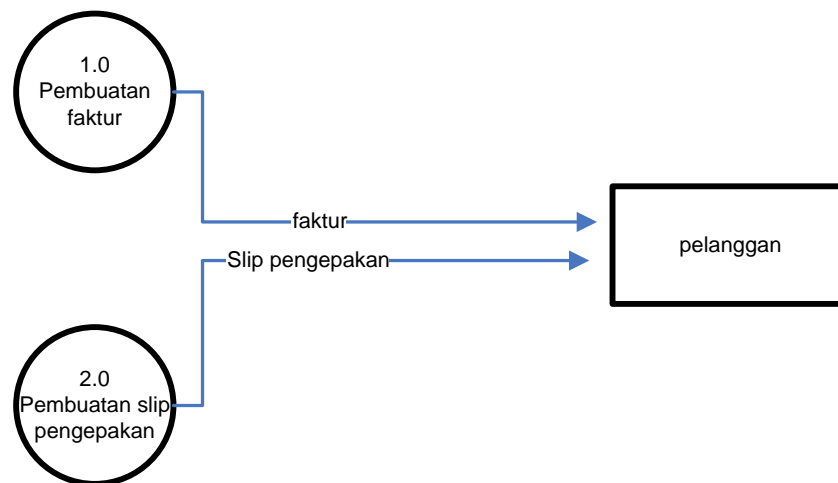
Arus data yang data yang menyebar menunjukkan sejumlah tembusan dari arus data yang sama dari sumber sama ke tujuan berbeda.



Gambar 3.2 Arus data menyebar

3. Convergen Data Flow (Arus Data Mengumpul)

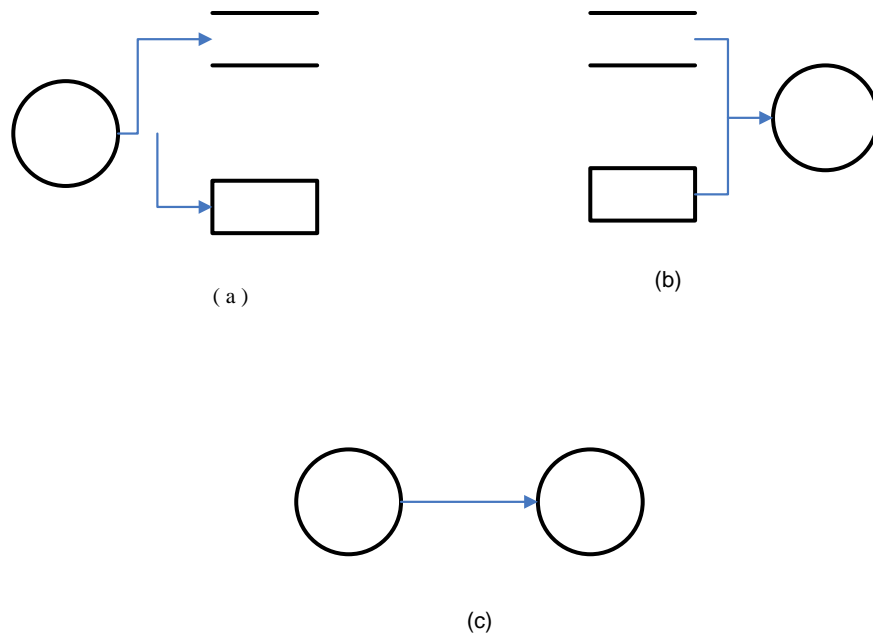
Arus data yang mengumpul yaitu Arus data yang berbeda dari sumber yang berbeda mengumpul ke tujuan yang sama.



Gambar 3.3 Arus data mengumpul

4. Sumber dan Tujuan

Arus data harus dihubungkan pada proses, baik dari maupun yang menuju proses.



Gambar 3.4 Sumber dan Tujuan dalam DAD

Keterangan :

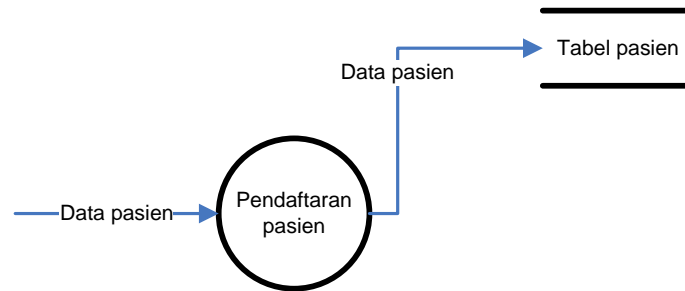
- a. Dari proses ke bukan proses
- b. Dari bukan proses ke proses
- c. Dari proses ke proses

5. Data Storage

Merupakan komponen untuk membuat model sekumpulan data, dapat berupa suatu file atau suatu sistem database dari suatu komputer, suatu arsip/dokumen, suatu agenda/buku.

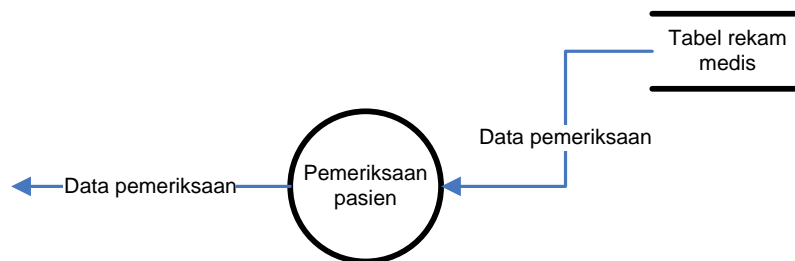
Yang perlu diperhatikan tentang data store :

1. Alur data dari proses menuju data store, hal ini berarti data store berfungsi sebagai tujuan/tempat penyimpanan dari suatu proses (proses write).



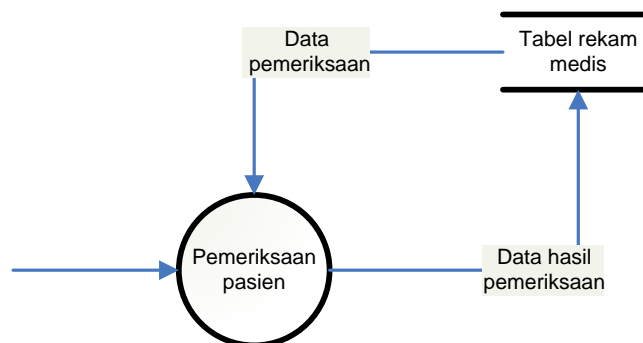
Gambar 3.5 Proses Write

2. Alur data dari data store ke proses, hal ini berarti data store berfungsi sebagai sumber/ proses yang memerlukan data (proses read).



Gambar 3.6 Proses Read

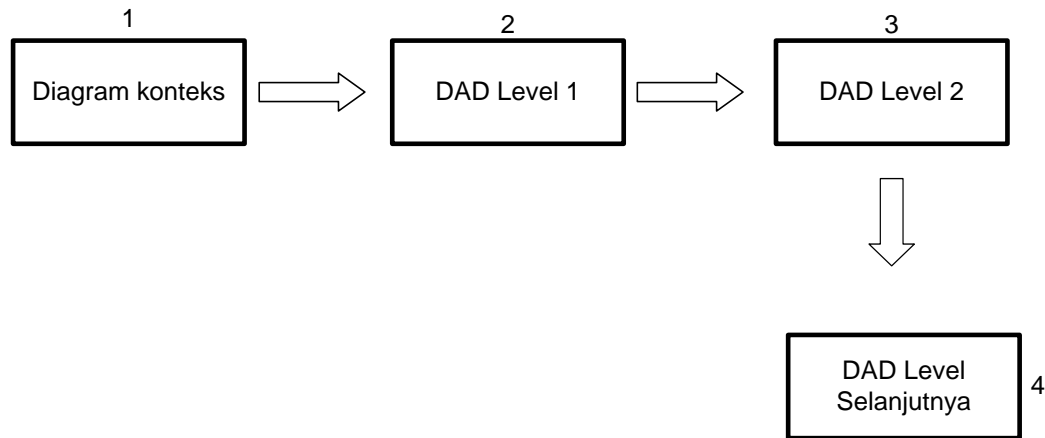
3. Alur data dari proses menuju data store dan sebaliknya berarti berfungsi sebagai sumber dan tujuan



Gambar 3.7 Proses Write dan Read

B. DIAGRAM-DIAGRAM DI DALAM DAD

Secara umum konsep untuk menggambarkan DAD sebuah sistem dimulai dari menggambar diagram konteks, DAD Level 1, DAD Level 2 dan level selanjutnya (sesuai kebutuhan)



Gambar 3.8 Alur Penggambaran DAD

I. Diagram Konteks

Disebut juga diagram tingkat atas, merupakan diagram sistem yang menggambarkan aliran-aliran data yang masuk dan keluar dari sistem dan yang masuk dan keluar dari entitas luar.

Hal yang harus diperhatikan :

- Memberikan gambaran tentang seluruh sistem
- Terminal yang memberikan masukan ke sistem disebut *source*
- Terminal yang menerima keluaran disebut *sink/destination*
- Hanya ada satu proses
- Tidak boleh ada data store

Merupakan diagram yang merepresentasikan seluruh elemen sistem sebagai sebuah bubble tunggal dengan data input dan output yang ditunjukkan oleh anak panah yang masuk dan keluar secara berurutan, meliputi :

- a. Apa saja yang dibutuhkan ?

Mencakup sistem/proses yang dipandang secara keseluruhan

- b. Siapa saja pihak yang berhubungan langsung dengan sistem (yang memberi dan menerima)

Mencakup eksternal entiti yang terkait dengan sistem

- c. Data apa yang diberikan ke sistem (input) dan yang dihasilkan (output)

Mencakup arus data

SKENARIO 1

Diberikan aplikasi kalkulator, seperti terlihat pada gambar 3.9. Gambarlah diagram konteks dari kalkulator.



Gambar 3.9 Kalkulator

Step 1 : Menentukan sistem atau proses yang dipandang secara keseluruhan dari gambar (3.9)

Nama sistem/proses → “sistem kalkulator” atau “sistem perhitungan”

Step 2 : Menentukan pengguna sistem

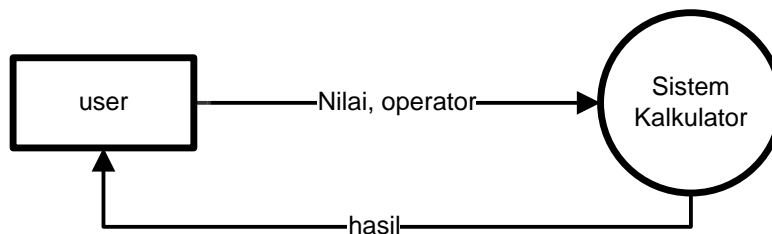
Pengguna → “user”

Dalam contoh kasus ini pengguna bisa siapa saja

Step 3 : Data apa saja yang diberikan ke sistem

Data → nilai (angka), operator (fungsi perkalian, pertambahan dsb)

Step 4 : Menggambar DAD Level Konteks



Gambar 3.10 Diagram konteks Kalkulator

SKENARIO 2

Minimarket milik pak joko menjual mulai peralatan rumah tangga , alat tulis, dan barang kelontong untuk kebutuhan sehari-hari. Pak joko ingin membuat sistem yang bisa digunakan untuk transaksi penjualan di kasir. Sebagai seorang analis anda diminta untuk menggambarkan diagram konteks dari sistem kasir tersebut.

Step 1 : Menentukan sistem atau proses yang dipandang secara keseluruhan dari gambar (3.3)

Nama sistem/proses : “sistem informasi minimarket”

Step 2 : Menentukan pengguna sistem

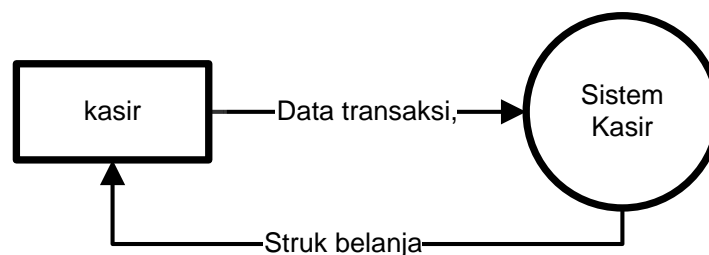
Pengguna sistem : kasir

Step 3 : Data apa saja yang diberikan ke sistem

Input : data transaksi

Output : struk belanja

Step 4 : Menggambar DAD Level Konteks



Gambar 3.11 Diagram konteks Sistem Kasir

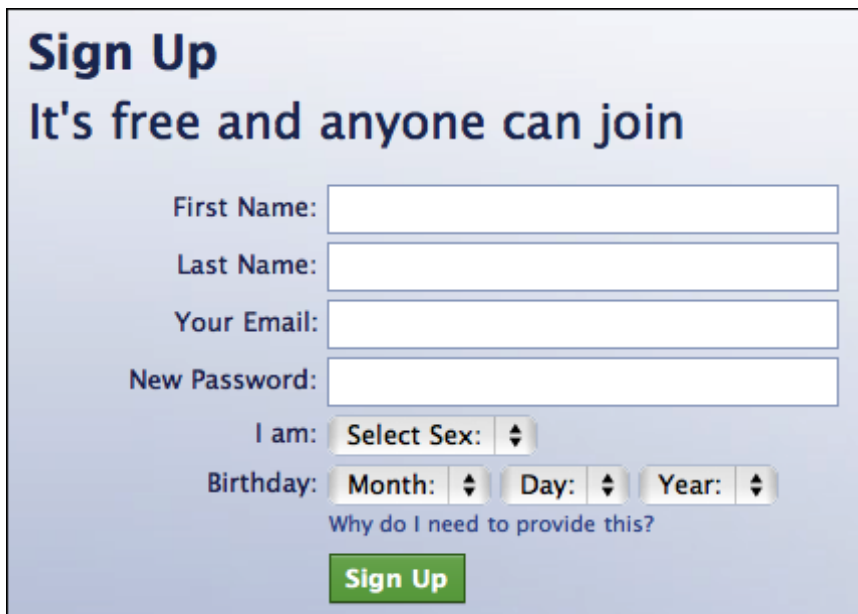
Latihan :

1. Gambarkan Diagram konteks dari search engine google



Gambar 3.12 Search Engine Google

2. Pada saat ingin menggunakan facebook, pengguna diwajibkan untuk mendaftar terlebih dahulu (sign up). User harus menginputkan data-data pribadi untuk proses pendaftaran. Gambarkan diagram konteks dari proses pendaftaran pada facebook

The image shows a screenshot of the Facebook 'Sign Up' page. At the top, it says 'Sign Up' in large blue letters, followed by 'It's free and anyone can join' in smaller blue letters. Below this, there are four text input fields: 'First Name:', 'Last Name:', 'Your Email:', and 'New Password:'. Under the 'New Password' field, there is a link that says 'Why do I need to provide this?'. Below the input fields, there is a section for 'I am:' with a 'Select Sex:' dropdown menu. To the right of this, there is a 'Birthday:' section with three dropdown menus for 'Month:', 'Day:', and 'Year:'. At the bottom of the form, there is a green button with the text 'Sign Up' in white.

Gambar 3.12 Sign Up Facebook

II. DAD Level 1

Setelah pembuatan DAD Level Konteks, selanjutnya adalah pembuatan DAD Level 1, dimana pada DAD Level adalah penggambaran dari Diagram Konteks yang lebih rinci (Overview Diagram) atau biasanya disebut sebagai **dekomposisi**.

Diagram ini adalah dekomposisi dari diagram Context.

Cara :

1. Tentukan proses-prose yang ada pada sistem.
2. Tentukan apa yang diberikan/diterima masing-masing proses ke/dari sistem sambil memperhatikan konsep keseimbangan (alur data yg keluar/masuk dari suatu level harus sama dengan alur data yg masuk/keluar pada level berikutnya)
3. Apabila diperlukan, munculkan data store sebagai sumber maupun tujuan alur data.
4. Gambarkan diagram level satu.
5. Hindari perpotongan arus data
6. Beri nomor pada proses utama (nomor tidak menunjukkan urutan proses). Misal. 1.0, 2.0, 3.0 dst

SKENARIO

Minimarket milik pak joko menjual mulai peralatan rumah tangga , alat tulis, dan barang kelontong untuk kebutuhan sehari-hari. Pak joko ingin membuat sistem yang bisa digunakan untuk transaksi penjualan di kasir. Sebagai seorang analis sistem, anda diminta untuk menggambarkan DAD level 1 dari sistem kasir tersebut.

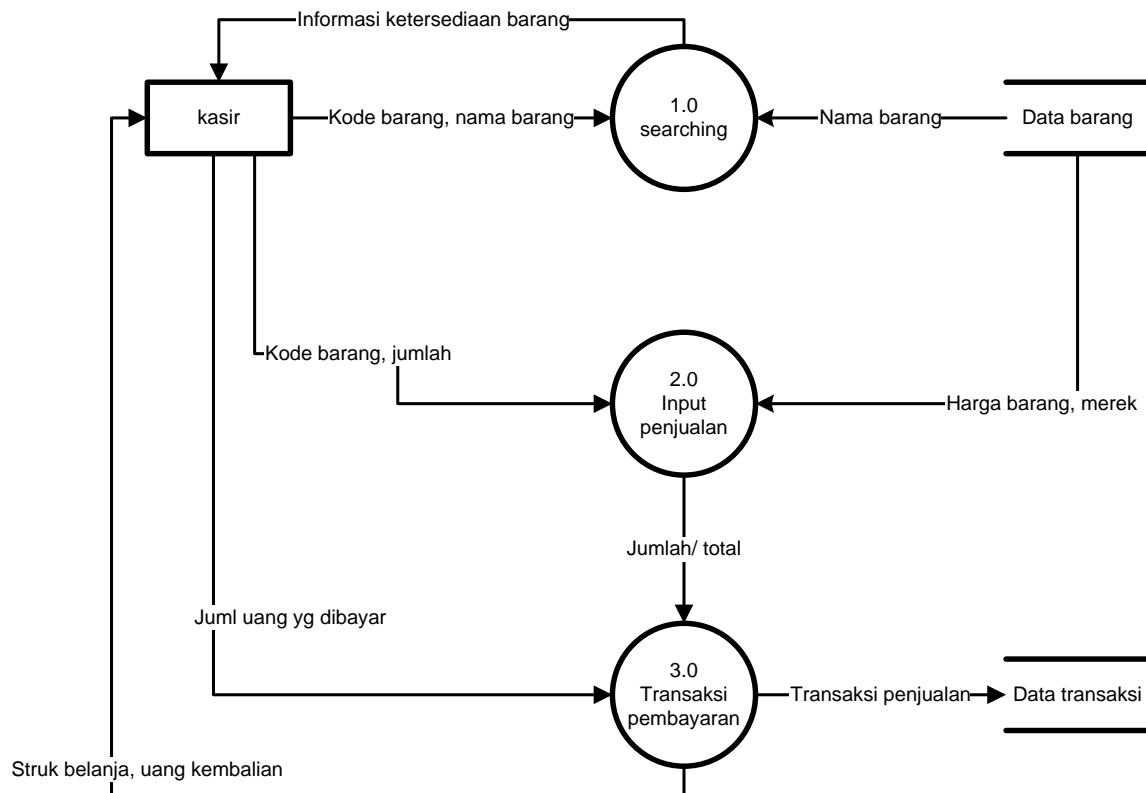
Step 1 : Menentukan proses-proses/ event yang terjadi pada kasir

Proses : proses pencarian barang, input transaksi, pembayaran, cetak struk.

Step 2 : Menentukan arus data yang mengalir (input dan output) disetiap proses.

Step 3 : Menggunakan storage untuk menyimpan data

Step 4 : Menggambarkan ke dalam DAD



Gambar 3.13 DAD Level 1 Sistem Kasir

Latihan :

1. Gambarkan diagram konteks dan DAD Level 1 untuk sistem ATM



Gambar 3.14 ATM

2. Sistem apotek ini bisa digunakan oleh apoteker untuk melayani berbagai macam penjualan obat. Sistem ini dapat juga digunakan oleh asisten apoteker untuk manajemen data stok obat. Lakukan analisis sistem, kemudian gambarkan diagram konteks dan level 1 dari sistem informasi apotek tersebut.

III. DAD Level 2

DAD Level 2 merupakan diagram yang dibentuk dari dekomposisi proses yang terdapat pada DAD Level 1. Tidak semua proses yang terdapat pada DAD Level 1 harus di down grade (dekomposisi) ke dalam DAD Level 2, melainkan sesuai dengan kebutuhan. Jika proses yang terdapat di Diagram Level 1 butuh mencakup banyak proses di dalamnya, maka hal ini perlu dikerjakan ke dalam Diagram Level 2.

SKENARIO 1

Minimarket milik pak joko menjual mulai peralatan rumah tangga , alat tulis, dan barang kelontong untuk kebutuhan sehari-hari. Pak joko ingin membuat sistem yang bisa digunakan untuk transaksi penjualan di kasir. Sebagai seorang analis sistem, anda diminta untuk menggambarkan DAD level 2 dari sistem kasir tersebut.

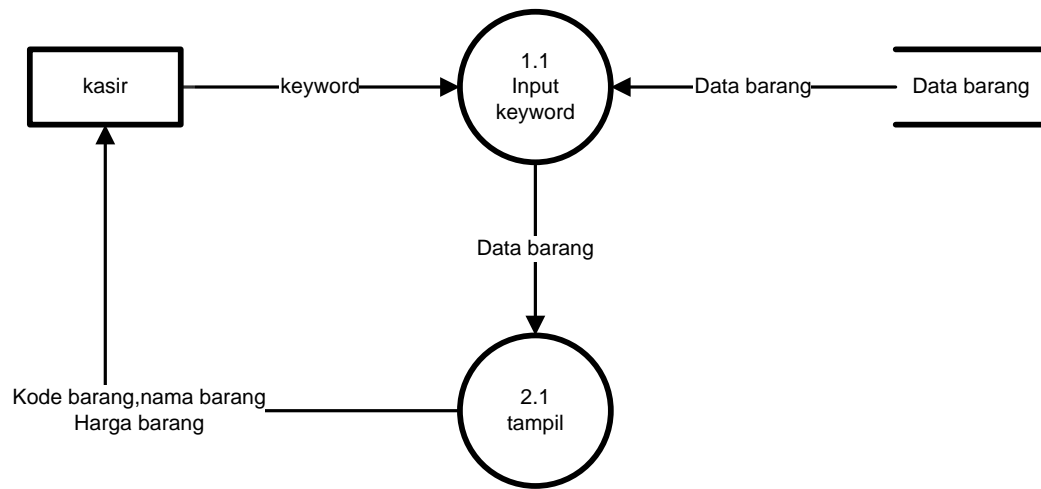
Step 1 Menentukan proses/event dari proses yang terdapat pada DAD Level 1

Proses-proses yang terdapat pada proses searching adalah proses input keyword dan proses tampil

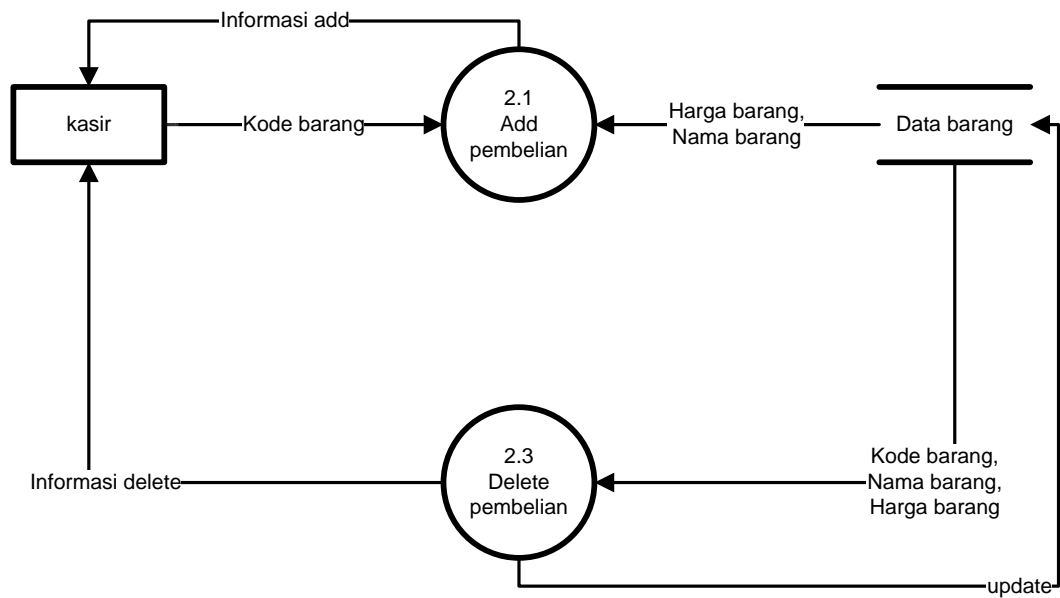
Step 2 : Menentukan arus data yang mengalir (input dan output) disetiap proses.

Step 3 : Menggunakan storage untuk menyimpan data

Step 4 : Menggambarkan ke dalam DAD



Gambar 3.15 DAD Level 2 Proses Searching



Gambar 3.16 DAD Level 2 proses input penjualan

IV. DAD Level 3 dan Level selanjutnya

DAD Level 3 merupakan diagram yang dibentuk dari dekomposisi proses yang terdapat pada DAD Level 2.

Catatan : DAD level tiga, empat dst merupakan dekomposisi dari level sebelumnya. Proses dekomposisi dilakukan sampai dengan proses siap dituangkan ke dalam program. Aturan yang digunakan sama dengan level dua.

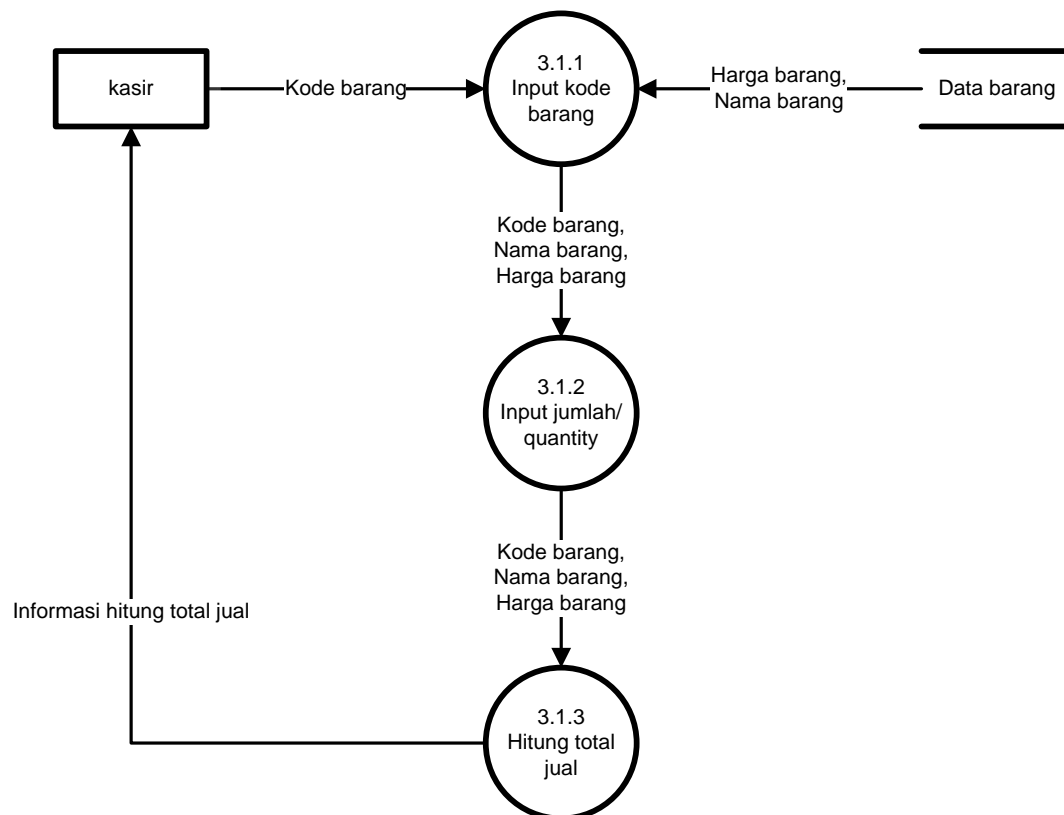
SKENARIO

Step 1 Menentukan proses/event dari proses yang terdapat pada DAD Level 2

Proses-proses yang terdapat pada proses searching adalah proses input keyword dan proses tampil

Step 2 : Menentukan arus data yang mengalir (input dan output) disetiap proses.

Step 3 : Menggunakan storage untuk menyimpan data



Gambar 3. 17 DAD Level 3 Proses Add Pembelian

Latihan :

Minimarket milik pak Joko menjual mulai peralatan rumah tangga , alat tulis, dan barang kelontong untuk kebutuhan sehari-hari. Pak Joko ingin membuat sistem yang bisa digunakan untuk transaksi jual beli di kasir dan manajemen data barang dagangan oleh orang kepercayaan pak Joko (admin). Selain itu sistem bisa digunakan untuk pengelolaan laporan keuangan oleh akuntan . Pak Joko sebagai direktur dapat menggunakan fitur melihat dan mencetak laporan. Sebagai seorang analis yang handal, anda bertugas untuk menganalisis kebutuhan sistem, kemudian gambarkan diagram konteks, DAD level 1, DAD level 2, dan seterusnya (jika perlu).

TUGAS

Disebuah daerah yang jauh dari perkotaan, terdapat sebuah perpustakaan. Semakin hari pengguna perpustakaan tersebut semakin bertambah, sehingga pemilik perpustakaan membutuhkan aplikasi komputer yang dapat digunakan untuk aktivitas di perpustakaan. Sebagai seorang analis anda diminta untuk membuat perancangan perangkat lunak, meliputi :

- Analisis Sistem
- Analisis Kebutuhan Fungsional dan Non Fungsional
- DAD (Konteks, DAD level 1, DAD level 2)

Ilustrasi aktivitas yang terjadi pada perpustakaan dapat anda lihat pada gambar 3.18.



Gambar 3.18 Suasana/Aktivitas di Perpustakaan

PERTEMUAN IV

BAGAN ALIR (FLOWCHART)

Tujuan Pembelajaran

1. Mengenalkan Flowchart sebagai tool perancangan sistem
2. Pengenalan Jenis-jenis flowchart dan aplikasi yang sesuai
3. Mengenalkan studi kasus dengan menggunakan tools flowchart

Keluaran Pembelajaran (Skill)

1. Agar mahasiswa dapat menjelaskan fungsi flowchart dalam perancangan sistem
2. Agar mahasiswa dapat membedakan aplikasi mana yang menggunakan flowchart sebagai tools perancangan sistem.
3. Agar mahasiswa dapat menyebutkan jenis flowchart serta aplikasi yang sesuai.
4. Membuat flowchart dari studi kasus yang diberikan oleh dosen dengan benar.

Alat dan bahan : Microsoft Visio

Teori

Flowchart merupakan alat bantu yang bisa digunakan untuk kegiatan analisa sistem dan perancangan (desain) sistem. Suatu skema representasi suatu proses atau algoritma. Flowchart merupakan salah satu tool yang digunakan untuk Quality Control. **Flowchart adalah** bentuk gambar/diagram yang mempunyai aliran satu atau dua arah secara sekuensial. Flowchart digunakan untuk merepresentasikan maupun mendesain program.

Oleh karena itu flowchart harus bisa merepresentasikan komponen-komponen dalam bahasa pemrograman. Baik flowchart maupun algoritma bisa dibuat sebelum maupun setelah pembuatan program. Flowchart dan Algoritma yang dibuat sebelum membuat program digunakan untuk mempermudah pembuat program untuk menentukan alur logika program, sedangkan yang dibuat setelah pembuatan program digunakan untuk menjelaskan alur program kepada orang lain.

Flowchart berbeda dengan DAD, yang paling jelas adalah dari tahapan/langkah alur data serta simbol-simbol yang digunakan. Untuk itu, perhatikan penjelasan berikut sehingga bisa mengetahui apa saja perbedaan yang ada pada flowchart dan DAD.

Macam-macam Flowchart

1. Bagan Alir Sistem (System Flowchart)

Merupakan bagan yang menunjukkan arus pekerjaan secara keseluruhan dari sistem.

Contoh. Pendaftaran pasien di RS, Pendaftaran matakuliah praktikum, dll.

2. Bagan Alir Dokumen (Document Flowchart)

Merupakan bagan alir yang menunjukkan arus data dari laporan dan formulirformulir termasuk tembusannya.

Contoh. Pelaporan bulanan Perusahaan.

3. Bagan Alir Skematik (Schematic Flowchart)

Menggambarkan prosedur di dalam sistem. Bagan ini menggunakan simbol bagan alir sistem, juga menggambarkan komputer dan peralatan lainnya.

Contoh. Bagan alir proses robot, bagan alir proses pencetakan dokumen.

4. Bagan Alir Program (Program Flowchart)




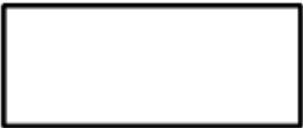
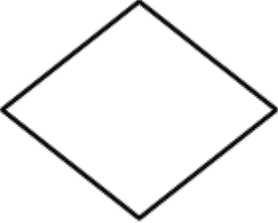
Merupakan bagan yang menjelaskan secara rinci langkah-langkah dari proses program.

Contoh. Bagan alir untuk proses penghitungan faktorial, bagan alir untuk proses penghitungan suhu ruangan.

5. Bagan Alir Proses (Process Flowchart)

Merupakan bagan alir yang banyak digunakan di teknik (misal. teknik industri).

Tabel 4. 1 Simbol Flowchart

Nama	Simbol	Keterangan
Oval		Menunjukkan notasi untuk awal dan akhir bagan alir
Aliran data		Menunjukkan petunjuk dari aliran fisik pada program
Data		Menunjukkan suatu operasi input atau suatu operasi output.
Rectangle		Menunjukkan suatu proses yang akan digunakan
Diamond		Menotasikan suatu keputusan (atau cabang) yang akan dibuat. Program akan memilih satu dari dua rute.

Keterangan :

- Tanda Oval, biasanya disebut sebagai *start and end symbol*. Biasanya diberi nama “Mulai” dan “Selesai”, atau “Start” dan “End”. Ataupun frase lain yang menunjukkan bahwa program/bagan alir tersebut mulai dan selesai.
- Diamond sebagai tanda pemilihan adalah **Conditional** (or **decision**). Simbol ini mengandung pertanyaan Yes/No atau Ya/Tidak atau Benar/Salah. Simbol ini memiliki dua simbol aliran data yang keluar, biasanya dari sudut yang bawah dan sudut yang kanan, satunya menyatakan “Benar(true)” dan satunya menyatakan “False(salah)”.

SKENARIO

Menghitung Faktorial N ($N!$) dimana $N! = 1 * 2 * 3 * 4 * 5 \dots N$

Fungsi dari bagan alir untuk menghitung N faktorial ini akan mempermudah kita dalam menuliskan pada program komputer, karena dari bagan alir ini, kita akan mudah untuk memahami algoritma yang digunakan, karena flowchart ini merupakan tahapan suatu instruksi seperti halnya algoritma suatu program.

Steps 1

Tentukan inisialisasinya, misalnya :

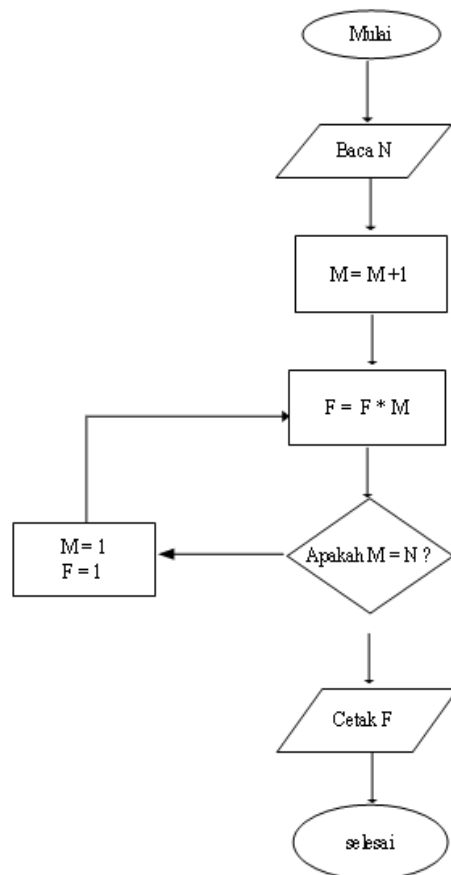
N = bilangan faktorial yang dicari , M = bilangan 1,2,3,4, ... N

Steps 2

Tentukan rumus untuk menghitung faktorial

Steps 3

Gambarkan ke dalam flowchart



Gambar 4.1 Flowchart

Latihan :

Buatlah Flowchart program untuk menghitung dan menampilkan volume balok dengan rumus (panjang x lebar x tinggi) sedangkan panjang, lebar dan tinggi balok diinputkan melalui keyboard pada saat program dieksekusi.

Tugas :

Buatlah Flowchart Rental VCD.

Yang perlu diperhatikan saat akan membuat diagram alir proses peminjaman VCD yaitu :

- Investigasi data
- Wawancara/Observasi sistem
- Narasi
- Investigasi Data
 1. Kartu Anggota
 2. Fotokopi KTP/SIM
 3. Lembar Identitas
 4. Nota Peminjaman
 5. Data VCD
 6. Laporan bulanan
- Wawancara

Hasil Wawancara

 1. Orang yang terlibat: Anggota, adm, manager
 2. Calon anggota harus mendaftar dengan membawa identitas diri (KTP/SIM)

Hasil Quisioner

 1. Keterlambatan tidak didenda didenda
 2. Kesulitan dalam mencari data vcd pada arsip vcd
- Narasi
 1. Calon anggota penyewaan vcd mengisi lembar identitas dan memberikan kartu pengenalan SIM/KTP
 2. Petugas memeriksa apakah data tersebut sudah ada pada arsip anggota

3. Jika tidak ada petugas akan membuat kartu anggota baru dan memberikannya pada anggota
4. Pada saat meminjam anggota harus menunjukan kartu anggota dan memberikan data vcd yang akan dipinjam
5. Petugas akan mencari dari arsip vcd Jika vcd tidak ada atau sedang dipinjam maka petugas akan memberitahukan status kosong ke anggota Jika ada maka petugas akan membuat nota peminjaman dan memberikannya ke anggota

PERTEMUAN V

DOKUMEN PERANCANGAN SISTEM

Tujuan Pembelajaran

1. Membuat hal-hal yang berkaitan dengan perancangan sistem
2. Mendokumentasikan hal-hal yang berkaitan dengan perancangan sistem

Alat dan bahan :

Microsoft Office Visio

Keluaran Pembelajaran (Skill)

1. Mahasiswa dapat membuat dokumen perancangan sistem secara lengkap
2. Mahasiswa dapat mendokumentasikan analisis sistem
3. Mahasiswa dapat mendokumentasikan analisis kebutuhan fungsional dan kebutuhan non fungsional
4. Mahasiswa dapat mendokumentasikan DAD
5. Mahasiswa dapat mendokumentasikan Flowchart

Teori

Dalam membangun perangkat lunak dilakukan perancangan untuk melakukan proses pembuatan maupun pengembangan sistem. Proses tersebut antara lain mulai dari menganalisis sistem yang akan dibangun/dikembangkan, kemudian membuat perncangan data. Yaitu dengan membuat analisis sistem, batasan sistem, DAD , dan flowchart yang akan digunakan untuk membangun atau mengembangkan perangkat lunak tersebut.

Skenario

Step 1 Menuliskan Analisis Sistem

Step 2 Menuliskan Analisis Kebutuhan Fungsional dan Non Fungsional Sistem

Step 3 Menggambarkan DAD lengkap (Diagram konteks, DAD level 1, DAD level 2 dst sesuai dengan kebutuhan)

Step 4 Menggambarkan Flowchart

Tugas :

1. Buatlah dokumentasi perancangan Sistem Informasi Perpustakaan yang telah anda kerjakan di BAB III
2. Buatlah dokumentasi perancangan Sistem Informasi Akademik pada kampus anda.

BAB VI

USE CASE DIAGRAM

Tujuan pembelajaran

1. Mengenalkan UML sebagai paradigma baru perancangan perangkat lunak
2. Mengenalkan konsep use case diagram
3. Menjelaskan cara membuat use case diagram menggunakan tool

Keluaran pembelajaran

1. Mahasiswa dapat membedakan metode perancangan perangkat lunak berbasis objek dengan yang konvensional
2. Mahasiswa dapat mengoperasikan tool pembuatan UML
3. Mahasiswa dapat menjelaskan fungsi use case diagram dan semua notasi yang digunakan di dalamnya
4. Mahasiswa dapat membuat use case diagram untuk contoh kasus yang diberikan

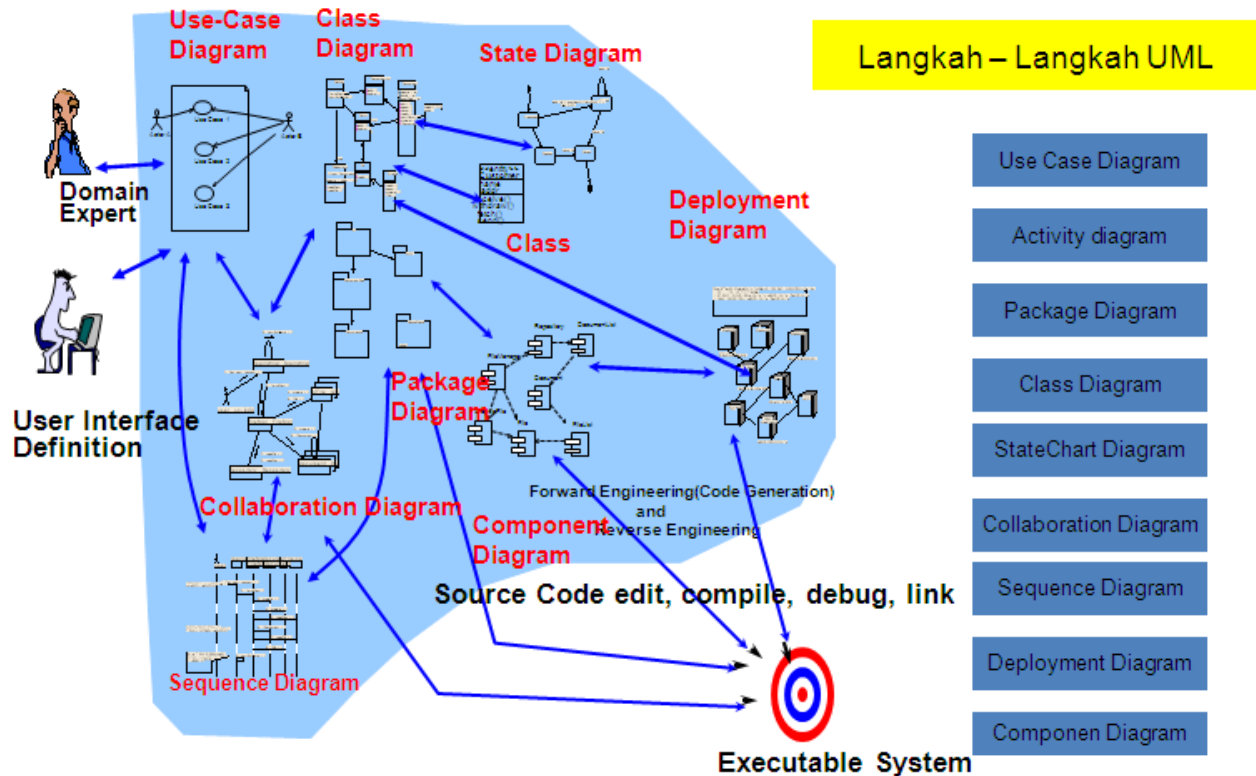
Alat dan Bahan

StarUML

Teori

Unified Modelling Language (UML) adalah sebuah "bahasa" yg telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem perangkat lunak. UML menawarkan sebuah standar untuk merancang model sebuah sistem. Dengan menggunakan UML kita dapat membuat model untuk semua jenis aplikasi perangkat lunak, dimana aplikasi tersebut dapat berjalan pada perangkat keras, sistem operasi dan jaringan apapun, serta ditulis dalam bahasa pemrograman apapun. Tetapi karena UML juga menggunakan class dan operation dalam konsep dasarnya, maka ia lebih cocok untuk penulisan perangkat lunak dalam bahasa-bahasa berorientasi objek seperti C++, Java, C# atau VB.NET. Walaupun demikian, UML tetap dapat digunakan untuk pemodelan aplikasi prosedural dalam VB atau C.

Pada UML mendefinisikan diagram-diagram pada gambar :



Tabel 6.1 Use case model

Use Case Text	Use Case Diagram
- Lebih detail	- Lebih abstrak, kurang detail
- Tidak ada visualisasi (berbentuk teks)	- Bentuk visual (gambar)
- Cocok bagi developer	- Cocok untuk client

1. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Use case merepresentasikan interaksi antara aktor dengan sistem. Use case menggambarkan pekerjaan tertentu, misalnya login ke sistem, membuat daftar belanja, dan sebagainya.

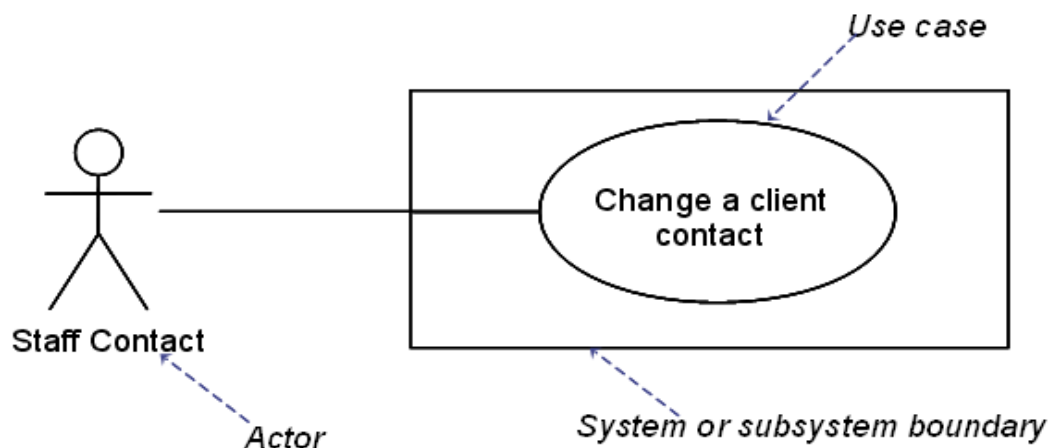
Aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu ketika kita

sedang menyusun analisis kebutuhan sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua fitur yang ada pada sistem. Kadangkala notasi use case kurang detail, terutama untuk beberapa kegiatan tertentu.

Use case dapat memasukkan (**include**) fungsionalitas use case lain sebagai bagian dari proses dalam dirinya. Secara umum diasumsikan bahwa use case yang di-include akan dipanggil setiap kali use case yang meng-include dieksekusi secara normal. Sebuah use case dapat di-include oleh lebih dari satu use case lain, sehingga duplikasi fungsionalitas dapat dihindari dengan cara menarik keluar fungsionalitas yang sejenis.

Use case juga dapat meng-**extend** use case lain dengan behaviour-nya sendiri. Sementara hubungan generalisasi antar use case menunjukkan bahwa use case yang satu merupakan spesialisasi dari yang lain.

Use Case Diagram menggambarkan interaksi antara aktor dengan proses atau sistem yang dibuat. Use Case Diagram mempunyai beberapa bagian penting seperti : Actor, Use Case, Association, Generalization.



Gambar 6.2 Actor, use case dan sistem boundary

1. Actor

Actor merupakan bagian dari Use Case yang bertindak sebagai subjek (pelaku) dalam suatu proses.


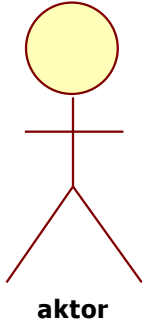
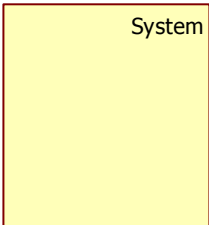
2. Use Case





Use Case adalah proses-proses yang terjadi dalam suatu software. Use Case juga menggambarkan apa yang sedang dilakukan oleh seorang Actor.

3. Relasi

Relasi menggambarkan hubungan antara actor dan use case.

Tabel 6.2 Simbol-simbol dalam use case diagram

Simbol	Deskripsi
<p>Use Case :</p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor; atau apa yang dilakukan oleh sistem /apa yang terjadi pada sistem . Biasanya dinyatakan dengan menggunakan kata kerja di awal di awal frase nama uce case</p>
<p>Aktor :</p> 	<p>Merepresentasikan orang, peran atau system lain yang berinteraksi dengan sistem</p>
<p>Boundary (sistem) :</p> 	<p>Merupakan batas antara sistem dan aktor</p>

<p>Generalization :</p> 	<p>Hubungan antara induk dan anak dengan Induk disebut root / base</p> <p>Terbagi menjadi 2 :</p> <ul style="list-style-type: none"> ➤Actor Generalization ➤Use Case Generalization <p>Generalization/inheritance dipakai ketika ada sebuah keadaan yang lain sendiri/perlakuan khusus (single condition)</p>
<p>Asosiasi / association</p> 	<p>Relasi struktural antar obyek (use case, actor). Dinotasikan dengan garis penghubung antar obyek. Associations bukan menggambarkan aliran data/informasi . Associations digunakan untuk menggambarkan bagaimana actor terlibat dalam use case .</p>
<p><<extend>> :</p> 	<p>Perluasan dari use case lain jika kondisi atau syarat terpenuhi . Tanda panah terbuka harus terarah ke parent/base use case</p>
<p><<include>> :</p> 	<p>termasuk didalam use case lain (required/ diharuskan) . Tanda panah terbuka harus terarah ke sub use case .</p>

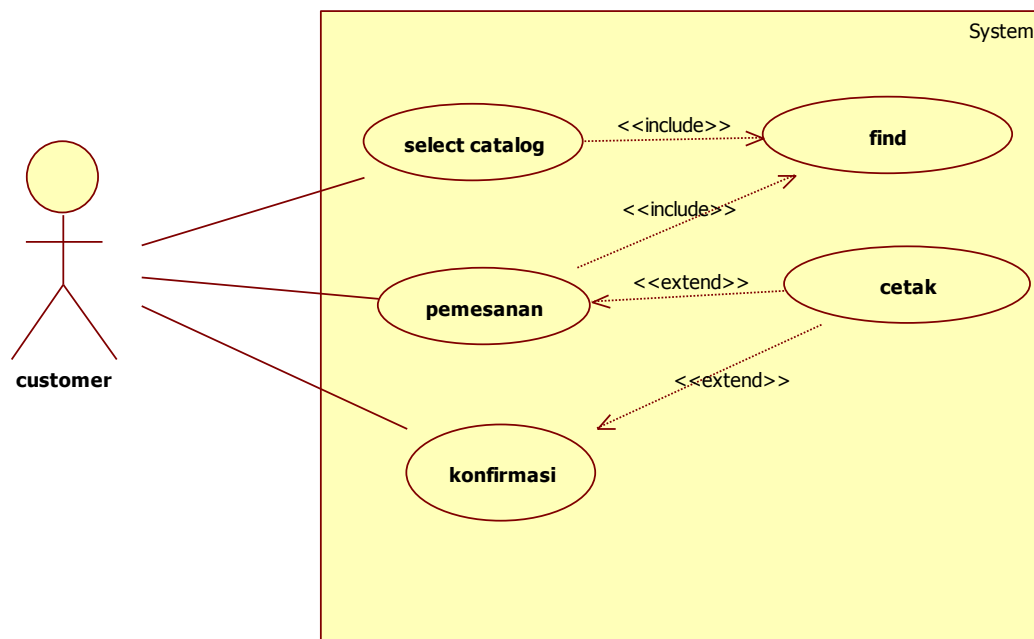
Skenario :

Buatlah use case diagram, jika anda diminta untuk membangun “Sistem Informasi Toko Sepatu Online ”. Analisislah siapa pengguna sistem ini (aktor), dan fitur-fitur (use case & hubungannya) apa saja yang anda sediakan dalam sistem ini. Kemudian jelaskan use case-use case tersebut dan hubungannya dengan use case lain (jika ada).

1. Step 1 : Menentukan aktor dan fitur-fitur atau use case yang akan disediakan oleh sistem

Aktor :

1. Admin
 2. Customer/pelanggan toko sepatu online.
2. Step 2 : Menggambar use case diagram



Gambar 6.3 Use Case Diagram Toko Sepatu Online

Latihan :

1. Buatlah use case diagram, jika anda diminta untuk membangun “Buku Alamat dan Notelp” seperti yellowpages. Analisislah siapa pengguna sistem ini (aktor), dan fitur-fitur (use case & hubungannya) apa saja yang anda sediakan dalam sistem ini. Kemudian jelaskan use case-use case tersebut dan hubungannya dengan use case lain (jika ada).

2. Buatlah use case diagram, jika anda diminta untuk membangun “Sistem Informasi Kos di Jogja”. Analisislah siapa pengguna sistem ini (aktor), dan fitur-fitur (use case & hubungannya) apa saja yang anda sediakan dalam sistem ini. Kemudian jelaskan use case-use case tersebut dan hubungannya dengan use case lain (jika ada).
3. Buatlah use case diagram, jika anda diminta untuk membangun “Sistem Informasi Wisata Jogja”. Analisislah siapa pengguna sistem ini (aktor), dan fitur-fitur (use case & hubungannya) apa saja yang anda sediakan dalam sistem ini. Kemudian jelaskan use case-use case tersebut dan hubungannya dengan use case lain (jika ada).

2. USE CASE TEXT

Use case text merupakan deskripsi dari use case diagram. Adapun Isi dokumentasi :

Name of Use Case	Nama use case
Actor	Pelaku
Pre conditions	kondisi yang harus dipenuhi sebelum use case dijalankan
Post conditions	kondisi yang harus dipenuhi setelah use case dijalankan
Purpose	apa yang akan dicapai oleh use case
Relationship	hubungan dengan use case lain.
Description/Scenario	resume langkah atau tahapan dalam use case

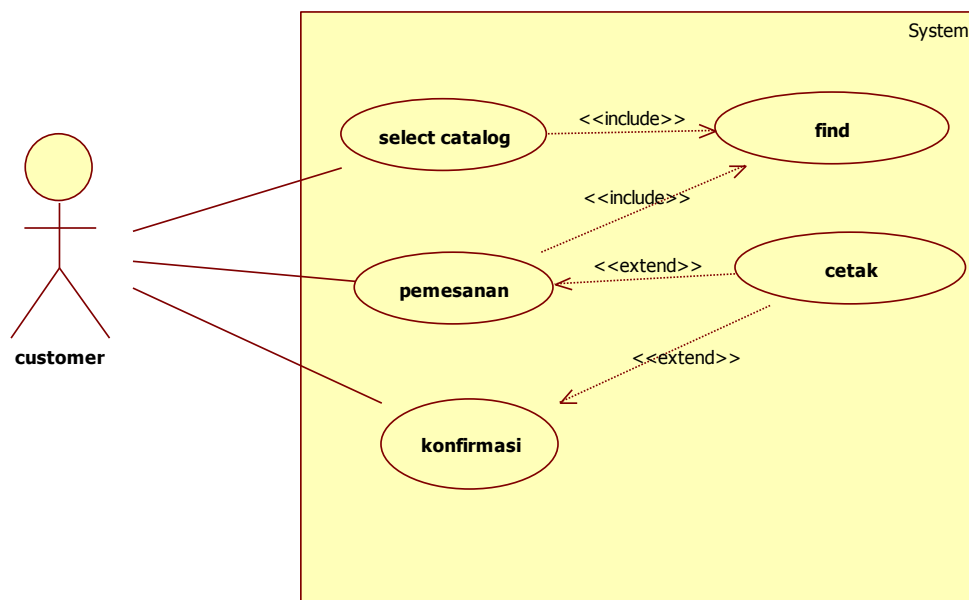
Contoh : (Masih pake skenario yang ada di use case diagram “SI Toko Sepatu Online”, diambil salah satu use case nya)

Name of Use Case	Pemesanan
Actor	Customer
Pre conditions	- Customer telah memilih barang yang akan dipesan.
Post conditions	- Customer mendapatkan nomor pesanan dan total tagihan

Purpose	Use case pemesanan digunakan untuk proses pesan barang.
Relationship	<<extend>> cetak
Description/Scenario	<ul style="list-style-type: none"> – Customer menginputkan data barang yang akan dipesan – Customer menginputkan data nama pemesan, alamat, dan no_Telp – Sistem akan menampilkan nomor pesanan dan total tagihan. – Customer berkesempatan dapat mencetak form pemesanan yang telah terkonfirmasi pada sistem.

Latihan :

Buatlah use case text dari contoh kasus pada use case diagram Sistem Informasi Toko Sepatu Online



Gambar 6.3 Use Case Diagram Toko Sepatu Online

BAB VII

ACTIVITY DIAGRAM

Tujuan pembelajaran

- Mengenalkan konsep activity diagram
- Menjelaskan cara membuat activity diagram menggunakan tool

Keluaran pembelajaran

- Praktikan dapat menjelaskan fungsi activity diagram dan semua notasi yang digunakan di dalamnya
- Praktikan dapat membuat activity diagram untuk contoh kasus yang diberikan

Alat dan Bahan

StarUML

Teori

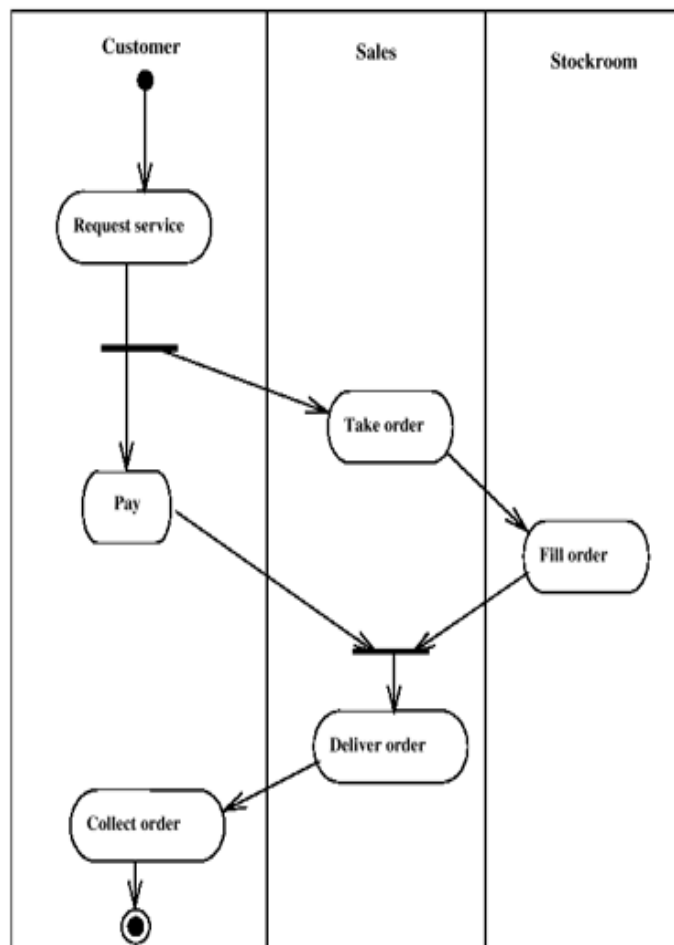
Activity diagrams menggambarkan berbagai alir kegiatan dalam sistem yang sedang dirancang, bagaimana masing-masing alir berawal, decision yang mungkin terjadi, dan bagaimana mereka berakhir. Activity diagram juga dapat menggambarkan proses paralel yang mungkin terjadi pada beberapa eksekusi.

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah action dan sebagian besar transisi di-trigger oleh selesainya state sebelumnya (internal processing). Oleh karena itu activity diagram tidak menggambarkan behavior internal sebuah sistem (dan interaksi antar subsistem) secara eksak, tetapi lebih menggambarkan proses-proses dan jalur-jalur kegiatan dari level atas secara umum.

Sebuah kegiatan dapat direalisasikan oleh satu use case atau lebih. Kegiatan menggambarkan proses yang berjalan, sementara use case menggambarkan bagaimana aktor

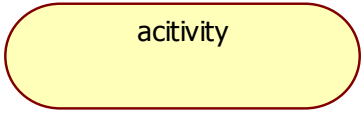


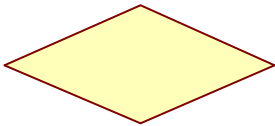

menggunakan sistem untuk melakukan kegiatan. Sama seperti state, standar UML menggunakan segiempat dengan sudut membulat untuk menggambarkan kegiatan. Decision digunakan untuk menggambarkan behaviour pada kondisi tertentu. Untuk mengilustrasikan proses-proses paralel (fork dan join) digunakan titik sinkronisasi yang dapat berupa titik, garis horizontal atau vertikal. Activity diagram dapat dibagi menjadi beberapa object swimlane untuk menggambarkan objek mana yang bertanggung jawab untuk kegiatan tertentu.

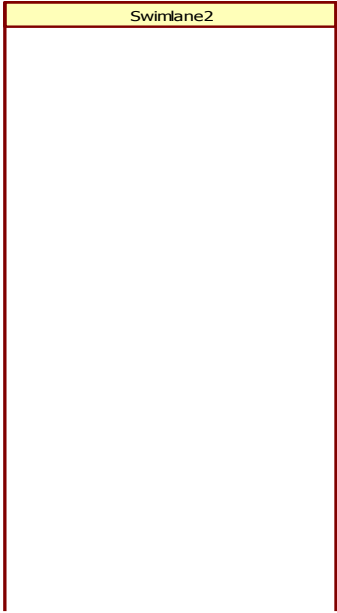
Contoh activity diagram dengan swimlane:



Gambar 7. 1 Activity Dengan Swimline

Tabel 8.1 Simbol-simbol yang terdapat pada Activity Diagram. (ditambahin gambar)

Simbol	Deskripsi
Activity : 	Menggambarkan sebuah pekerjaan/tugas dalam workflow.
Start State : 	Menunjukkan dimulainya suatu workflow pada sebuah activity diagram. Hanya ada satu start state dalam sebuah workflow.
End state : 	Menggambarkan akhir atau terminal dari pada sebuah activity diagram. Bisa terdapat lebih dari satu end state pada sebuah activity diagram.
Decision/Merge 	Suatu titik/point pada activity diagram yang mengindikasikan suatu kondisi dimana ada kemungkinan perbedaan transisi. Salah satu kondisi masuk sudah cukup untuk masuk ke aktivitas berikutnya .
Fork/join 	Proses yang bisa berjalan secara paralel. Syarat maju ke aktivitas berikutnya. Semua aktivitas harus sudah selesai

<p>Swimlane :</p> 	<p>Sebuah cara untuk mengelompokkan activity berdasarkan Actor (mengelompokkan activity dalam sebuah urutan yang sama)</p>
---	--

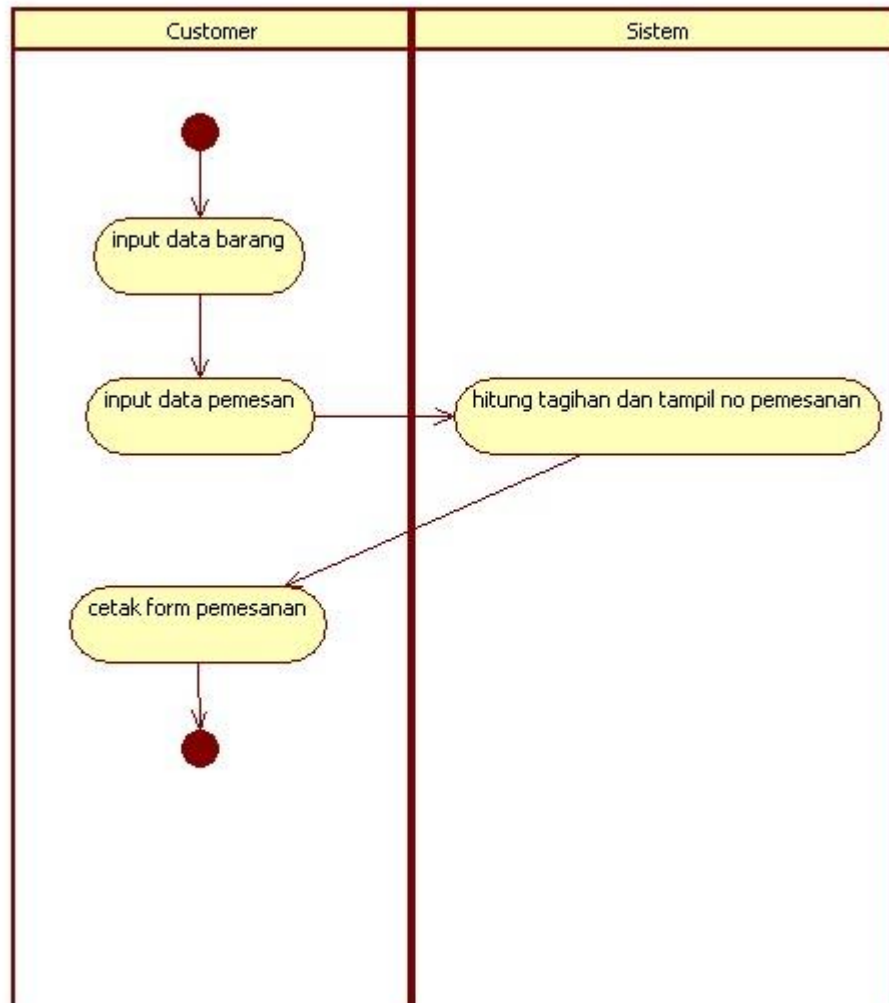
Skenario

1. Dengan contoh skenario yang telah dibahas pada bab sebelumnya, yaitu pada bab use case diagram, buatlah salah satu activity diagram berdasarkan pada description (use case text) yang ada pada salah satu use case yang telah disediakan oleh sistem,

Name of Use Case	Pemesanan
Actor	Customer
Pre conditions	- Customer telah memilih barang yang akan dipesan.
Post conditions	- Customer mendapatkan nomor pesanan dan total tagihan
Purpose	Use case pemesanan digunakan untuk proses pemesanan barang.
Relationship	<<extend>> cetak
Description/Scenario	Customer menginputkan data barang yang akan dipesan Customer menginputkan data nama pemesan, alamat, dan no_Telp

	<p>Sistem akan menampilkan nomor pesanan dan total tagihan.</p> <p>Customer berkesempatan dapat mencetak form pemesanan yang telah terkonfirmasi pada sistem.</p>
--	---

Jawab :



Gambar 7. 2 Diagram Activity Pemesanan

Latihan

1. Buatlah activity diagramnya untuk proses transaksi transfer uang pada ATM.
2. Diberikan use case text dibawah ini, kemudian buatlah activity diagramnya.

Name of Use Case	Peminjaman
Actor	admin
Pre conditions	<ul style="list-style-type: none">– Sistem telah menyimpan data buku– Sistem telah menyimpan data anggota
Post conditions	<ul style="list-style-type: none">– Sistem dapat menyimpan transaksi peminjaman, dan menampilkan tanggal kembali.
Purpose	Use case ini digunakan untuk melaksanakan proses peminjaman.
Relationship	<<include>> use case find_book, <<include>> find_member
Description/Scenario	<ul style="list-style-type: none">– Admin akan memasukkan data peminjaman dengan menginputkan data buku– Sistem akan otomatis menemukan data buku tersebut– Admin menginputkan data peminjam/data anggota– Sistem akan otomatis menemukan data anggota/member tersebut– Sistem akan menyimpan dan menampilkan tanggal kembali sesuai dengan lama peminjaman.

BAB VIII

CLASS DIAGRAM

Tujuan pembelajaran

1. Mengenalkan konsep class diagram
2. Menjelaskan cara membuat class diagram menggunakan tool

Keluaran pembelajaran

1. Praktikan dapat menjelaskan fungsi class diagram dan semua notasi yang digunakan di dalamnya
2. Praktikan dapat membuat class diagram untuk contoh kasus yang diberikan

Alat dan bahan

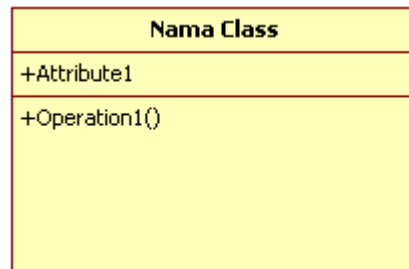
StarUML

Teori

Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut (metoda/fungsi). Class diagram menggambarkan struktur dan deskripsi class, package dan objek beserta hubungan satu sama lain seperti containment, pewarisan, asosiasi, dan lain-lain.

Class memiliki tiga area pokok :

1. Nama (dan stereotype)
2. Atribut
3. Metode (operation)



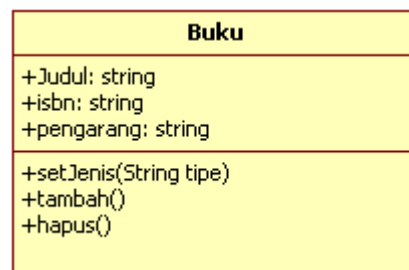
Gambar 8.1 Komponen Class Diagram

Contoh :

Step 1 : menentukan nama objek-objeknya.

Step 2 : Menentukan atribut-atributnya beserta tipe datanya dan sifat aksesnya.

Step 3 : Menambahkan method atau behaviour -nya.

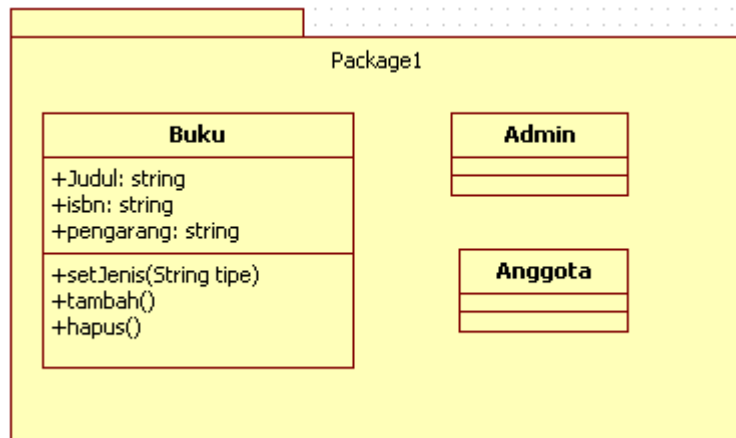


Gambar 8.2 Contoh Class Diagram

Atribut dan metoda dapat memiliki salah satu sifat berikut :

1. Private, tidak dapat dipanggil dari luar class yang bersangkutan
2. Protected, hanya dapat dipanggil oleh class yang bersangkutan dan anak-anak yang mewarisinya
3. Public, dapat dipanggil oleh siapa saja

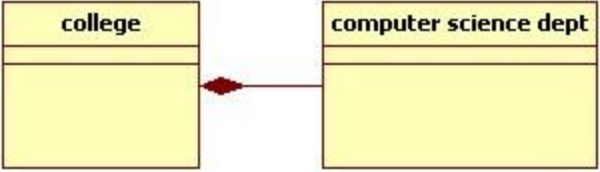
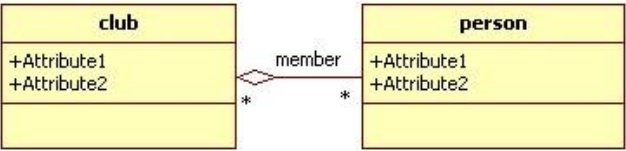
Class dapat merupakan implementasi dari sebuah interface, yaitu class abstrak yang hanya memiliki metoda. Interface tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah class. Dengan demikian interface mendukung resolusi metoda pada saat run-time. Sesuai dengan perkembangan class model, class dapat dikelompokkan menjadi package. Kita juga dapat membuat diagram yang terdiri atas package.



Gambar 8.3 Pengelompokkan class dalam package

Tabel 8.1 adalah tabel simbol-simbol relasi yang digunakan dalam pembuatan class diagram.

Simbol relasi	Deskripsi
<p>Asosiasi / association</p> <pre> classDiagram class Pegawai { +id_peg: int +nama: string } class Divisi { +id_div: int +nama_div: string } Pegawai "1..*" -- "1" Divisi </pre>	<p>Relasi antar kelas dengan makna umum, asosiasi biasanya disertai dengan multiplicity</p>
<p>Generalisasi</p> <pre> classDiagram class Customer { +nama: string +alamat: string } class Corporate { +namaCP } class Personal { +nama +no kartu kredit } Customer < -- Corporate Customer < -- Personal </pre>	<p>Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus)</p>

 <p>Composition</p>	<p>Composition → 'contains a' (whole-part relationship). Bagian yang tidak boleh dipisahkan. Dinotasikan dengan black diamond</p>
 <p>Agregasi / Aggregation</p>	<p>Aggregation → 'has a' atau 'Part-of' relationship (bagian dari terdiri dari). Dinotasikan dengan diamond “kosong”. Jika dipisah, tidak merubah makna</p>



Gambar 8. 4 Contoh relasi class diagram

SKENARIO

Menggambar class diagram :

1. Step 1 : Identifikasi kelas-kelas yg relevan, tentukan atribut-atribut dan metode/operasi kelas
2. Step 2 : Identifikasi Asosiasi dan multiplicity antar Kelas/Objek
3. Step 3 : Identifikasikan keberadaan relasi Generalisasi/Spesialisasi (relasi “is-a”) antar Kelas
4. Step 4 : Identifikasikan keberadaan relasi Agregasi/bagian dari (relasi “part-of”) antar Kelas.
5. Step 5 : Siapkan Diagram Kelas, bangun hasil-hasil diatas dalam diagram kelas tsb.

Latihan :

1. Buat sebuah contoh Kelas/Objek dan relasi/asosiasinya pada satu aplikasi persewaan mobil.
2. Buatlah class diagram dari sistem informasi bengkel.

BAB IX

SEQUENCE DIAGRAM

Tujuan pembelajaran

1. Mengenalkan konsep sequence diagram
2. Menjelaskan cara membuat sequence diagram menggunakan tool

Keluaran pembelajaran

1. Praktikan dapat menjelaskan fungsi sequence diagram dan semua notasi yang digunakan di dalamnya
2. Praktikan dapat membuat sequence diagram untuk contoh kasus yang diberikan

Alat dan bahan

StarUML

Teori



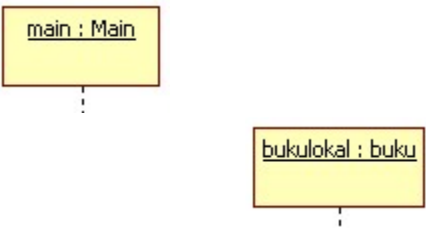

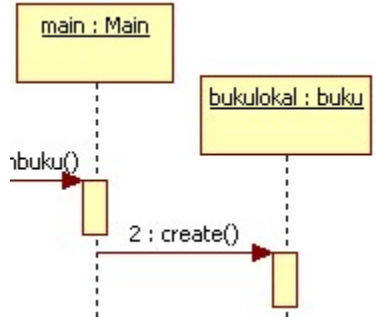
Sequence diagram menggambarkan interaksi antar objek di dalam dan di sekitar sistem (termasuk pengguna, display, dan sebagainya) berupa message yang digambarkan terhadap waktu. Sequence diagram terdiri atas dimensi vertikal (waktu) dan dimensi horizontal (objek-objek yang terkait). Sequence diagram biasa digunakan untuk menggambarkan skenario atau rangkaian langkah-langkah yang dilakukan sebagai respons dari sebuah event untuk menghasilkan output tertentu. Diawali dari apa yang men-trigger kegiatan tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

Masing-masing objek, termasuk aktor, memiliki lifeline vertikal. Message digambarkan sebagai garis berpanah dari satu objek ke objek lainnya. Pada fase desain berikutnya, message akan dipetakan menjadi operasi/metoda dari class. Activation bar menunjukkan lamanya eksekusi sebuah proses, biasanya diawali dengan diterimanya sebuah message.

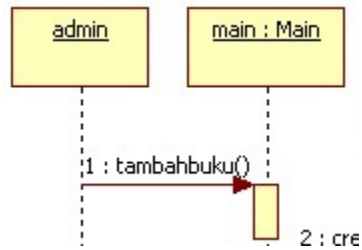
Untuk objek-objek yang memiliki sifat khusus, standar UML mendefinisikan ikon khusus untuk objek boundary, controller dan persistent entity.

Tabel 9.1 adalah tabel simbol-simbol relasi yang digunakan dalam pembuatan sequence diagram.

Tabel 9.1 simbol-simbol sequence diagram

Simbol relasi	Deskripsi
 <p>Aktor</p>	Orang, proses , atau sistem lain yang berinteraksi dengan sistem.
 <p>Lifeline</p>	Menyatakan kehidupan suatu objek.
 <p>Objek</p>	Menyatakan objek yang berinteraksi pesan. Penamaan terdiri dari : nama objek : nama class
 <p>Waktu aktif</p>	Menyatakan objek dalam keadaan aktif dan berinteraksi pesan
 <p>Pesan tipe create</p>	Menyatakan suatu objek membuat objek baru yang lain, arah panah mengarah pada objek yang dibuat. Objek baru yang terbentuk, berada lebih rendah dari pada objek yang lain yang membentuk.

Pesan tipe call

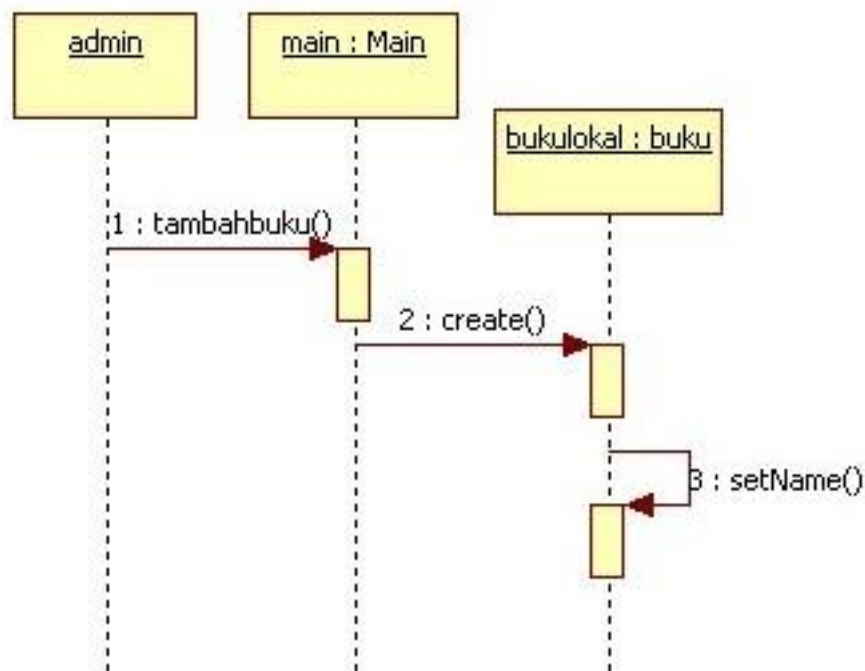


Menyatakan suatu objek memanggil operasi/method yang ada pada objek lain atau dirinya sendiri.

Arah panah mengarah pada objek yang memiliki operasi/method tersebut.

Operasi/method yang dipanggil harus ada pada diagram kelas sesuai dengan kelas objek yang berinteraksi.

Contoh sequence diagram :



Gambar 9.1 Contoh sequence diagram proses login

Latihan :

1. Buatlah sequence diagram proses registrasi mahasiswa baru di UGM. Buat diagramnya menggunakan StarUML.