

BAB II

TINJAUAN PUSTAKA

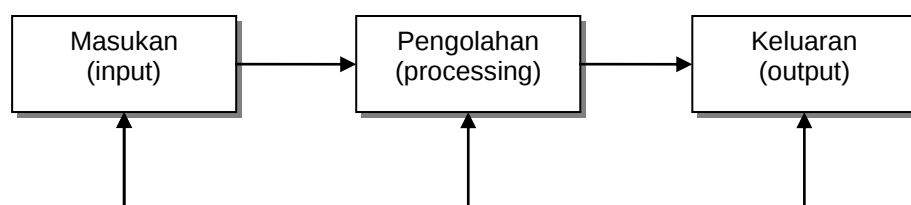
II.1. Konsep Dasar Sistem

Konsep dasar sistem akan menguraikan beberapa pengertian sistem, karakteristik sistem, pengertian dan komponen sistem informasi

II.1.1. Pengertian Sistem

Secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel-variabel yang saling terorganisasi, saling berinteraksi, dan saling bergantung sama lain. Sistem dapat didefinisikan sebagai sekelompok elemen yang terintegrasi dengan maksud yang sama untuk mencapai suatu tujuan. (Eko Nugroho, 2008 : 17)

Susunan suatu sistem pada dasarnya terdiri dari unsur-unsur seperti unit masukan (*input*), unit pengolahan (*processing*), serta unit keluaran (*output*). Input atau masukan masuk ke dalam sistem melalui unit input. Selanjutnya, input di proses oleh unit pemroses dan hasilnya ditampilkan ataupun dicetak keluar melalui unit output.



Gambar II.1. Model Sistem

(Sumber: Hanif Al Fatta, 2007 : 4)

II.1.2. Karakteristik Sistem

Untuk memahami atau mengembangkan suatu sistem, maka perlu membedakan unsur-unsur dari sistem yang membentuknya. Berikut adalah karakteristik sistem yang dapat membedakan suatu sistem dengan sistem yang lainnya :

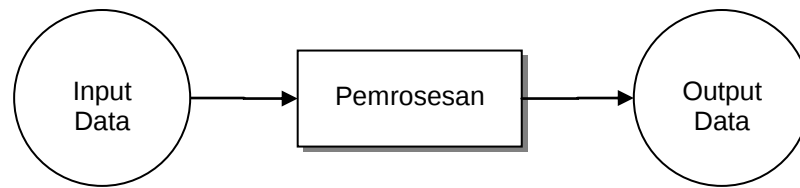
1. Batasan (*boundary*) : Penggambaran dari suatu elemen atau unsur mana yang termasuk di dalam sistem dan mana yang di luar sistem.
2. Lingkungan (*environment*) : Segala sesuatu di luar sistem, lingkungan yang menyediakan asumsi, kendala, dan input terhadap suatu sistem.
3. Masukan (*input*) : Sumber daya (data, bahan baku, peralatan, energi) dari lingkungan yang dikonsumsi dan dimanipulasi oleh suatu sistem.
4. Keluaran (*output*) : sumber daya atau produk (informasi, laporan, dokumen, tampilan layer computer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.
5. Komponen (*component*) : Kegiatan-kegiatan atau proses dalam suatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.
6. Penghubung (*interface*) : Tempat di mana komponen atau sistem dan lingkungannya bertemu atau berinteraksi.
7. Penyimpanan (*storage*) : Area yang dikuasai dan digunakan untuk penyimpanan sementara dan tetap dari informasi, energi, bahan baku, dan sebagainya. Penyimpanan merupakan suatu media penyangga di antara komponen tersebut bekerja dengan berbagai tingkatan yang ada dan

memungkinkan komponen yang berbeda dari berbagai data yang sama. (Hanif Al Fattah, 2007 : 5)

II.1.3. Pengertian Sistem Informasi

Untuk memahami pengertian sistem informasi, harus dilihat keterkaitan antara data dan informasi sebagai entitas penting pembentuk sistem informasi. Data merupakan nilai, keadaan, atau sifat yang berdiri sendiri lepas dari konteks apapun. Sementara informasi adalah data yang telah diolah menjadi sebuah bentuk yang berarti bagi penerimanya dan bermanfaat dalam pengambilan keputusan saat ini atau mendatang. Dari pengertian di atas kita dapat mendefinisikan sistem informasi sebagai integrasi antara orang, data, alat dan prosedur yang bekerja sama dalam mencapai suatu tujuan. Jadi sistem informasi adalah data yang telah diproses, atau data yang memiliki arti yang didalamnya terdapat elemen orang, alat, data dan prosedur atau cara. (Eko Nugroho, 2008 : 17)

Akhirnya Sistem Informasi Manajemen (SIM) dapat didefinisikan sebagai suatu alat untuk menyajikan informasi dengan cara sedemikian rupa sehingga bermanfaat bagi penerimanya (Kertahadi, 1995). Tujuannya adalah untuk menyajikan informasi guna pengambilan keputusan pada perencanaan, pemrakarsaan, pengorganisasian, pengendalian kegiatan operasi subsistem suatu perusahaan, dan menyajikan sinergi organisasi pada proses (Murdick dan Ross, 1993). Dengan demikian, sistem informasi berdasarkan konsep (*input, processing, output – IPO*) dapat dilihat pada gambar berikut :



Gambar II.2. Konsep sistem informasi

(Sumber: Hanif Al Fatta, 2007 : 9)

II.2. Pengertian Sistem Informasi Akuntansi

Sistem Informasi Akuntansi adalah sistem yang bertujuan untuk mengumpulkan dan memproses data serta melaporkan informasi yang berkaitan dengan transaksi keuangan. Setelah memproses transaksi dengan mencatat penjualan tersebut ke dalam jurnal penjualan, mengklasifikasikan transaksi dengan menggunakan kode rekening, dan memposting transaksi tersebut ke dalam jurnal. Kemudian, secara periodik Sistem Informasi Akuntansi akan menghasilkan output berupa laporan keuangan yang terdiri dari Neraca dan Laba Rugi . (Anastasia Diana – Lilis Setiawati, 2011 : 5)

II.3. Pengertian laporan Kas

Kas merupakan aset yang paling kritis artinya bagi perusahaan. Pada umumnya, aktivitas perusahaan dilakukan dengan dan untuk menghasilkan kas. Akan tetapi, agar perusahaan dapat *survive*, yang tak kalah pentingnya adalah menempatkan dan mengelola kas dengan benar. Kekurangan atau kelebihan kas bisa mendatangkan permasalahan. Operasi perusahaan dapat terhenti karena kekurangan kas untuk pembelian bahan baku. Gaji karyawan yang tidak dibayar dapat menyebabkan karyawan mogok kerja. Utang yang jatuh tempo tetapi tidak dibayar dapat menyebabkan masalah hukum.

Sebaliknya, kelebihan kas dapat juga sangat merugikan perusahaan. Kas yang menganggur (*idle*), selain menimbulkan resiko penggelapan atau kecurangan lainnya, juga menimbulkan kerugian penurunan nilai intrinsik. Kas yang menganggur juga membuat perusahaan kehilangan kesempatan berinvestasi pada bidang-bidang yang menguntungkan. Pengelolaan kas turut menentukan keberhasilan perusahaan. (Golrida Karyawati P, 20013 : 97)

II.3.1. Likuidasi

Likuiditas adalah kemampuan perusahaan dalam memenuhi kewajiban jangka pendeknya. Pengertian lain adalah kemampuan seseorang atau perusahaan untuk memenuhi kewajiban atau utang yang segera harus dibayar dengan harta lancarnya Menurut PP 25 tahun 1999 pasal 1.

Likuiditas diukur dengan rasio aktiva lancar dibagi dengan kewajiban lancar. Perusahaan yang memiliki likuiditas sehat paling tidak memiliki rasio lancar sebesar 100 persen. Ukuran likuiditas perusahaan yang lebih menggambarkan tingkat likuiditas perusahaan ditunjukkan dengan rasio kas (kas terhadap kewajiban lancar).

II.3.2. Metode Langsung

Metode langsung menganalisa sumber kas (arus kas masuk) dan penggunaan kas (arus kas keluar) dari nilai pendapatan operasi dan beban-beban sehubungan operasi yang tersaji dalam laporan laba/rugi. (Golrida Karyawati P, 20013 : 104)

II.4. Basis Data (Database)

Database merupakan sekumpulan data berbentuk tabel yang digunakan untuk menyimpan suatu informasi. Penggunaan database tidak bisa dipungkiri memang banyak sekali digunakan hampir disemua jenis aplikasi saat ini. Mulai dari aplikasi multimedia, browser web, aplikasi penjualan dan kepegawaian. (Wahana Komputer, 2010 : 150)

Pada era kemajuan teknologi seperti sekarang ini, nilai informasi sangatlah penting, terlebih bagi kemajuan perusahaan. Oleh karena itu penggunaan dan penguasaan database sangat penting. Dalam database ada sebutan-sebutan untuk satuan data yaitu:

1. Karakter, ini adalah satuan data terkecil. data terdiri atau susunan karakter yang pada akhirnya mewakili data yang memiliki arti dari sebuah fakta.
2. *Field*, adalah kumpulan dari karakter yang mewakili fakta tertentu misalnya seperti nama siswa, tanggal lahir, dan lain-lain. Dalam dunia perancangan database, *field* juga disebut atribut. Bila dipandang dari sudut pemrograman berorientasi objek maka name dan properti tipe. Properti name atau nama adalah properti dari *field* yang berisi *field* yang mewakili data sejenis yang disimpannya. Sedangkan properti tipe adalah properti yang mengatur tipe data dari data yang akan ditampungnya. Misalnya nama *field* nya adalah nama siswa maka tipe datanya adalah char, bila nama *field* nya adalah tanggal lahir maka tipe datanya adalah date.
3. *Record*, adalah kumpulan dari *field*. Pada *record* anda dapat menemukan banyak sekali informasi penting dengan cara mengombinasikan *field-field* yang ada.

4. Tabel, adalah sekumpulan dari *record-record* yang memiliki kesamaan entity dalam dunia nyata. Kumpulan dari tabel adalah database, wujud fisik sebuah database dalam komputer adalah sebuah file yang didalamnya terdapat berbagai tingkatan data yang telah disebutkan di atas.
5. File, adalah bentuk fisik dari penyimpanan data. File database berisi semua data yang telah disusun dan diorganisasikan sedemikian rupa sehingga memudahkan pemberian informasi. (Wahana Komputer, 2010 : 24)

II.5. ERD (*Entity Relationship Diagram*)

Pada dasarnya ERD(*Entity Relationship Diagram*) adalah sebuah diagram yang secara konseptual memetakan hubungan antar penyimpanan pada diagram DFD di atas. ERD ini digunakan untuk melakukan permodelan terhadap struktur data dan hubungannya. Penggunaannya ERD ini dilakukan untuk mengurangi tingkat kerumitan penyusunan sebuah database yang baik.

Entity dapat berarti sebuah obyek yang dapat dibedakan dengan obyek lainnya. Obyek tersebut dapat memiliki komponen-komponen data (atribut atau field) yang membuatnya dapat dibedakan dari obyek yang lain. Dalam dunia database *entity* memiliki atribut yang menjelaskan karakteristik dari entity tersebut. Ada dua macam atribut yang dikenal deskriptif. Hal ini berarti setiap entity memiliki himpunan yang diperlukan sebuah primary key untuk membedakan anggota-anggota dalam himpunan tersebut.

Atribut dapat memiliki sifat-sifat sebagai berikut:

1. *Atomic*, atomik adalah sifat dari atribut yang menggambarkan bahwa atribut tersebut berisi nilai yang spesifik dan tidak dapat dipecah lagi.

Contoh dari sifat atomik adalah field status dari tabel karyawan yang hanya berisi menikah atau single

2. *Multivalued*, sifat ini menandakan atribut ini bisa memiliki lebih dari satu nilai untuk tiap entity tertentu. Misalnya adalah field hobi, hobi dari tiap karyawan mungkin dan hampir pasti lebih dari satu. Misalnya karyawan A memiliki hobi membaca, nonton TV dan bersepeda.
3. *Composite*, atribut yang bersifat komposit adalah atribut yang nilainya adalah gabungan dari beberapa atribut yang bersifat atomik. Contohnya adalah atribut alamat yang dapat dipecah menjadi atribut atomik berupa alamat, kode pos, no telepon, dan kota.

Ada beberapa derajat relasi yang dapat terjadi, yaitu :

1. *One to one*, menggambarkan bahwa antara 1 anggota entity A hanya dapat berhubungan dengan 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-1.
2. *One to many*, menggambarkan bahwa 1 anggota entity A dapat memiliki hubungan dengan lebih dari 1 anggota entity B. Biasanya derajat relasi ini digambarkan dengan simbol 1-N.
3. *Many to many*, menggambarkan bahwa lebih dari satu anggota A dapat memiliki hubungan dengan lebih dari satu anggota entity B. Simbol yang digunakan adalah N-N. (Wahana Komputer: 2010 ; 30 - 31)

II.6. Normalisasi

Untuk merealisasikan database pada tabel-tabel maka diperlukan sebuah tahapan yang disebut normalisasi. Normalisasi data adalah proses di mana tabel-tabel pada database dites dalam hal kesalingtergantungan di antara field-field pada sebuah tabel. Misalnya jika pada sebuah tabel terdapat ketergantungan terhadap lebih dari satu field dalam tabel tersebut, maka tabel tersebut harus dipecah menjadi banyak tabel.

Pada proses normalisasi data, aturan yang dijadikan acuan adalah metode ketergantungan fungsional. Teorinya adalah bahwa tiap kolom dalam sebuah tabel selalu memiliki hubungan yang unik dengan sebuah kolom kunci. Misalnya pada sebuah tabel data_mahasiswa ada *field* nomor induk mahasiswa, data *field* nama mahasiswa serta *field* alamat. Maka ketergantungan fungsionalnya dapat dinyatakan sebagai berikut: *nim* -> *nm_mhs* dan *nmr_induk* -> *alamat*. Artinya *nm_mhs* memiliki ketergantungan fungsional terhadap *nim*. *Field* *nm_mhs* isinya juga ditentukan oleh *field* *nim*. Maksud dari semua itu adalah *nim* adalah *field* kunci yang menentukan karena tidak ada nomor induk yang sama pada satu universitas, jadi *field* *nmr_induk* dapat dijadikan patokan untuk mengisi *nm_mhs* dan *field* lainnya.

Ada beberapa langkah dalam normalisasi tabel, yaitu:

1. *Decomposition*, dekomposisi adalah proses mengubah bentuk tabel supaya memenuhi syarat tertentu sebagai tabel yang baik. Dekomposisi dapat dikatakan berhasil jika tabel yang dikenal dekomposisi bila digabungkan kembali dapat menjadi tabel awal sebelum di-dekomposisi. Dekomposisi

akan sering dilakukan dalam proses normalisasi untuk memenuhi syarat-syaratnya

2. Bentuk tidak normal, pada bentuk ini semua data yang ada pada tiap entity (diambil atributnya) masih ditampung dalam satu tabel besar. Data yang ada pada tabel ini masih ada yang redundansi dan ada juga yang kosong. Semuanya masih tidak tertata rapi.
3. Normal Form pertama (1st Normal Form), pada tahapan ini tabel di-dekomposisi dari tabel bentuk tidak normal yang kemudian dipisahkan menjadi tabel-tabel kecil yang memiliki kriteria tidak memiliki atribut yang bernilai ganda dan komposit. Semua atribut harus bersifat atomik.
4. Normal Form kedua (2nd Normal Form), pada tahapan ini tabel dianggap memenuhi normal kedua jika pada tabel tersebut semua atribut yang bukan kunci primer bergantung penuh terhadap kunci primer tabel tersebut .
5. Normal Form ketiga (3rd Normal Form), setiap atribut pada tabel selain kunci primer atau kunci utama harus bergantung penuh pada kunci utama. Bentuk normal ketiga biasanya digunakan bila masih ada tabel yang belum efisien. Biasanya penggunaan bentuk normal(normalisasi) hanya sampai pada bentuk ketiga, dan tabel yang dihasilkan telah memiliki kualitas untuk membentuk sebuah database yang dapat diandalkan. Semua tabel diatas juga telah memenuhi bentuk normal tahap ketiga. (Wahana Komputer, 2010 : 32)

II.7. Kamus Data

Kamus data (*data dictionary*) digunakan untuk memperjelas aliran data yang digambarkan pada DFD. Kamus data adalah kumpulan daftar elemen data yang mengalir pada sistem perangkat lunak sehingga masukan (*input*) dan keluaran (*output*) dapat dipahami secara umum (memiliki standar cara penulisan). Kamus data biasanya berisi :

1. Nama-nama dari data
2. Digunakan pada – merupakan proses-proses yang terkait data
3. Deskripsi – merupakan deskripsi data
4. informasi tambahan – seperti tipe data, nilai data, batas nilai data, dan komponen yang membentuk data.

Kamus data memiliki beberapa simbol untuk menjelaskan informasi tambahan sebagai berikut :

Tabel II.1 Simbol Kamus Data

Simbol	Keterangan
=	Disusun atau terdiri dari
+	Dan
[]	Baik...atau...
{ } ⁿ	n kali diulang / bernilai banyak
()	Data opsional
...	Batas komentar

(Sumber: Rosa A.S-M.Shalahuddin: 2011; 68)

Kamus data pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang

tidak dapat dipetakan pada table hasil perancangan basis data berarti hasil perancangan hasil data dengan perancangan dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

II.8. Pemrograman Visual Basic 2010

Visual Basic merupakan salah satu aplikasi pemrograman visual yang paling banyak digunakan dan cukup populer karena kemudahan penggunaannya. Selain itu, Visual Basic yang mengadopsi bahasa pemrograman basic juga memiliki segudang fitur dan fasilitas yang memudahkan para programmer mengembangkan sebuah aplikasi.

Visual Basic 2010 merupakan salah satu bagian dari produk pemrograman terbaru yang dikeluarkan oleh *microsoft* yaitu Microsoft Visual Studio 2010 sebagai produk lingkungan pengembangan terintegrasi atau IDE andalan yang dikeluarkan oleh *microsoft*, seperti Visual Basic, Visual C++, Visual Web Developer, Visual C# dan Visual F#. Semua IDE pemrograman tersebut sudah mendukung penuh implementasi .Net Framework terbaru yaitu .Net Framework 4.0 yang merupakan pengembangan dari .Net Framework 3.5. Adapun database standar yang disertakan adalah Microsoft SQL Server 2008 Express. Microsoft Visual Studio 2010 menambahkan perbaikan – perbaikan fitur dan fitur baru yang lebih lengkap dibandingkan versi sebelumnya yaitu Microsoft Visual Studio 2008 (Wahana Komputer, 2010 : 2)

II.9. Database SQL Server 2008

Microsoft SQL Server 2008 Express Edition merupakan edisi lite dari Microsoft SQL Server 2008. Fitur – fitur yang terdapat didalamnya bisa dikatakan hampir sama dengan versi sebelumnya. Hal yang membedakan hanyalah fungsi-fungsi enterprise dan ketiadaan database manager yang bersifat independen atau berupa aplikasi mandiri. Bagi sebagian user dan developer aplikasi memang agak sedikit menyusahkan apabila hendak melakukan maintenance database secara langsung, tetapi bagi pengguna Visual Basic 2010 hal tersebut tidak menjadi masalah, sebab Visual Basic 2010 sudah ditanamkan fitur untuk mengelola database yang berasal dari Microsoft SQL Server 2008 Express Editio. (Wahana Komputer, 2010 : 152)

II.10. *Unified Modelling Language (UML)*

II.10.1. Pengertian *UML*

Unified Modeling Language (UML) adalah “bahasa” yang telah menjadi standar dalam industri untuk visualisasi, merancang dan mendokumentasikan sistem piranti lunak. *UML* menawarkan sebuah standar untuk merancang model sebuah sistem.

UML merupakan salah satu alat bantu yang sangat handal dalam bidang pengembangan sistem berorientasi objek karena *UML* menyediakan bahasa pemodelan visual yang memungkinkan pengembangan sistem yang membuat *blue print* atas visinya dalam bentuk yang baku. (Yuni Sugiarti, 2013:34)

UML diaplikasikan untuk maksud tertentu, biasanya antara lain untuk :

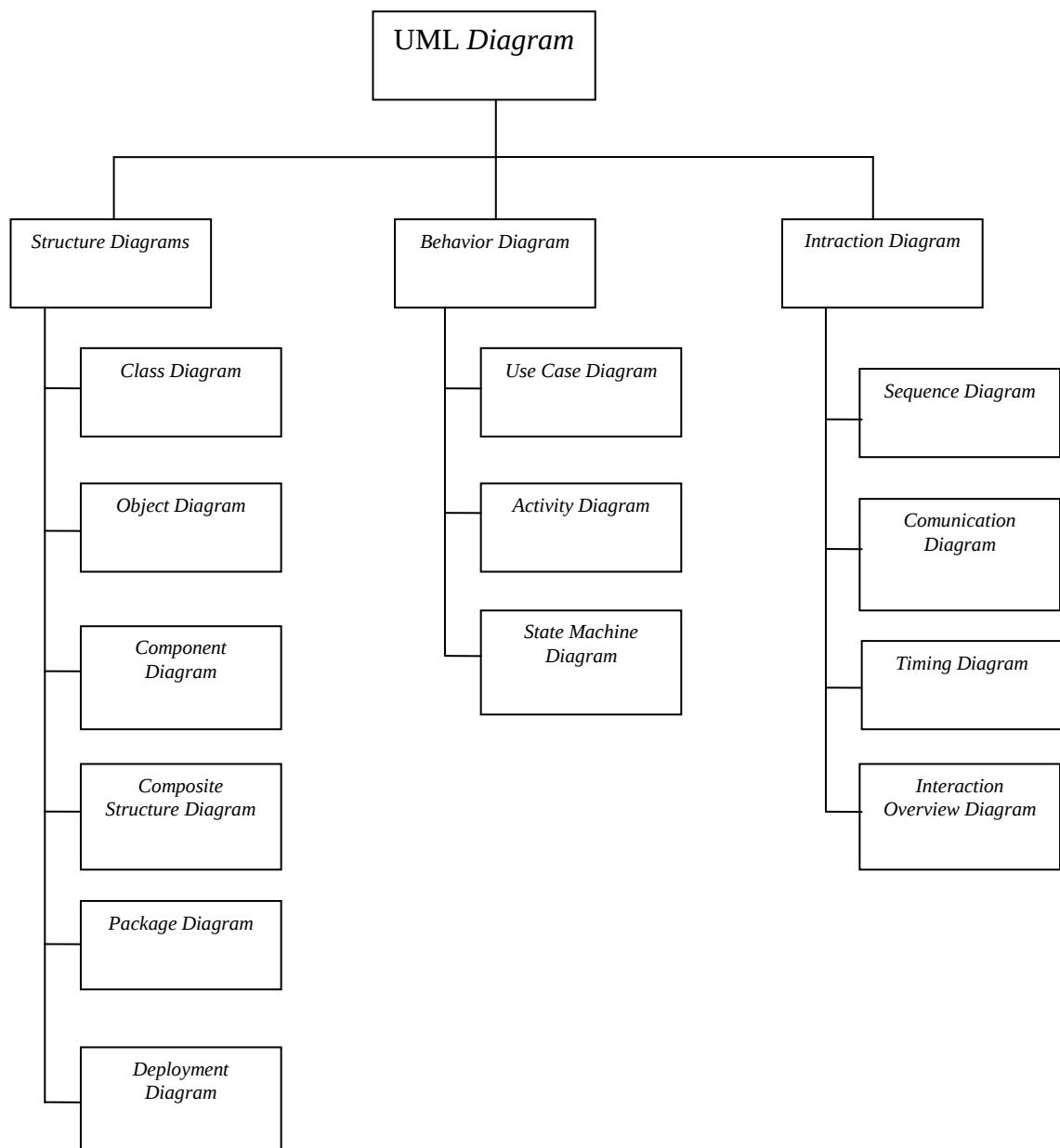
1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.
4. Mendokumentasikan sistem yang ada, proses-proses dan organisasinya.

UML telah diaplikasikan dalam investasi perbankan, lembaga kesehatan, departemen pertahanan, sistem terdistribusi, sistem pendukung alat kerja, retail, sales, dan supplier.

Blok pembangunan utama *UML* adalah diagram. Beberapa diagram ada yang rinci (jenis *timing diagram*) dan lainnya ada yang bersifat umum (misalnya diagram kelas). Para pengembang sistem berorientasikan objek menggunakan bahasa model untuk menggambarkan, membangun dan mendokumentasikan sistem yang mereka rancang. *UML* memungkinkan para anggota team untuk bekerja sama dalam mengaplikasikan beragam sistem. Intinya, *UML* merupakan alat komunikasi yang konsisten dalam mensupport para pengembang sistem saat ini. Sebagai perancang sistem mau tidak mau pasti menjumpai *UML*, baik kita sendiri yang membuat sekedar membaca diagram *UML* buatan orang lain (Prabowo Pudji Widodo, Herlawati; 2011 : 6-7).

II.10.2. Diagram UML

Pembagian kategori dan macam-macam diagram tersebut dapat dilihat pada gambar II.3. dibawah ini :



Gambar II.3. Diagram UML
 (Sumber : Rosa A.S - M.Shalahuddin: 2011; 121)

Berikut ini penjelasan singkat dari pembagian kategori tersebut :

1. *Structure Diagrams*

Yaitu kumpulan *diagram* yang digunakan untuk menggambarkan suatu struktur statis dari sistem yang dimodelkan.

2. *Behavior Diagrams*

Yaitu kumpulan diagram yang digunakan untuk menggambarkan kelakuan sistem atau rangkaian perubahan yang terjadi pada sebuah sistem.

3. *Interaction Diagrams*

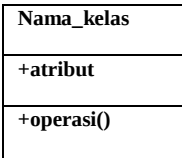
Yaitu kumpulan diagram yang digunakan untuk menggambarkan interaksi sistem dengan sistem lain maupun interaksi antar subsistem pada suatu sistem.

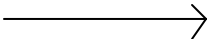
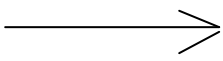
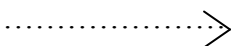
II.10.3. *Class Diagram*

Diagram kelas atau *Class Diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem.

Berikut adalah simbol-simbol yang ada pada diagram kelas:

Tabel II.2. Komponen *Class Diagram*

Simbol	Deskripsi
Kelas/Operasi 	Kelas pada struktur system
Antarmuka/interface  Nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek

Asosiasi/association 	Relasi antar kelas dengan makna umum,asosiasi biasanya juga disertai dengan multiplicity
Asosiasi berarah/directed association 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan multiplicity
Generalisasi 	Relasi antar kelas dengan makna generalisasi-spesialisasi(umum khusus)
Kebergantungan/dependency 	Relasi antar kelas dengan makna kebergantungan antar kelas

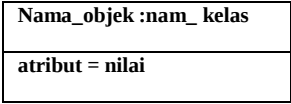
(Sumber: Yuni Sugiarti : 2013; 57-59)

II.10.4. *Object Diagram*

Object Diagram menggambarkan struktur sistem dari penamaan object dan jalannya object dalam sistem. Pada diagram object harus dipastikan semua kelas yang sudah didefinisikan pada diagram kelas harus dipakai objeknya, karena jika tidak pendefinisian kelas itu tidak dapat dipertanggung jawabkan.

Berikut adalah simbol-simbol yang ada pada diagram objek:

Tabel II.3. Komponen *Object Diagram*

Simbol	Deskripsi
Kelas 	Objek dari kelas yang berjalan saat dijalankan
Link 	Relasi antar objek

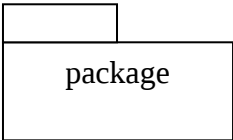
(Sumber: Rosa A.S-M.Shalahuddin: 2011; 124)

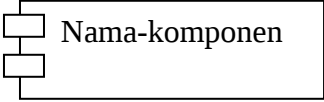
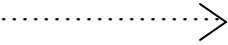
II.10.5. *Component Diagram*

Diagram komponen atau *component diagram* dibuat untuk menunjukkan organisasi dan ketergantungan diantara kumpulan komponen dalam sebuah sistem. diagram komponen focus pada komponen sistem yang dibutuhkan dan ada di dalam sistem.

Berikut adalah simbol-simbol yang ada pada diagram komponen:

Tabel II.4. Komponen *Component Diagram*

Simbol	Deskripsi
Package 	<i>Package</i> merupakan sebuah bungkus dari satu atau lebih komponen

Komponen  Nama-komponen	Komponen system
Kebergantungan/dependency 	Kebergantungan antar komponen, arah panah mengarah pada komponen yang dipakai
Antarmuka/interface  Nama_interface	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek agar tidak mengakses langsung komponen
Link 	Relasi antar komponen

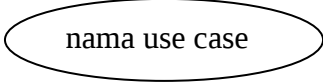
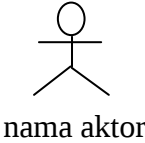
(Sumber: Rosa A.S-M.Shalahuddin: 2011; 125 - 126)

II.10.6. Use Case Diagram

Use case atau diagram use case merupakan pemodelan untuk melakukan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem informasi yang akan dibuat. Syarat penamaan pada use case adalah nama didefinisikan sesimpel mungkin dan dapat dipahami.

Berikut adalah simbol-simbol yang ada pada diagram *use case*:

Tabel II.5. Komponen *Use Case Diagram*

Simbol	Deskripsi
<i>Use case</i> 	Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
Aktor / <i>actor</i> 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi malapung symbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang
Asosiasi / <i>association</i> 	Komunikasi antar aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor
Ekstensi / <i>extend</i>	Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan itu

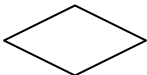
(Sumber: Yuni Sugiarti : 2013; 41-42)

II.10.7. Activity Diagram

Diagram aktivitas atau *activity* diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem.

Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel II.6. Komponen Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktifitas sistem, sebuah diagram aktivitas memiliki sebuah status awal
Aktivitas 	Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja
Percabangan / <i>decision</i> 	Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu
Penggabungan / <i>join</i> 	Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir

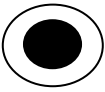
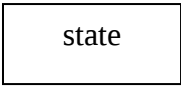
(Sumber: Rosa A.S-M.Shalahuddin: 2011; 134-135)

II.10.8. State Machine Diagram

State machine diagram atau diagram mesin statu digunakan untuk menggambarkan perubahan status atau transisi status dari sebuah mesin atau sistem.

Berikut komponen-komponen dasar yang ada dalam *state machine diagram*:

Tabel II.7. Komponen *state machine diagram*

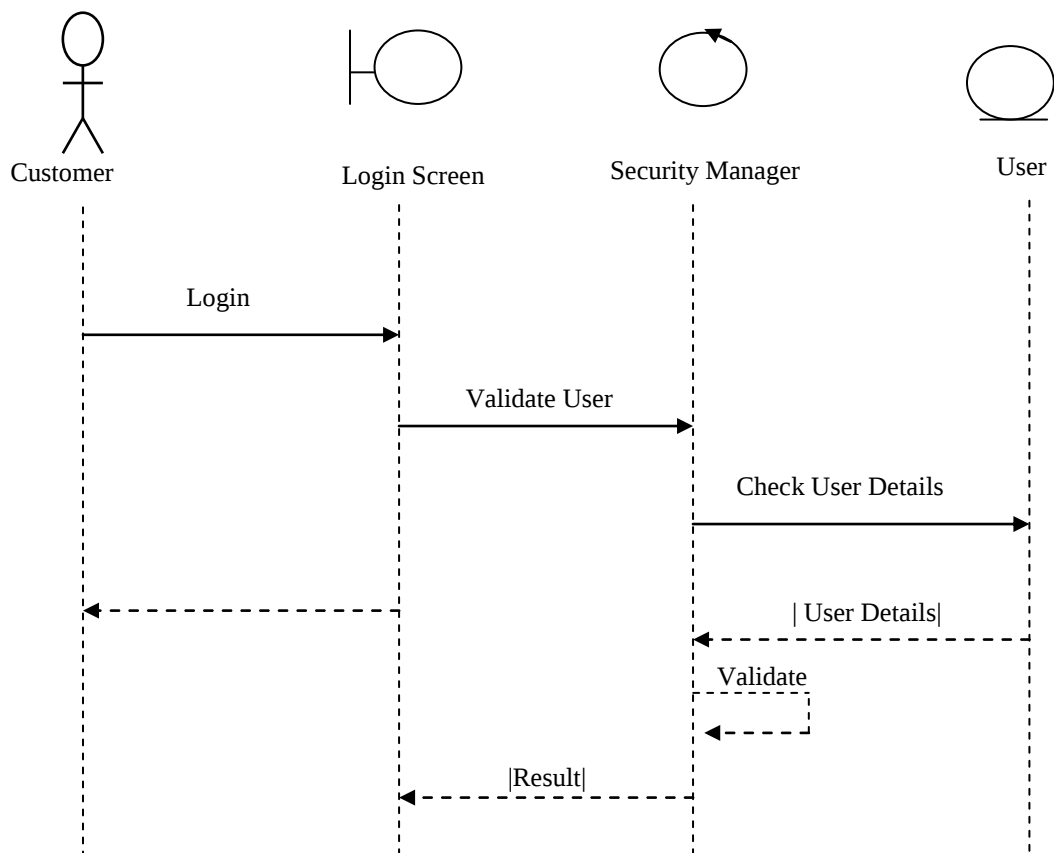
Simbol	Deskripsi
Start (Initial State) 	Start atau <i>initial state</i> adalah <i>state</i> atau keadaan awal pada saat sistem mulai hidup
End (Final State) 	<i>End</i> atau <i>final state</i> adalah <i>state</i> keadaan akhir dari daur hidup suatu system
Event 	<i>Event</i> adalah kegiatan yang menyebabkan berubahnya status mesin
State 	<i>State</i> atau status adalah keadaan sistem pada waktu tertentu. <i>State</i> dapat berubah jika ada <i>event</i> tertentu yang memicu perubahan tersebut

(Sumber: Rosa A.S-M.Shalahuddin: 2011; 136-137)

II.10.9. Sequence Diagram

Menurut Douglas (2004 : 174) menyebutkan ada tiga diagram primer UML dalam memodelkan scenario interaksi, yaitu diagram urutan (*sequence diagram*), diagram waktu (*timing diagram*) dan diagram komunikasi (*communication diagram*).

Menurut Pilone (2005 : 174) menyatakan bahwa diagram yang paling banyak dipakai adalah diagram urutan. Gambar II.7. memperlihatkan contoh diagram urutan dengan notasi-notasinya yang akan dijelaskan nantinya (Prabowo Pudji Widodo, Herlawati; 2011 : 174-175)



Gambar II.4. Diagram Urutan

(Sumber : Prabowo Pudjo Widodo, Herlawati; 2011:175)