



PENERBIT ANDI®

ANALISIS dan Perancangan Sistem Informasi

Menggunakan Model Terstruktur dan UML

**Muhamad Muslihudin
Oktafianto**

Bahan dengan hak cipta

Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML

Penulis: Muhamad Muslihudin dan Oktafianto

Hak Cipta © 2016 pada penulis.

Editor : Arie Pramesta
Setter : Vincentius Wahyu
Desain Cover : Ditya
Korektor : Putri Christian

Hak Cipta dilindungi undang-undang. Dilarang memperbanyak atau memindahkan sebagian atau seluruh isi buku ini dalam bentuk apapun, baik secara elektronis maupun mekanis, termasuk memfotocopy, merekam atau dengan sistem penyimpanan lainnya, tanpa izin tertulis dari Penulis.

Penerbit: CV ANDI OFFSET (Penerbit ANDI)

Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282 Yogyakarta 55281

Percetakan: ANDI OFFSET

Jl. Beo 38-40, Telp. (0274) 561881 (Hunting), Fax. (0274) 588282 Yogyakarta 55281

Perpustakaan Nasional: Katalog dalam Terbitan (KDT)

Muslihudin, Muhamad

Analisis dan Perancangan Sistem Informasi Menggunakan Model Terstruktur dan UML/Muhamad Muslihudin dan Oktafianto

– Ed. I. – Yogyakarta: Andi,

23 - 22 - 21 - 20 - 19 - 18 - 17 - 16 - 15

x + 118 hlm; 16 x 23 Cm.

10 9 8 7 6 5 4 3 2 1

ISBN: 978 - 979 - 29 - 5549 - 1

I. Judul

1. Information Systems

2. Oktafianto

DDC'23: 658.403.801.1

DAFTAR ISI

KATA PENGANTAR.....	iii
DAFTAR ISI.....	v

BAB 1 KONSEP DASAR SISTEM INFORMASI.....	1
1.1. Definisi Sistem	1
1.2. Karakteristik Sistem.....	4
1.3. Definisi Subsistem	5
1.4. Sistem yang Buruk.....	8
1.5. Definisi Informasi.....	9
1.5.1.Kualitas Informasi.....	10
1.6. Definisi Sistem Informasi.....	10
1.6.1.Komponen Sistem Informasi.....	12
1.6.2.Tipe-Tipe Sistem Informasi.....	14
1.6.3.Decision Support System	16
1.6.4.Expert Sistem	16

BAB II <i>STAKEHOLDER</i> DALAM SISTEM INFORMASI.....	19
2.1. Stakeholder.....	19
2.1.1. Manajer Sistem Informasi	20
2.1.2. Analisis Sistem	21
2.1.3. Programmer	24
2.1.4. Manajer Bisnis	24
2.1.5. Teknisi Lainnya	25
2.2. Studi Kasus.....	25
 BAB III ANALISIS SISTEM	27
3.1. Definisi Analisis Sistem	27
3.2. Definisi Desain Sistem	28
3.3. Analisis Sistem Informasi	28
3.4. Pendekatan-Pendekatan	
Analisis Model Driven.....	29
3.4.1. Analisis Terstruktur	29
3.4.2. Rekayasa Informasi	
dan Pemodelan Data	29
3.4.3. Analisis Berorientasi Objek	30
3.5. Arti Penting Tahapan Analisis	31
3.6. Rangkuman	32
3.7. Latihan	32
 BAB IV SIKLUS HIDUP PENGEMBANGAN	
SISTEM INFORMASI.....	33
4.1. Pengembangan Sistem Informasi.....	33
4.2. Tahap-Tahap SDLC	34
4.2.1. Tahap Perencanaan	34
4.2.2. Usulan	36
4.2.3. Keputusan Manajemen	36
4.2.4. Kerangka Acuan Kerja	36
4.2.5. Anggaran (Dana)	37
4.2.6. Penunjukan Tim Pelaksana.....	37

4.2.7. Menilai Kelayakan Proyek	37
4.3. Desain Sistem.....	41
4.4. Pembuatan Sistem.....	42
4.5. Implementasi Sistem	43
4.6. Pemeliharaan Sistem	43
 BAB V PERANCANGAN SISTEM MODEL	
TERSTRUKTUR.....	45
5.1. Pengertian Model Struktur	
dari Diagram Konteks	45
5.2. Simbol-Simbol	
dalam Model Struktur	47
5.3. Penjelasan Diagram Konteks.....	48
5.3.1. Syarat Membuat DFD.....	49
5.3.2. Aturan Main Menggambar DFD	50
5.3.3. Cara Membuat DFD	51
 BAB VI PERANCANGAN SISTEM MODEL UML	55
6.1. Analisis dan Desain Berorientasi	
Objek (<i>Object-Oriented Analysis</i>	
<i>and Design Process</i>)	55
6.1.1. Objek (Object).....	56
6.1.2. Kelas (Class)	56
6.1.3. Pembungkusan (Encapsulation)	57
6.1.4. Pewarisan (Inheritance)	
dan Generalisasi/Spesialisasi	57
6.1.5. Polimorfisme.....	58
6.2. Pengertian UML	58
6.2.1. Evolusi UML.....	60
6.2.2. Diagram-Diagram UML.....	61
6.3. Diagram Use Case (Use Case Diagram).....	64
6.3.1. Aktor.....	65

6.3.2. Use Case	68
6.3.3. Relasi Antar-use case / Aktor	71
6.4. Gaya Penulisan Diagram Use Case	77
6.4.1. Panduan Pembuatan Aktor	78
6.4.2. Panduan Pembuatan Use Case	78
6.4.3. Panduan Pembuatan Relasi	79
6.5. Rangkuman	79
BAB VII ENTITY RELATIONSHIP DIAGRAM (ERD)	83
7.1. Definisi Entity Relationship Diagram (ERD)	83
7.2. Notasi ERD	84
7.3. Derajat Relasi atau Kardinalitas	85
7.4. Tahapan-Tahapan ERD	86
BAB VIII PERANCANGAN ANTARMUKA	91
8.1. Aplikasi Antarmuka	91
8.2. Merancang Antarmuka Pengguna	92
8.3. Studi Kasus	93
BAB IX TESTING, IMPLEMENTASI, DAN PEMELIHARAAN SISTEM	97
9.1. Testing (Pengujian)	97
9.1.1. Filosofi Pengujian	98
9.2. Metode Pengujian Sistem	99
9.3. Sub-Testing	100
9.3.1. Unit Testing	100
9.3.2. Integration Testing	101
9.3.3. Pengujian Sistem	102
9.3.4. Acceptance Testing	102

9.4. Implementasi Sistem	103
9.4.1.Menerapkan Rencana Implementasi	104
9.4.2.Kegiatan Implementasi.....	104
9.4.3.Pemilihan dan Pelatihan Personel.....	105
9.4.4.Pemilihan Tempat dan Instalasi	
Perangkat Lunak dan Keras	105
9.5. Konversi Sistem	106
9.6. Pemeliharaan Sistem	107
9.7. Recovery System	108
9.8. Sistem Enhancement.....	110
9.9. Latihan	111
 DAFTAR PUSTAKA.....	 113
TENTANG PENULIS.....	115

BAB 1

KONSEP DASAR SISTEM INFORMASI

Pokok bahasan

- Definisi Sistem
- Kriteria Sistem
- Subsistem
- Sistem yang Buruk
- Definisi Informasi
- Definisi Sistem Informasi

1.1. Definisi Sistem

Untuk mengawali pembahasan tentang analisis dan perancangan sistem informasi, pemahaman akan sistem terlebih dahulu harus ditekankan. Definisi sistem perkembangan sesuai dengan konteks dimana pengertian sistem digunakan. Berikut akan diberikan beberapa definisi sistem secara umum, para ahli mendefinisikan sistem sebagai berikut:

- a. *Murdick dan Ross* (1993) mendefinisikan sistem sebagai seperangkat elemen yang digabungkan satu dengan lainnya untuk suatu tujuan bersama. Sementara, definisi sistem dalam kamus *Webster's Unbrigid* adalah elemen-elemen

yang saling berhubungan dan membentuk satu kesatuan atau organisasi.

- b. Menurut *Scott* (1996), sistem terdiri dari unsur-unsur seperti masukan (*input*), pengelolaan (*processing*), serta keluaran (*output*).
- c. *Mc. Leod* (1995) mendefinisikan sistem sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. Sumber daya yang mengalir dari elemen output dan untuk menjamin prosesnya berjalan dengan baik, maka dihubungkan dengan mekanisme kontrol.
- d. *Jogianto* (2008) mengemukakan bahwa sistem dapat didefinisikan sebagai kumpulan dari prosedur-prosedur yang mempunyai tujuan tertentu.
- e. Menurut *Abdul Kadir* (2005), sistem adalah sekumpulan elemen yang saling terkait atau terpadu yang dimaksudkan untuk mencapai suatu tujuan.
- f. *Bodnar dan Hoowood* (2009), sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berkaitan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu.
- g. *Indra* (2007) mengemukakan sistem adalah sekumpulan elemen atau subsistem yang saling berhubungan satu dengan yang lain membentuk satu kesatuan untuk melaksanakan suatu fungsi guna mencapai suatu tujuan.

Dari beberapa pengertian di atas dapat disimpulkan bahwa sistem adalah sekumpulan komponen atau jaringan kerja dari prosedur-prosedur yang saling berkaitan dan saling bekerja

*image
not
available*

*image
not
available*

dari suatu sistem ke sistem yang lainnya dengan melalui penghubung suatu subsistem dapat berinteraksi dengan subsistem lainnya membentuk suatu kesatuan.

5. Sistem masukan (*Input*) adalah energi yang dimasukkan ke dalam sistem. Masukan dapat berupa masukan perawatan dan masukan sinyal. Masukan perawatan adalah energi yang dimasukkan supaya sistem dapat beroperasi. Sedangkan masukan sinyal adalah energi yang diproses untuk mendapatkan keluaran.
6. Sistem keluaran adalah energi yang diolah, diklasifikasikan menjadi keluaran yang berguna untuk subsistem lain.
7. Sistem sasaran ialah suatu sistem yang mempunyai tujuan atau sasaran. Jika suatu sistem tidak mempunyai batasan sasaran, maka operasi sistem tidak akan ada gunanya.

1.3. Definisi Subsistem

Suatu sistem dapat terdiri dari bagian-bagian sistem atau subsistem. Contoh: sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem dapat terdiri dari subsistem-subsistem lagi atau terdiri dari komponen-komponen pendukung sistem itu sendiri. Subsistem perangkat keras (*hardware*) dapat terdiri dari alat masukan, alat proses, alat keluaran, dan media penyimpanan.

Subsistem-subsistem itu saling berinteraksi dan saling berhubungan membentuk suatu kesatuan, sehingga tujuan atau sasaran sistem tersebut dapat tercapai. Anda dapat membayangkan, bagaimana seandainya sistem komputer yang

*image
not
available*

*image
not
available*

*image
not
available*

1.5. Definisi Informasi

Beberapa definisi informasi yang dikemukakan oleh para ahli:

- a. *Jogiyanto* (2008), informasi adalah data yang diolah menjadi bentuk yang berguna bagi para pemakainya.
- b. *Sutarman* (2009), informasi adalah sekumpulan fakta (data) yang diorganisasikan dengan cara tertentu, sehingga mereka mempunyai arti bagi si penerima.
- c. *Edhy Sutanta* (2011), informasi merupakan hasil pengolahan data, sehingga menjadi bentuk yang penting bagi penerimanya dan mempunyai kegunaan sebagai dasar dalam pengambilan keputusan yang dapat dirasakan akibatnya secara langsung saat itu juga atau secara tidak langsung pada saat mendatang.
- d. *Krismiaji* dalam *Zabaer* (2005), informasi adalah data yang telah diorganisasi, dan telah memiliki kegunaan dan manfaat.
- e. Menurut *Raymond Mc. Leod*, informasi adalah data yang telah diolah menjadi bentuk yang memiliki arti bagi si penerima dan bermanfaat bagi pengambilan keputusan saat ini atau mendatang.

Dapat disimpulkan bahwa informasi merupakan data yang diolah menjadi bentuk yang berguna untuk membuat keputusan. Informasi berguna untuk pembuat keputusan karena informasi menurunkan ketidakpastian (atau meningkatkan pengetahuan). Informasi menjadi penting karena berdasarkan informasi itu para pengelola dapat mengetahui kondisi objektif perusahaannya. Informasi tersebut merupakan hasil pengolahan data atau fakta yang dikumpulkan dengan metode ataupun cara – cara tertentu.

*image
not
available*

*image
not
available*

*image
not
available*

Prosedur yakni tata cara yang meliputi strategi, kebijakan, metode, dan peraturan-peraturan dalam menggunakan sistem informasi berbasis komputer.

Sementara *Burch* dan *Grudnistki (1986)* berpendapat, sistem informasi yang terdiri dari komponen-komponen di atas disebut dengan istilah blok bangunan (*building block*), yaitu blok masukan (*input block*), blok model (*model block*), blok keluaran (*out block*), blok teknologi (*technology block*), blok basis data (*database block*) dan blok kendali (*control block*). Sebagai suatu sistem, keenam blok tersebut masing-masing saling berinteraksi satu dengan yang lainnya membentuk satu kesatuan untuk mencapai sasarnya.

1. Blok masukan

Input mewakili data yang masuk ke dalam sistem informasi. Input di sini termasuk metode-metode dan media untuk menangkap data yang akan dimasukkan, berupa dokumen-dokumen dasar.

2. Blok model

Blok ini terdiri dari kombinasi prosedur, logika, dan model matematika yang akan memanipulasi data input dan data yang tersimpan di basis data dengan cara tertentu untuk menghasilkan keluaran yang dihasilkan.

3. Blok keluaran.

Produk dari sistem informasi adalah keluaran yang merupakan informasi yang berkualitas dan dokumentasi yang berguna untuk semua tingkat manajemen serta semua pemakai sistem.

4. Blok teknologi.

Teknologi merupakan kotak alat (*tool-box*) dalam sistem informasi. Teknologi digunakan untuk menerima input,

*image
not
available*

*image
not
available*

*image
not
available*

- b. ES menanyakan pertanyaan (yang akan ditanyakan seorang pakar) dan pengguna memberikan jawaban.
- c. Jawaban digunakan untuk menentukan aturan mana yang dipakai dan ES sistem menyediakan rekomendasi berdasarkan aturan yang telah disimpan.
- d. Seorang *knowledge engineer* bertanggung jawab pada bagaimana melakukan akuisisi pengetahuan. Sama seperti seorang analis tetapi menggunakan teknik yang berbeda.

*image
not
available*

*image
not
available*

*image
not
available*

Sebagai pemimpin, manajer tidak harus terlibat langsung pada proses pembuatan sistem informasi. Untuk memonitor pekerjaan dari *stakeholder* yang lain, manajer secara efektif berkomunikasi dengan *stakeholder* yang lain melalui pemain kunci, yaitu analis sistem.

2.1.2. Analisis Sistem

Analisis sistem profesi yang bagus untuk memulai karier di bidang IT. Pekerjaan sebagai analis sistem menawarkan tantangan kerja yang dinamis dan variasi. Analis sistem merupakan individu kunci dalam proses pengembangan sistem. Analis sistem mempelajari masalah dan kebutuhan dari organisasi untuk menentukan bagaimana orang, data, proses, komunikasi, dan teknologi informasi dapat meningkatkan pencapaian bisnis. Seorang analis sistem juga merupakan orang yang paling bertanggung jawab pada proses analisis dan perancangan sistem informasi. Tugas utama dari seorang analis sistem adalah menentukan bentuk sistem yang akan dibangun nantinya. Keputusan ini tidak mudah. Kesalahan menentukan format sistem yang akan dibangun akan berakibat pada gagalnya proyek yang dikerjakan. Oleh sebab itu, seorang analis sistem yang sukses harus memiliki dan dibekali dengan beberapa keahlian spesifik, seperti :

1. Keahlian Analisis

Keahlian analisis diperlukan untuk memahami organisasi yang memerlukan sistem informasi yang akan dibangun. Keahlian analisis digunakan untuk memetakan permasalahan yang dihadapi oleh perusahaan klien yang bisa diselesaikan dengan sistem informasi dan yang tidak. Kemampuan analisis juga diperlukan untuk memecahkan masalah yang telah ditemukan lagi menggunakan teknologi

*image
not
available*

*image
not
available*

*image
not
available*

2.1.5. Teknisi Lainnya

Masih banyak lagi teknisi lain yang terlibat dalam pengembangan sistem. Salah satunya adalah *database administrator*. Untuk perusahaan besar dengan skala data yang besar, data di dalam *database* harus dijamin terorganisasi dengan baik, sehingga ketika aplikasi lain memerlukan transfer data dari *database*, bias dilayani dengan cepat. *Database administrator* juga bertanggung jawab pada keamanan data, baik dari serangan virus maupun pihak luar yang tidak punya hak akses untuk melihat dan mengubah data. Teknisi lainnya adalah teknisi jaringan dan teknisi perangkat keras. Perangkat lunak tidak akan berjalan dengan baik tanpa dukungan perangkat keras yang bekerja dengan baik. Transfer data juga tidak bisa dilakukan jika media-media transfernnya mengalami permasalahan. Untuk itu perlu personel khusus yang ditugaskan merawat perangkat keras maupun infrastruktur jaringannya.

2.2. Studi Kasus

Latihan

Kasus 1

Diskusikan pertanyaan berikut dengan teman-teman satu kelompok *stakeholder* mana yang paling berpengaruh pada kesuksesan pengembangan proyek sistem informasi?

Kasus 2

Akseslah situs web organisasi (Pemerintah, Bisnis, Pendidikan, Sosial). Pelajarilah fasilitas apa yang ditawarkan situs tersebut. Apakah situs tersebut bisa digolongkan ke dalam code generator? Apa alasan Anda?

*image
not
available*

*image
not
available*

*image
not
available*

2. Menentukan tingkat kelayakan kebutuhan sistem baru tersebut ditinjau dari beberapa aspek, di antaranya ekonomi, teknik, operasional, dan hukum.

3.4. Pendekatan-Pendekatan Analisis Model Driven

Model driven adalah analisis yang menggunakan gambar untuk mengkombinasikan masalah-masalah, persyaratan-persyaratan, dan solusi-solusi bisnis. Contoh model yang sering dipakai adalah flowchart dan DFD. Contoh dari analisis dengan pendekatan model driven adalah :

- a. Analisis terstruktur.
- b. Rekayasa informasi.
- c. Analisis berorientasi objek.

3.4.1. Analisis Terstruktur

Analisis terstruktur adalah salah satu pendekatan formal pertama untuk analisis sistem informasi. Analisis ini terfokus pada aliran data dari proses bisnis dan perangkat lunak. Analisis ini disebut *process oriented*. Analisis terstruktur sederhana dalam konsep. Pada analisis menggambarkan serangkaian proses dalam bentuk diagram alir data (*data flow diagram*) yang menggambarkan proses yang ada atau yang diusulkan bersama-sama dengan *input*, dan file mereka.

3.4.2. Rekayasa Informasi dan Pemodelan Data

Rekayasa informasi dulu terfokus pada data terstruktur yang tersimpan pada sebuah sistem. Jadi, dikatakan *entity relationship*. Untuk lebih jelasnya akan dibahas pada bab

*image
not
available*

*image
not
available*

*image
not
available*

BAB IV

SIKLUS HIDUP PENGEMBANGAN SISTEM INFORMASI

Pokok bahasan

- Definisi Perancangan Sistem

4.1. Pengembangan Sistem Informasi

SDLC (*Systems Development Life Cycle*, Siklus Hidup Pengembangan Sistem) atau *Systems Life Cycle* (Siklus Hidup Sistem), dalam rekayasa sistem dan rekayasa perangkat lunak, adalah proses pembuatan dan perubahan sistem serta model dan metodologi yang digunakan untuk mengembangkan sistem-sistem tersebut. Konsep ini umumnya merujuk pada sistem komputer atau informasi. SDLC juga merupakan pola yang diambil untuk mengembangkan sistem perangkat lunak, yang terdiri dari tahap-tahap: rencana (*planning*), analisis (*analysis*), desain (*design*), implementasi (*implementation*), uji coba (*testing*) dan pengelolaan (*maintenance*).

*image
not
available*

*image
not
available*

*image
not
available*

- Sasaran proyek.
- Ruang lingkup pekerjaan.
- Jangka waktu pelaksanaan.
- Prioritas pekerjaan.

4.2.5. Anggaran (Dana)

Berdasarkan kerangka acuan kerja di atas, disusunlah anggaran / dana untuk hardware, software, pelatihan SDM, pemeliharaan dan cadangan untuk keperluan yang tidak terduga.

4.2.6. Penunjukan Tim Pelaksana

Setelah semua kegiatan di atas diketahui, selanjutnya diputuskan apakah pengembangan sistem informasi akan dilakukan oleh perusahaan atau oleh pihak konsultan. Setelah menetapkan pelaksana, diminta untuk memasukkan proposal pelaksanaan sistem informasi sesuai dengan kerangka acuan kerja. Proposal tersebut akan dievaluasi untuk menetapkan apakah proyek tersebut layak dilaksanakan atau tidak.

4.2.7. Menilai Kelayakan Proyek

Penilaian kelayakan proyek mencakup kelayakan operasional, teknis dan ekonomis. Dalam praktik, yang dominan dinilai umumnya aspek ekonomisnya (dana).

- **Kelayakan Operasional**, menyangkut apakah secara operasional sistem yang baru dapat dilaksanakan dengan sumber daya manusia yang tersedia dan metode *training*

*image
not
available*

*image
not
available*

*image
not
available*

4. **Tingkat kompetisi produk harus dapat dideteksi dengan baik.** Tingkat kompetisi akan mempengaruhi pembangunan sistem. Oleh karena itu, pembangunan sistem harus dilakukan dengan berorientasi pada pemakai, karena pemakailah yang akan memberi penilaian terhadap sistem dalam lingkungan kompetisi yang sangat ketat.
5. **Lingkungan operasional sistem.** Di mana sistem akan dioperasikan? Apakah hanya di dalam toko atau dapat diakses secara bebas melalui komputer pribadi atau warnet-warnet? Hal ini akan mempengaruhi perancangan terhadap fleksibilitas sistem.
6. **Sistem harga.** Apakah dalam melakukan transaksi, harga produk didasarkan pada ketentuan yang sudah ditetapkan atau ada aturan-aturan pemberian diskon atau bahkan terjadi tawar-menawar. Pemodelan sistem harga ini tentunya akan menentukan model pemasukan data dalam sistem.

4.3. Desain Sistem

Analisis sistem digunakan untuk menjawab pertanyaan *what?*
Desain sistem digunakan untuk menjawab pertanyaan *how?*
Desain berkonsentrasi pada bagaimana sistem dibangun untuk memenuhi kebutuhan pada fase analisis.

Manfaat desain sistem adalah memberikan gambaran rancang bangun (*blue print*) yang lengkap, sebagai penuntun (*guideline*) bagi programmer dalam membuat aplikasi. Sistem informasi yang terkomputerisasi setidaknya terdiri dari:

- **Hardware:** terdiri dari komponen input, proses, output, dan jaringan.

*image
not
available*

*image
not
available*

*image
not
available*

BAB V

PERANCANGAN SISTEM MODEL TERSTRUKTUR

Pokok bahasan

- Definisi Perancangan Sistem Terstruktur
- Diagram Konteks
- Data Flow Diagram DFD
- Studi Kasus

5.1. Pengertian Model Struktur dari Diagram Konteks

Model struktur merupakan representasi bergambar mengenai kenyataan. Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh *input* ke sistem atau *output* dari sistem. Ia akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh *boundary* (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada *store* dalam diagram konteks.

Diagram konteks merupakan tingkatan tertinggi dalam diagram aliran data dan hanya memuat satu proses, menunjukkan sistem

*image
not
available*

*image
not
available*

*image
not
available*

sama sekali tidak memuat penyimpanan data dan tampak sederhana untuk diciptakan.

2. **Diagram Nol (diagram level-1)** : merupakan satu lingkaran besar yang mewakili lingkaran-lingkaran kecil yang ada di dalamnya. Merupakan pemecahan dari diagram konteks ke diagram nol. Di dalam diagram ini memuat penyimpanan data.
3. **Diagram Rinci** : merupakan diagram yang menguraikan proses apa yang ada dalam diagram nol.

5.3.1. Syarat Membuat DFD

Syarat-syarat pembuatan DFD ini adalah :

1. Pemberian nama untuk tiap komponen DFD.
2. Pemberian nomor pada komponen proses.
3. Penggambaran DFD sesering mungkin agar enak dilihat.
4. Penghindaran penggambaran DFD yang rumit.
5. Pemastian DFD yang dibentuk itu konsiten secara logika.

Fungsi dari Data Flow Diagram adalah :

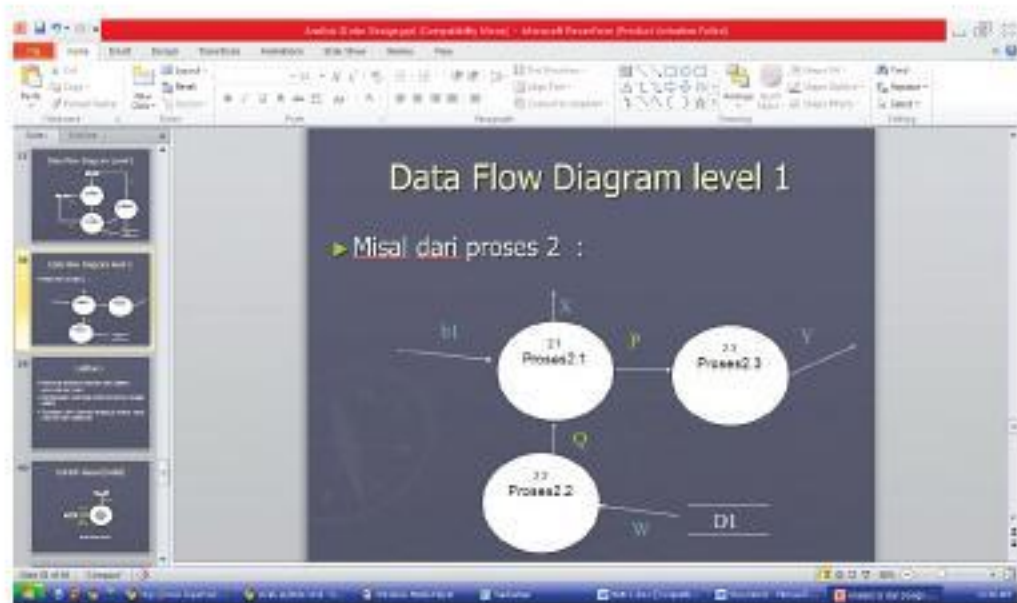
- a. Data Flow Diagram (DFD) adalah alat pembuatan model yang memungkinkan profesional sistem untuk menggambarkan sistem sebagai suatu jaringan proses fungsional yang dihubungkan satu sama lain dengan alur data, baik secara manual maupun komputerisasi.

*image
not
available*

*image
not
available*

*image
not
available*

Data Flow Diagram Level 1



*image
not
available*

*image
not
available*

*image
not
available*

karena terdapat struktur sekaligus karakteristiknya. Kita dapat membentuk kelas baru yang lebih spesifik dari kelas *general-nya*. Kelas dan objek merupakan jantung dari pemrograman berorientasi objek. Untuk menghasilkan program jenis ini sangat penting untuk selalu berpikir dalam konsep objek. Pembahasan mengenai kelas secara lebih rinci dapat dilihat pada Bab III mengenai diagram kelas.

6.1.3. Pembungkusan (*Encapsulation*)

(Nugroho, 2005: 6) mengartikan pembungkusan sebagai penggabungan potongan-potongan informasi dan perilaku – perilaku spesifik yang bekerja pada informasi tersebut, kemudian mengemasnya menjadi apa yang disebut sebagai objek. Dalam perbankan kita mengenal objek rekening yang memiliki perilaku – perilaku misalnya buka, tutup, penarikan, penyimpanan, ubah nama, ubah alamat dan sebagainya. Akibatnya, perubahan – perubahan pada sistem perbankan yang berkaitan dengan rekening-rekening dapat secara sederhana diimplementasikan satu kali saja pada objek rekening. Keuntungan lainnya adalah membatasi efek – efek perubahan pada sistem. Misalnya, saat manajemen bank menentukan jika seseorang memiliki rekening pinjaman di bank yang bersangkutan, rekening pinjaman itu harus dapat juga digunakan sebagai sarana bagi penarikan rekening.

6.1.4. Pewarisan (*Inheritance*) dan Generalisasi/Spesialisasi

Menurut Whitten(2004: 411), konsep dimana metode dan atau atribut yang ditentukan di dalam sebuah objek kelas

*image
not
available*

*image
not
available*

*image
not
available*

7. Diagram Aktivitas (*Activity Diagram*), bersifat dinamis. Diagram aktivitas adalah tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem. Diagram ini terutama penting dalam suatu sistem serta pemodelan fungsi-fungsi suatu sistem dan memberi tekanan pada aliran kendali antarobjek.
8. Diagram Komponen (*Component Diagram*), bersifat statis. Diagram komponen ini memperlihatkan organisasi serta ketergantungan sistem/perangkat lunak pada komponen – komponen yang telah ada sebelumnya. Diagram ini berhubungan dengan diagram kelas dimana komponen secara tipikal dipetakan ke dalam satu atau lebih kelas-kelas, antarmuka-antarmuka serta kolaborasi-kolaborasi.
9. Diagram *Deployment* (*Deployment Diagram*), bersifat statis. Diagram ini memperlihatkan konfigurasi saat aplikasi dijalankan (*run-time*). Memuat simpul – simpul beserta komponen-komponen yang ada di dalamnya. Diagram *deployment* berhubungan erat dengan diagram komponen dimana diagram ini memuat satu atau lebih komponen-komponen. Diagram ini sangat berguna saat aplikasi kita berlaku sebagai aplikasi yang dijalankan pada banyak mesin (*distributed computing*).

Kesembilan diagram ini tidak mutlak harus digunakan dalam pengembangan perangkat lunak, semuanya dibuat sesuai dengan kebutuhan. Pada *UML* dimungkinkan kita menggunakan diagram-diagram lainnya (misalnya *Data Flow Diagram*, *Entity Relationship Diagram* dan sebagainya).

*image
not
available*

*image
not
available*

*image
not
available*

sistem penggajian setiap hari Jumat, karyawan tidak menginisiasi kejadian itu, tapi merupakan penerima utama dari sesuatu yang bernilai.

2. *Primary system actor* (Aktor sistem utama), yaitu *stakeholder* yang secara langsung berhadapan dengan sistem untuk menginisiasi atau memicu kegiatan atau sistem. Pelaku sistem utama dapat berinteraksi dengan para pelaku bisnis utama untuk menggunakan sistem aktual. Mereka memfasilitasi kejadian dengan menggunakan sistem yang melakukan peninjauan daya beli pelanggan, operator telepon yang memberi bantuan kepada pelanggan dan kasir bank yang memproses transaksi bank. Pelaku bisnis utama dan pelaku sistem utama kemungkinan memiliki persamaan, yaitu sama-sama pelaku bisnis yang berhadapan secara langsung dengan sistem, misalnya seorang yang melayani jasa penyewaan mobil via *website*.
3. *External server actor* (Aktor server eksternal), yaitu *stakeholder* yang melayani kebutuhan pengguna *use-case* (misalnya biro kredit yang memiliki kuasa atas perubahan kartu kredit).
4. *External receiving actor* (Aktor penerima eksternal), yaitu *stakeholder* yang bukan pelaku utama, tapi menerima nilai yang terukur atau teramati (*output*) dari *use-case* (misalnya gudang menerima paket permintaan untuk menyiapkan pengiriman sesudah seorang pelanggan memesannya).

Dari banyak sistem informasi, ada kejadian bisnis yang dipicu oleh kalender atau waktu berdasarkan jam.

*image
not
available*

*image
not
available*

*image
not
available*

pesanan kamar karena pemesanan kamar sudut pandangnya aktor tamu, sedangkan pencatatan pesanan sudut pandangnya hotel.

6.3.3. Relasi Antar-*use case* / Aktor

Pada diagram *use case*, relasi digambarkan sebagai sebuah garis antara dua simbol. Pemaknaan relasi berbeda-beda tergantung bagaimana garis tersebut digambar dan tipe simbol apa yang digunakan untuk menghubungkan garis tersebut. Relasi yang digunakan UML 2.0 adalah generalisasi, inklusi dan ekstensi.

6.3.3.1. Generalisasi (*Generalization*)

Generalisasi pada aktor dan *use case* dimaksudkan untuk menyederhanakan model dengan cara menarik keluar sifat-sifat pada aktor-aktor maupun *use case-use case* yang sejenis. *Chonoles (2003)* memberi cara untuk mengetahui kapan dibutuhkan generalisasi berdasarkan tujuan, yakni:

1. Mekanisme berbeda dengan satu tujuan yang sama (Generalisasi *use case*): Jika ada lebih dari satu alternatif teknik dan cara agar aktor dapat mencapai tujuannya, tentu akan diperoleh penggunaan bersama (*sharing*) seperti: peralatan pendukung, *bisnis rules*, validasi data dan sebagainya. Dengan generalisasi kita dapat membuat penggunaan bersama itu secara eksplisit dan mengurangi duplikasi dengan cara membuat satu *use case* baru yang mengakomodasi penggunaan bersama itu.

*image
not
available*

*image
not
available*

*image
not
available*

kelas dan pesan kursi. Pesan tiket disebut *use case* pemanggil (*calling use case*) sedangkan pesan kelas dan pesan kursi disebut *use case* terpanggil (*called use case*), *use case* pesan tiket belum lengkap karena harus pesan kelas dan pesan kursi terlebih dahulu. Konsep orientasi objek memisahkan ketiga *use case* tersebut, sehingga pelanggan bisa berhubungan dengan *use case* pesan kelas dan pesan kursi tanpa harus pesan tiket (misalnya pelanggan hanya ingin kelas VIP dan di kursi depan, jika tidak tersedia, maka pelanggan tidak jadi pesan tiket).

Baik inklusi maupun ekstensi semuanya bermaksud memperluas perilaku *use case* dasarnya. Untuk lebih jelasnya, Tabel 6.1 berikut ini dapat dijadikan patokan perbedaan antaran inklusi dan ekstensi (Pender, 2003).

Tabel 6.1. Perbedaan Antara Inklusi dan Ekstensi

Inklusi	Ekstensi
<i>Use case</i> terpanggil (<i>include use case</i>) selalu diperlukan oleh <i>use case</i> dasar	<i>Use case</i> ekstensi tidak selalu dibutuhkan oleh <i>use case</i> dasar
Yang memutuskan kapan dipanggilnya <i>use case include</i> adalah <i>use case</i> dasar.	Yang memutuskan kapan dipanggilnya <i>use case</i> ekstensi itu sendiri.
Panah hubungan dengan <i>use Case</i> dasar ke <i>use case include</i> .	Panah hubungan dari <i>use case</i> Ekstensi ke <i>use case</i> dasar

*image
not
available*

*image
not
available*

*image
not
available*

bentuk metodologi yang mereka namakan *Object-Oriented Software Engineering (OOSE)*. Tujuan dibentuknya metode ini adalah agar dihasilkan fokus yang baik pada pengembang dan tujuan utama tanpa terpengaruh oleh implementasi praktis.

Elemen *use case* terdiri dari:

1. Diagram *use case*, disertai dengan narasi dan skenario.
2. Aktor (*actor*), mendefinisikan entitas di luar sistem yang memakai sistem.
3. Asosiasi (*association*), mengindikasikan aktor mana yang berinteraksi dengan *use case* dalam suatu sistem.
4. <<*include*>> dan <<*extend*>>. Merupakan indikator yang menggambarkan jenis relasi dan interaksi antar-*use case*.
5. Generalisasi (*generalization*), menggambarkan hubungan turunan antara *use case* atau antaraktor.

Use case mengekspresikan apa yang *user* harapkan terhadap sistem. Narasi *use case* menjelaskan secara detail bagaimana *user* berinteraksi dengan sistem saat mengakses *use case*. Skenario memecah penjelasan narasi untuk menyediakan penjelasan detail terhadap segala kemungkinan yang terjadi pada *use case*, apa yang terjadi dan apa respon sistem.

*image
not
available*

*image
not
available*

*image
not
available*



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.



You have either reached a page that is unavailable for viewing or reached your viewing limit for this book.

*image
not
available*