

BAB II

TINJAUAN PUSTAKA

II.1 Sistem

1. Defenisi Sistem

Definisi sistem berkembang sesuai dengan konteks dimana pengertian sistem itu digunakan. Berikut beberapa definisi sistem secara umum :

1. Kumpulan dari bagian-bagian yang bekerja sama untuk mencapai tujuan yang sama.
2. Sekumpulan objek-objek yang saling berelasi dan berinteraksi serta hubungan antar objek bisa dilihat sebagai satu kesatuan yang dirancang untuk mencapai satu tujuan.

Dengan demikian, secara sederhana sistem dapat diartikan sebagai suatu kumpulan atau himpunan dari unsur atau variabel – variabel yang terorganisasi, saling berinteraksi dan saling bergantung satu sama lain. (*Hanif Al-Fatta;2007:3*).

Sistem merupakan kumpulan elemen yang saling berkaitan yang bertanggung jawab memproses masukan (*input*) sehingga menghasilkan keluaran (*output*). Suatu sistem didalam suatu organisasi yang memperemukan kebutuhan pengolahan transaksi harian, mendukung operasi, bersifat manajerial, dan merupakan kegiatan strategi dari suatu organsisasi, serta menyediakan laporan – laporan yang diperlukan oleh pihak luar.

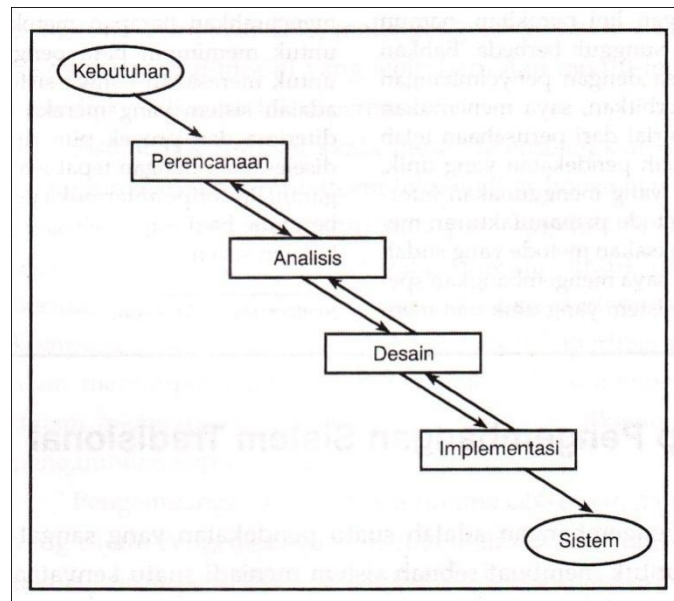
Berdasarkan dukungan kepada pemakainya, sistem informasi dibagi menjadi:

1. Sistem Pemrosesan Transaksi (*Transaction Processing System* atau TPS)

2. Sistem Informasi Manajemen (*Management Information System* atau MIS)
3. Sistem Otomasi Perkantoran (*Office Automation System/OAS*)
4. Sistem Pendukung Keputusan (*Decision Support System* atau DSS)
5. Sistem Informasi Eksekutif (*Executive Information System* atau EIS)
6. Sistem Pendukung Kelompok (*Group Support System* atau GSS)
7. Sistem Pendukung Cerdas (*Intelligent Support System* atau ISS).

(Kusrini, M.Kom ; 2007 : 11)

Pengembangan adalah suatu pendekatan yang sangat rapi dan berurutan untuk membuat sebuah sistem menjadi suatu kenyataan. Diperlukan suatu metodologi untuk menyediakan suatu struktur pengembangan sistem. Ada banyak Siklus Hidup Pengembangan Sistem (SDLC) "tradisional" untuk sistem informasi, termasuk DSS. Setiap perangkat rancang-bangun perangkat lunak dengan bantuan komputer (*CASE*) telah mengadopsi suatu variasi. Untuk berbagai hal sulit, masing-masing organisasi yang mengembangkan sebuah sistem dapat menciptakan variasi *in-house* agar sesuai dengan kebutuhan spesifik mereka. Masing-masing metodologi menekankan langkah-langkah berbeda dalam cara yang berbeda. Akan tetapi, semua SDLCs yang secara intuitif dan praktis harus mengikuti petunjuk dan proses tertentu. (Efraim Turban ; 2005 : 401-402).



Gambar II.1 SDLC

(Efraim Turban ; 2005 : 402)

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 lainnya :

1. Batasan (*Boundary*).

Penggambaran dari suatu sistem 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. **Masukkan (*input*).**

Sumber daya (data, bahan baku, peralatan, energi) lingkungan yang di konsumsi dan dimanipulasi oleh suatu sistem.

4. **Keluaran (*Output*).**

Sumber daya atau produk (informasi, laporan, dokumen, tampilan layar komputer, barang jadi) yang disediakan untuk lingkungan sistem oleh kegiatan dalam suatu sistem.

5. **Komponen (*Component*).**

Kegiatan-kegiatan atau proses dalam sesuatu sistem yang mentransformasikan input menjadi bentuk setengah jadi (*output*). Komponen ini bisa merupakan subsistem dari sebuah sistem.

6. **Penghubung (*Interface*).**

Tempat dimana komponen atau sistem dan lingkungan bertemu atau berinteraksi.

7. **Penyimpanan (*Storage*).**

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- Fatta ; 2007:5-6*).

II.2. Sistem Pendukung Keputusan (*Decision Support System*)

1. Defenisi Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) adalah sistem berbasis komputer interaktif, yang membantu para pengambil keputusan untuk menggunakan data dan berbagai model untuk memecahkan masalah-masalah tidak terstruktur (Turban dkk, 2005).

Menurut Kusriani (2007) mendefinisikan sistem pendukung keputusan merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan dan manipulasi data. Sistem itu digunakan untuk membantu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi tidak terstruktur, dimana tak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat.

Senada dengan pakar lainnya Turban dkk (2005) mendefinisikan sistem pendukung keputusan sebagai sekumpulan prosedur berbasis model untuk data pemrosesan dan penilaian guna membantu para manajer mengambil keputusan. Dari beberapa definisi diatas dapat dikatakan bahwa sistem pendukung keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu dalam pengambilan keputusan berkaitan dengan persoalan yang bersifat semiterstruktur.

Aplikasi DSS menggunakan CBIS (*Computer Based Information Systems*) yang fleksibel, interaktif, dan dapat diadaptasi, yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur. Aplikasi DSS menggunakan data, memberikan antarmuka pengguna yang mudah, dan dapat menggabungkan pemikiran pengambil keputusan. DSS lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis

dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas. DSS tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambil keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia.

Dari beberapa definisi diatas dapat dikatakan bahwa sistem pendukung keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu dalam pengambilan keputusan berkaitan dengan persoalan yang bersifat semiterstruktur.

2. Karakteristik dan Kapabilitas Sistem Pendukung Keputusan

Sistem pendukung keputusan dirancang secara khusus untuk mendukung seseorang yang harus mengambil keputusan-keputusan tertentu. Dibawah ini merupakan karakteristik dan kapabilitas sistem pendukung keputusan (Turban, dkk, 2005) :

- a. Dukungan untuk pengambilan keputusan, terutama pada situasi semiterstruktur dan tak terstruktur, dengan menyertakan penilaian manusia dan informasi terkomputerisasi.
- b. Dukungan untuk semua level manajerial, dari eksekutif puncak sampai manajer lini.
- c. Dukungan untuk individu dan kelompok. Masalah yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasi yang berbeda atau bahkan dari organisasi lain.

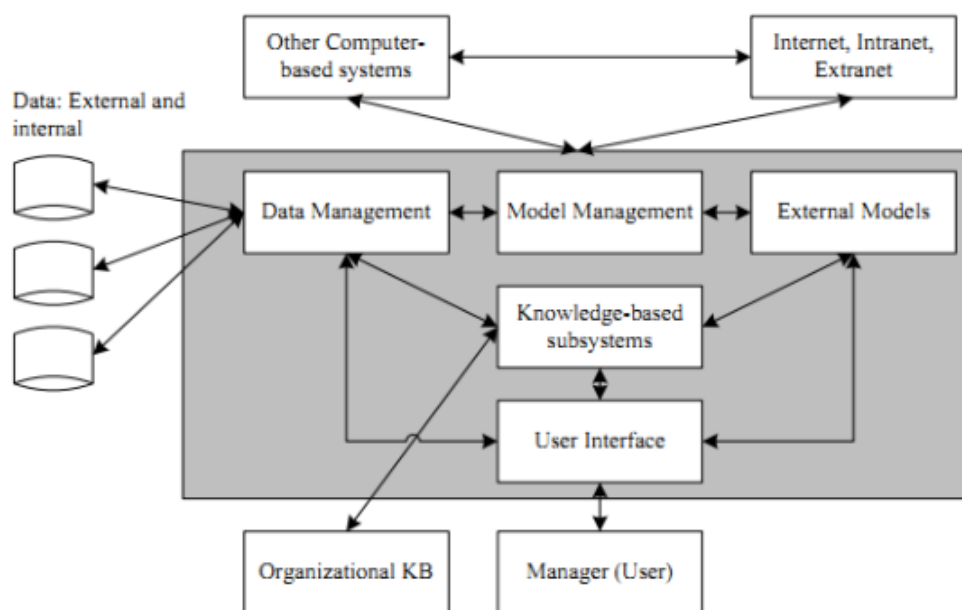
- d. Dukungan untuk keputusan independen dan atau sekuensial. Keputusan dapat dibuat satu kali, beberapa kali atau berulang (dalam *interval* yang sama).
- e. Dukungan di semua fase proses pengambilan keputusan: intelegensi, desain, pilihan, dan implementasi.
- f. Dukungan diberbagai proses dan gaya pengambilan keputusan.
- g. Adaptivitas sepanjang waktu. Pengambilan keputusan seharusnya reaktif, dapat menghadapi perubahan kondisi secara cepat dan dapat mengadaptasikan *Decision Support System* (DSS) untuk memenuhi perubahan tersebut.
- h. Pengguna merasa seperti di rumah. Ramah-pengguna, kapabilitas grafis yang sangat kuat dan antarmuka manusia-mesin interaktif dengan satu bahasa alami dapat sangat meningkatkan keefektifan DSS.
- i. Peningkatan terhadap keefektifan pengambilan keputusan (akurasi, timeliness, kualitas) ketimbang pada efisiensinya (biaya pengambilan keputusan).
- j. Kontrol penuh oleh pengambil keputusan terhadap semua langkah proses pengambilan keputusan dalam memecahkan suatu masalah.
- k. Pengguna akhir dapat mengembangkan dan memodifikasi sendiri sistem sederhana. Sistem yang lebih besar dapat dibangun dengan bantuan ahli sistem informasi. Perangkat lunak On-Line Analytical Processing (OLAP) dalam kaitannya dengan data warehouse membolehkan pengguna untuk membangun DSS yang cukup besar dan kompleks.

1. Biasanya model-model digunakan untuk menganalisis situasi pengambilan keputusan.

3. Komponen Sistem Pendukung Keputusan

Menurut Hermawan (2005) Sistem Pendukung Keputusan terdiri atas tiga komponen penting, yaitu subsistem manajemen data, subsistem manajemen model dan antarmuka pengguna.

Berikut adalah gambaran tiga komponen utama dalam sistem pendukung keputusan (Hermawan, 2005) :



Gambar II.2 Skematik Sistem Pendukung Keputusan

Subsistem sistem pendukung keputusan pada Gambar II.1 dijelaskan sebagai berikut:

- a. Subsistem manajemen data berisi data yang relevan untuk suatu situasi dan dikelola oleh Database Management System (DBMS). Subsistem ini

dapat di interkoneksi dengan data warehouse perusahaan yang relevan untuk pengambilan keputusan.

- b. Subsistem manajemen model merupakan paket perangkat lunak yang menyimpan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lainnya yang memberikan kemampuan analitik yang tepat. Perangkat lunak ini sering disebut Model Based Management System (MBMS) dan tepat di implementasikan pada sistem pengembangan web untuk berjalan pada server aplikasi.
- c. Subsistem antarmuka pengguna merupakan dukungan komunikasi antara sistem dengan pengguna. Web browser menjadi salah satu antarmuka yang menampilkan dalam bentuk grafis dan interaktif dengan pengguna.

4. Tujuan Sistem Pendukung Keputusan

Menurut Kusri (2007), tujuan sistem pendukung keputusan adalah:

- a. Membantu manajer dalam pengambilan keputusan atas masalah semiterstruktur.
- b. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk mengganti fungsi manajer.
- c. Meningkatkan efektivitas keputusan yang diambil lebih dari perbaikan efisiensinya.
- d. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputasi secara cepat dengan biaya rendah.

- e. Peningkatan produktifitas. Membangun satu kelompok pengambil keputusan, terutama para pakar, bisa sangat mahal. Pendukung terkomputerisasi bisa mengurangi ukuran kelompok dan memungkinkan para anggotanya untuk berada di berbagai lokasi yang berbeda-beda (menghemat biaya perjalanan). Selain itu, produktivitas staf pendukung (misalnya analis keuangan dan hukum) bisa ditingkatkan. Produktivitas juga bisa ditingkatkan menggunakan peralatan optimalisasi yang menentukan cara terbaik untuk menjalankan sebuah bisnis.
- f. Dukungan kualitas. Komputer bisa meningkatkan kualitas keputusan yang dibuat. Sebagai contoh, semakin banyak data yang diakses, makin banyak juga alternatif yang bisa dievaluasi. Analisis resiko bisa dilakukan dengan cepat dan pandangan dari para pakar (beberapa dari mereka berada di lokasi yang jauh) bisa dikumpulkan dengan cepat dan dengan biaya yang lebih rendah. Keahlian bahkan bisa diambil langsung dari sebuah sistem komputer melalui metode kecerdasan tiruan. Dengan komputer, para pengambil keputusan bisa melakukan simulasi yang kompleks, memeriksa banyak skenario yang memungkinkan, dan menilai berbagai pengaruh secara cepat dan ekonomis. Semua kapabilitas tersebut mengarah kepada keputusan yang lebih baik.
- g. Berdaya saing. Manajemen dan pemberdayaan sumber daya perusahaan. Tekanan persaingan menyebabkan tugas pengambilan keputusan menjadi sulit, persaingan didasarkan tidak hanya pada harga, tetapi juga pada kualitas, kecepatan, kustomasi produk, dan dukungan pelanggan.

Organisasi harus mampu secara sering dan cepat mengubah mode operasi, merekayasa ulang proses dan struktur, memberdayakan karyawan, serta berinovasi. Teknologi pengambilan keputusan bisa menciptakan pemberdayaan yang signifikan dengan cara memperbolehkan seseorang untuk membuat keputusan yang baik secara cepat, bahkan jika mereka memiliki pengetahuan yang kurang.

- h. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.

II.3. Profile Matching

Menurut Kusri (2007) metode *profile matching* atau pencocokan *profil* adalah metode yang sering digunakan sebagai mekanisme dalam pengambilan keputusan dengan mengasumsikan bahwa terdapat tingkat variabel prediktor yang ideal yang harus dipenuhi oleh subyek yang diteliti, bukannya tingkat minimal yang harus dipenuhi atau dilewati. Dalam proses *profile matching* secara garis besar merupakan proses membandingkan antara nilai data aktual dari suatu *profile* yang akan dinilai dengan nilai profil yang diharapkan, sehingga dapat diketahui perbedaan kompetensinya (disebut juga gap), semakin kecil gap yang dihasilkan maka bobot nilainya semakin besar (Handojo, 2011).

Berikut adalah beberapa tahapan dan perumusan perhitungan dengan metode

profile matching (Kusri, 2007) :

1. Pembobotan

Pada tahap ini, akan ditentukan bobot nilai masing-masing aspek dengan menggunakan bobot nilai yang telah ditentukan bagi masing-masing aspek itu sendiri. Adapun inputan dari proses pembobotan ini adalah selisih dari profil karyawan dan profil jabatan.

Dalam penentuan peringkat pada aspek kapasitas intelektual, sikap kerja dan perilaku untuk jabatan yang sama pada setiap gap, diberikan bobot nilai sesuai dengan tabel berikut :

Tabel II.1 Bobot Nilai Gap

no.	selisih gap	bobot nilai	keterangan
1	0	5	Kompetensi sesuai dengan kebutuhan
2	1	4.5	kompetensi induvidu kelebihan 1 tingkat/level
3	-1	4	kompetensi inuvidu kurang 1 tingkat /level
4	2	3.5	kompetensi induvidu kelebihan 2 tingkat/level
5	-2	3	kompetensi inuvidu kurang -2 tingkat /level
6	3	2.5	kompetensi induvidu kelebihan 3 tingkat/level
7	-3	2	kompetensi inuvidu kurang -3 tingkat /level
8	4	1.5	kompetensi induvidu kelebihan 4 tingkat/level
9	-4	1	kompetensi inuvidu kurang -4 tingkat /level

2. Pengelompokan Core dan Secondary Factor

Setelah menentukan bobot nilai gap kriteria yang dibutuhkan, kemudian tiap kriteria dikelompokan lagi menjadi dua kelompok yaitu core factor dan secondary factor.

a. Core Factor (Faktor Utama)

Core factor merupakan aspek (kompetensi) yang paling menonjol/paling dibutuhkan oleh suatu jabatan yang diperkirakan dapat menghasilkan kinerja optimal. Untuk menghitung core factor digunakan rumus :

$$NCI = \frac{\sum NC}{\sum IC}$$

Keterangan :

NCI = Nilai rata-rata core factor aspek kapasitas intelektual

NC = Jumlah total nilai core factor aspek kapasitas intelektual

IC = Jumlah item core factor

b. Secondary factor (Faktor Pendukung)

Secondary factor adalah item-item selain aspek yang ada pada core factor.

Untuk menghitung secondary factor digunakan rumus :

$$NSI = \frac{\sum NS}{\sum IS}$$

Keterangan :

NSI = Nilai rata-rata secondary factor aspek kapasitas intelektual

NS = Jumlah total nilai secondary factor aspek kapasitas intelektual

IS = Jumlah item secondary factor

Rumus diatas adalah rumus untuk menghitung core factor dan secondary factor dari aspek kapasitas intelektual. Rumus diatas juga digunakan untuk menghitung core factor dan secondary factor dari aspek sikap kerja dan perilaku.

3. Perhitungan Nilai Total

Dari perhitungan core factor dan secondary factor dari tiap-tiap aspek, kemudian dihitung nilai total dari tiap-tiap aspek yang diperkirakan berpengaruh pada kinerja tiap-tiap profile. Untuk menghitung nilai total dari masing-masing aspek, digunakan rumus :

$$N = (X) \% NCI + (X) \% NSI$$

Keterangan :

N = Nilai Total Tiap Aspek

NCI = Nilai Core Factor

NSI = Nilai Secondary Factor

(X)% = Nilai Persentase

4. Perangkingan

Hasil akhir dari proses profile matching adalah rangking dari kandidat yang diajukan untuk mengisi suatu jabatan/posisi tertentu. Penentuan mengacu rangking pada hasil perhitungan yang ditunjukkan pada rumus dibawah ini :

$$\text{Rangking} = 20\% NKI + 30\% NSK + 50\% NP$$

Keterangan :

NKI = Nilai Kapasitas Intelektual

NSK = Nilai Sikap Kerja

NP = Nilai Perilaku

II.4. Kredit

Dalam kehidupan sehari-hari, kata kredit bukan merupakan perkataan yang bagi masyarakat kita. Perkataan kredit tidak saja dikenal oleh masyarakat di kota-kota besar, tetapi sampai di desa-desa pun kata kredit tersebut sudah sangat populer. Istilah kredit berasal dari bahasa Yunani (*Credere*) yang berarti kepercayaan (*truth* atau *faith*). Oleh karena itu dasar dari kredit ialah kepercayaan. Seseorang atau suatu badan yang memberikan kredit (kreditur) percaya bahwa penerima kredit (debitur) di masa mendatang akan sanggup memenuhi segala sesuatu yang telah dijanjikan. Apa yang telah dijanjikan itu dapat berupa barang, uang atau jasa.

Dengan demikian prestasi dan kontraprestasi dapat berbentuk sebagai berikut :

- a. Barang terhadap barang
- b. Barang terhadap uang
- c. Barang terhadap jasa
- d. Jasa terhadap uang
- e. Jasa terhadap barang
- f. Uang terhadap uang
- g. Uang terhadap barang
- h. Uang terhadap jasa

Dengan akan diterimanya kontraprestasi pada masa yang akan datang, maka jelas tergambar bahwa kredit dalam arti ekonomi adalah penundaan pembayaran dari prestasi yang diberikan sekarang, baik dalam bentuk barang,

uang, maupun jasa. Disini terlihat pula bahwa faktor waktu merupakan faktor utama yang memisahkan prestasi dan kontraprestasi. Dengan demikian kredit itu dapat pula berarti bahwa pihak kesatu memberikan prestasi baik berupa barang, uang, atau jasa kepada pihak lain, sedangkan kontraprestasi akan diterima kemudian (dalam jangka waktu tertentu).

II.4.1. Tujuan dan Fungsi kredit

1. Tujuan Kredit

Keuntungan merupakan dari pemberian kredit yang menjelma dalam bentuk bunga yang diterima. Dan karna pancasila adalah sebagai dasar dan falsafah negarta kita, maka tujuan kredit tidak semata-mata mencari keuntungan, melainkan disesuaikan dengan tujuan negara yaitu untuk mencapai masyarakat adil dan makmur berdasarkan pancasila. Dengan demikian maka tujuan kredit yang diberikan oleh suatu pihak bank, khususnya bank pemerintah yang akan mengembangkan tugas sebagai *agent of development* adalah untuk :

- a. Turut menyukseskan program pemerintah di bidang ekonomi dan pembangunan.
- b. Meningkatkan aktivitas perusahaan agar dapat menjalankan fungsinya guna menjamin terpenuhnya kebutuhan masyarakat.
- c. Memperoleh laba agar kelangsungan hidup perusahaan terjamin, dan dapat memperluas usahanya.

Dari tujuan tersebut, tersimpul adanya kepentingan seimbang antaranya :

1. Kepentingan pemerintah.
2. Kepentingan masyarakat (rakyat) , dan
3. Kepentingan pemilik modal (pengusaha). (*Thomas Suyatno – 1988:hal 25*)

2. Fungsi Kredit

Dalam kehidupan perekonomian yang modern, bank memegang peranan yang sangat penting. Oleh karena itu organisasi-organisasi bank selalu diikutsertakan dalam menentukan kebijakan di bidang moneter, pengawasan devisa, pencatatan efek-efek dan lain-lain. Hal ini antara lain disebabkan usaha pokok bank adalah memberikan kredit, dan kredit yang diberikan oleh bank mempunyai pengaruh yang sangat luas dalam segala bidang kehidupan, khususnya di bidang ekonomi.

Fungsi kredit perbankan dalam kehidupan perekonomian dan perdagangan antara lain sebagai berikut :

1. Kredit pada hakikatnya dapat meningkatkan daya guna.
2. Kredit dapat meningkatkan peredaran dan lalu lintas uang.
3. Kredit dapat pula meningkatkan daya guna dan peredaran barang
4. Kredit sebagai salah satu alat stabilitas ekonomi
5. Kredit dapat meningkatkan kegairahan berusaha
6. Kredit dapat meningkatkan pemerataan pendapatan
7. Kredit sebagai alat untuk meningkatkan hubungan internasional.

(Thomas Suyatno – 1988 ; 16-17)

3. Kelayakan Penerimaan Kredit

Menurut *Mangkepe (2004)* kelayakan penerimaan kredit seorang nasabah mutlak harus dilakukan untuk mengetahui kemampuan seorang nasabah dalam memenuhi kewajibannya membayar tagihan apabila permohonan kredit nasabah tersebut disetujui. Pada dasarnya penilaian kelayakan menerima kredit itu terbagi beberapa definisi sebagai berikut :

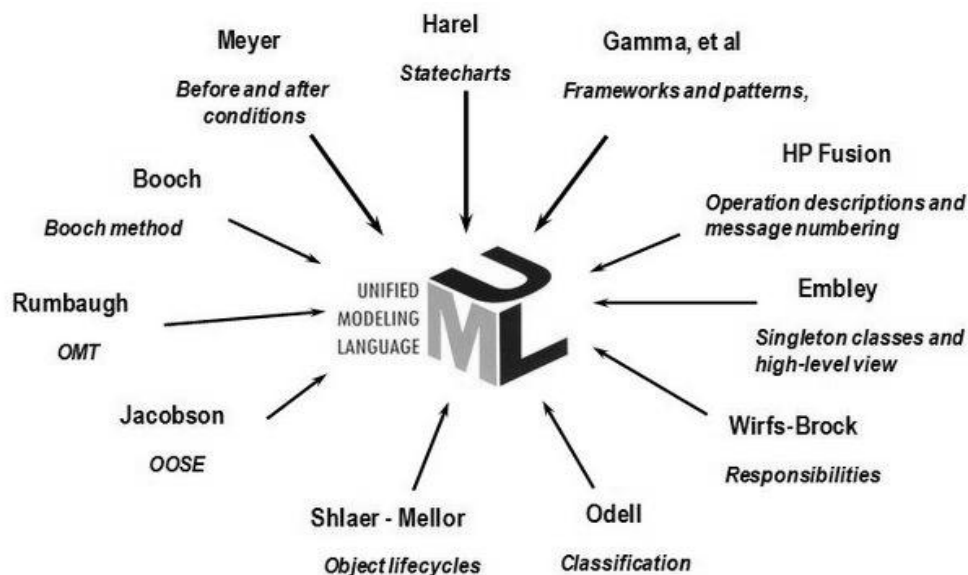
1. Kelayakan menerima kredit ini merupakan evaluasi terhadap nasabah secara keseluruhan.
2. Kelayakan menerima kredit ini pada dasarnya merupakan suatu proses mengestimasi dan menentukan nilai kelayakan seorang nasabah menerima kredit.
3. Kelayakan menerima kredit ini membandingkan realita nyata dengan standar yang ada pada perusahaan.
4. Kelayakan menerima kredit ini dilakukan oleh manajer. Kelayakan menerima kredit ini menentukan kebijakan selanjutnya.

II.5. UML (*Unified Modelling Language*)

UML (*Unified Modelling Language*) adalah salah satu alat yang sangat handal di dunia pengembangan system yang berorientasi objek. Hal ini disebabkan karena UML menyediakan bahasa pemodelan visual yang memungkinkan bagi pengembangan sistem untuk membuat cetak biru atas mereka dalam bentuk yang baku, mudah dimengerti serta dilengkapi dengan mekanisme

yang efektif untuk berbagi (*sharing*) dan mengkomunikasikan rancangan mereka dengan yang lain.

UML merupakan kesatuan dari bahasa pemodelan yang dikembangkan oleh Booch, *Objek Modeling Technique (OMT)* dan *Object Oriented Software Engineering (OOSE)*. Dengan UML, metode Booch, OMT dan OOSE digabungkan dengan membuang elemen-elemen baru yang belum ada pada metode terdahulu sehingga UML lebih ekspresif dan seragam daripada metode lainnya. Gambar berikut adalah unsure-unsur yang membentuk UML.



Gambar II.3. Unsur – Unsur Pembentuk UML

(Sumber : Munawar; 2005 : 17-18)

Sejarah UML sendiri cukup panjang, Sampai era tahun 1990 seperti kita ketahui puluhan metodologi pemodelan berorientasi objek telah bermunculan di dunia. Dimulai pada bulan Oktober 1994 Booch, Rumbaugh dan Jacobson, yang merupakan tiga tokoh yang boleh dikatakan metodologinya banyak digunakan

mempelopori usaha untuk penyatuan metodologi pendesainan berorientasi objek. Pada tahun 1995 direlease draft pertama dari UML (versi 0.8).

Sejak tahun 1996 pengembangan tersebut dikoordinasikan oleh Object Management Group (OMG – <http://www.omg.org>). Tahun 1997 UML versi 1.1 muncul, dan saat ini versi terbaru adalah versi 1.5 yang dirilis bulan Maret 2003. Booch, Rumbaugh dan Jacobson menyusun tiga buku serial tentang UML pada tahun 1999. Sejak saat itulah UML telah menjelma menjadi standar bahasa pemodelan untuk aplikasi berorientasi objek. Jadi kesimpulan menurut pendapat para ahli diatas bahwa *Unified Modelling Language (UML)* adalah suatu bahasa atau alat yang sangat berpengaruh pada pembuatan sebuah perancangan dan dokumentasi perangkat lunak yang berorientasi objek.

II.5.1 Konsep Dasar UML

Konsep dasar UML dari berbagai dokumen dan buku UML. Sebenarnya konsepsi dasar UML bisa kita rangkumkan dalam gambar di bawah ini.

Tabel II.2 Tabel Konsep UML

Major Area	View	Diagrams	Main Concepts
Structural	static view	class diagram	class, association, generalization, dependency, realization, interface
	use case view	use case diagram	use case, actor, association, extend, include, usecase generalization
	implementation view deployment	component diagram	component, interface, dependency
	View		incation
Dynamic	state maching view	statchart diagram	state, event, transition, action
	activity view	activity diagram	state, activity, completion transition, lork, join

	Interaction view	sequence diagram	interaction, object, message, activation
		collaboration diagram	collaboration, interaction, collaboration, role, message
Model management	model management view	class diagram	package, subsystem, model
Extensibility	All	All	constraint, stereotype, tagged values

(Sumber : *Sri Dharwiyanti and Romi Satria Wahono, Pengantar Unified Modeling Language (UML), 2003 : Hal 3*)

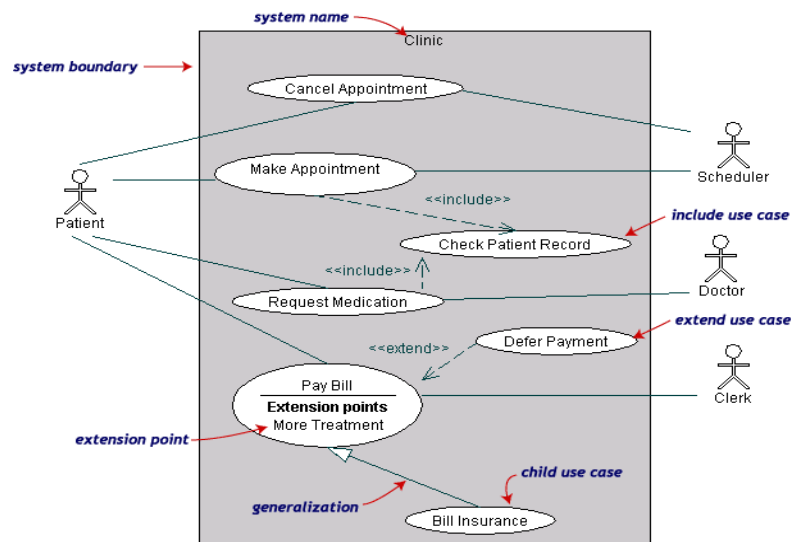
II.5.2. Use Case Diagram

Use case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Yang ditekankan adalah “apa” yang diperbuat sistem, dan bukan “bagaimana”. Sebuah use case merepresentasikan sebuah interaksi antara aktor dengan sistem. Use case merupakan sebuah pekerjaan tertentu, misalnya login ke sistem, meng-create sebuah daftar belanja, dan sebagainya. Seorang/sebuah aktor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu. Use case diagram dapat sangat membantu bila kita sedang menyusun requirement sebuah sistem, mengkomunikasikan rancangan dengan klien, dan merancang test case untuk semua feature yang ada pada sistem. Sebuah use case dapat meng-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 common. Sebuah 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.

(Sumber : Sri Dharwiyanti and Romi Satria Wahono, *Pengantar Unified Modeling Language (UML)*, 2003 : Hal 4).



Gambar II.4 Contoh Use Case Diagram

(Sumber : Sri Dharwiyanti and Romi Satria Wahono, *Pengantar Unified Modeling Language (UML)*, 2003 : Hal 5).

II.5.3. Class Diagram

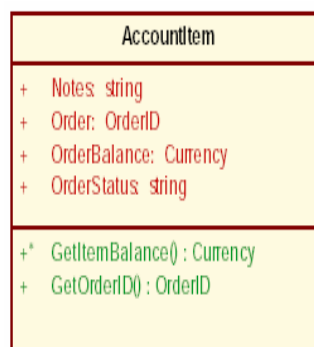
Class adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. Class menggambarkan keadaan (atribut/properti) suatu bentuk sistem, sekaligus menawarkan layanan untuk memanipulasi keadaan tersebut.

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 (*stereotype*)
2. Atribut
3. Metoda

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



Gambar II.5. Class Diagram

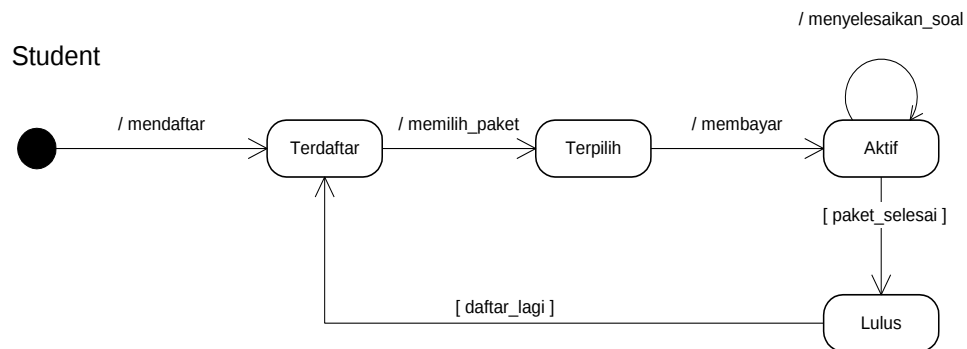
(Sumber : Sri Dharwiyanti and Romi Satria Wahono, *Pengantar Unified*

Modeling Language (UML), 2003 : Hal 15).

II.5.4. Statechart Diagram

Statechart diagram menggambarkan transisi dan perubahan keadaan (dari satu state ke state lainnya) suatu objek pada sistem sebagai akibat dari stimuli

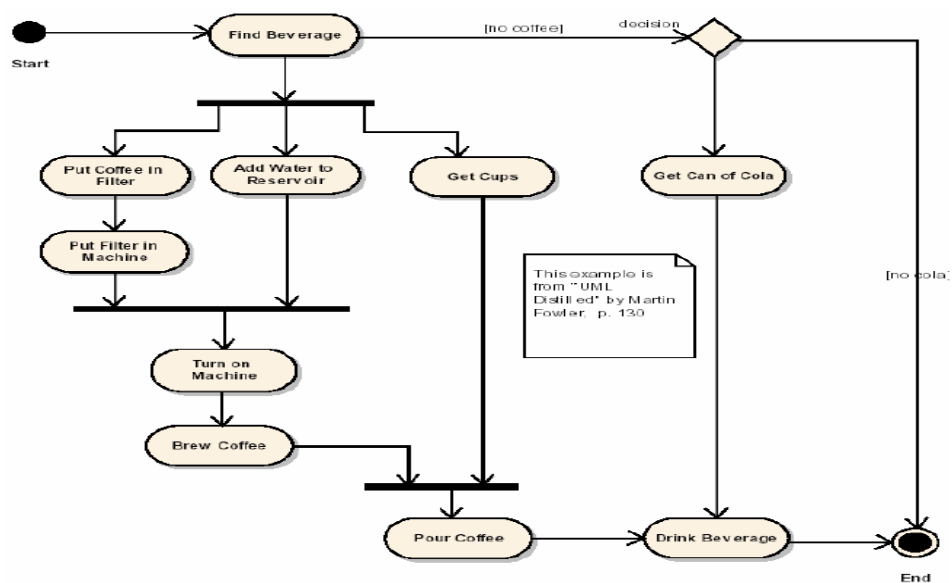
yang diterima. Pada umumnya statechart diagram menggambarkan class tertentu (satu class dapat memiliki lebih dari satu statechart diagram).



Gambar II.6. Contoh Statechart Diagram

II.5.5. Activity Diagram

Activity diagrams menggambarkan berbagai alir aktivitas 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.

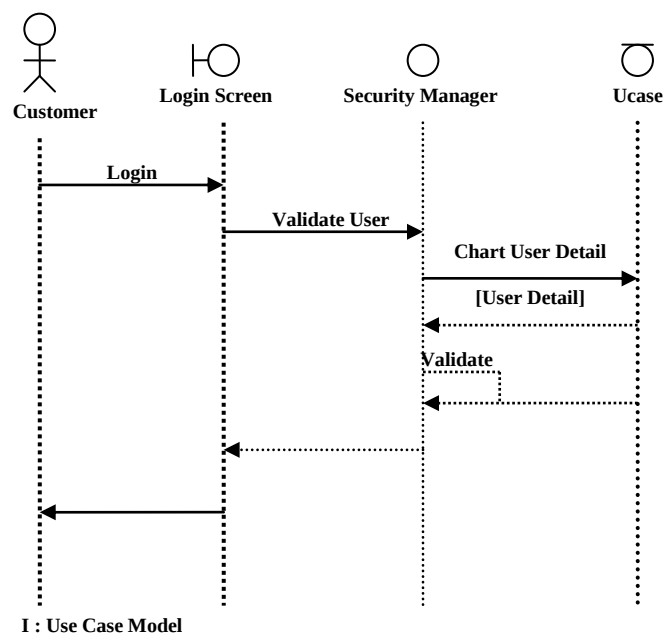


Gambar II.7. Contoh Activity Diagram – Tanpa Swimlane

II.5.6. Sequence Diagram

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 aktivitas tersebut, proses dan perubahan apa saja yang terjadi secara internal dan output apa yang dihasilkan.

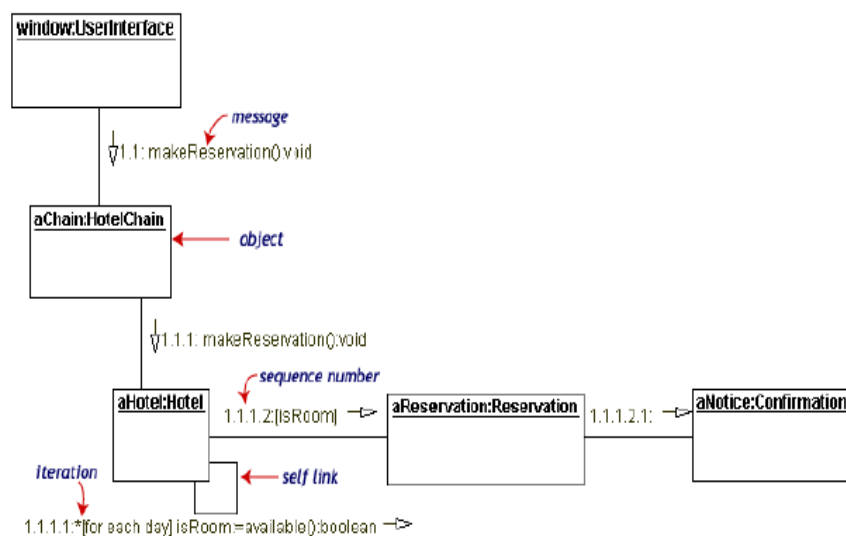
Contoh *Sequence Diagram* :



Gambar II.8. Contoh Sequence Diagram

II.5.7. Collaboration Diagram

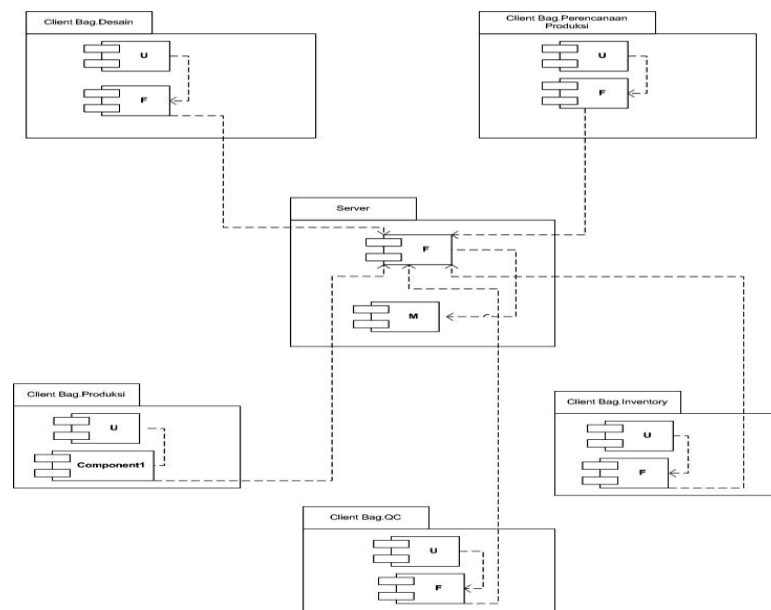
Collaboration diagram juga menggambarkan interaksi antar objek seperti sequence diagram, tetapi lebih menekankan pada peran masing-masing objek dan bukan pada waktu penyampaian message. Setiap message memiliki sequence number, di mana message dari level tertinggi memiliki nomor 1. Messages dari level yang sama memiliki prefiks yang sama.



Gambar II.9. Contoh Collaboration Diagram

II.5.8. Component Diagram

Komponen piranti lunak adalah modul berisi code, baik berisi source code maupun binary code, baik library maupun executable, baik yang muncul pada compile time, link time, maupun run time. Umumnya komponen terbentuk dari beberapa class dan/atau package, tapi dapat juga dari komponen-komponen yang lebih kecil. Komponen dapat juga berupa interface, yaitu kumpulan layanan yang disediakan sebuah komponen untuk komponen lain.



Gambar II.10. Contoh Component Diagram

Sumber : Sri Dharwiyanti and Romi Satria Wahono, *Pengantar Unified Modeling Language (UML)*, 2003

II.6. Database

Basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai objek, orang dan lain-lain. Data dinyatakan dengan nilai (angka deretan karakter atau simbol).

Basis data dapat didefinisikan dalam berbagai sudut pandang, seperti berikut :

1. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.

2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik. (Kusrini ; 2007 : 2)

1. DBMS (*Database Management System*)

DBMS (Database Management System) adalah perangkat lunak yang berfungsi untuk mengelola database. Mulai dari membuat database itu sendiri, sampai dengan proses yang berlaku dalam database tersebut, baik berupa entry, edit, hapus, query terhadap data, membuat laporan dan lain sebagainya secara efektif dan efisien. Salah satu jenis DBMS yang sangat terkenal saat ini adalah Relational DBMS (RDBMS). RDBMS merepresentasikan data dalam bentuk tabel-tabel yang saling berhubungan. Sebuah tabel disusun dalam bentuk baris (Record) dan kolom (Field).

Ada tiga kelompok perintah yang digunakan dalam mengelola dan mengorganisasikan data dalam RDBMS, yaitu :

1. Data Definition Language (DDL)

DDL merupakan perintah yang digunakan oleh seorang database administrator untuk mendefinisikan struktur database, baik membuat table baru, menentukan struktur penyimpanan tabel, model relasi antar tabel, validasi data.

2. Data Manipulation Language (DML)

DML adalah perintah-perintah yang digunakan untuk memanipulasi dan mengambil data pada suatu database. Manipulasi yang dapat dilakukan terhadap data adalah :

- a. Penambahan data
- b. Penyisipan data
- c. Penghapusan data
- d. Pengubahan data.

3. Data Control Language (DCL)

Bagian ini berkenaan dengan cara mengendalikan data, seperti siapa yang bisa melihat isi data, bagaimana data bisa digunakan banyak pengguna. Lebih mengarah pada sekuritas data. (Yuhefizard, 2008:3-4)

2. MySQL

MySQL merupakan database server yang dikembangkan oleh sebuah perusahaan bernama MySQL AB. Perusahaan yang beralokasi di Swedia ini memiliki hak resmi untuk mengembangkan dan mengelola sistem, memberikan dukungan penjualan dan layanan, serta mempekerjakan orang-orang yang memberikan kontribusi pada MySQL. Sebagaimana diungkapkan oleh pengembangan resminya, MySQL dilafalkan dengan ucapan “ My Ess Que Ell “, bukan “ my sequel” atau yang lainnya. Meskipun demikian, pihak MySQL AB juga tidak begitu mempermasalahkan bagaimana Anda akan mengucapkannya.

Secara garis besar, disini akan ditekankan lagi bahwa :

1. MySQL merupakan suatu *Database Managenent System* (DBMS).
2. MySQL adalah suatu *Relational Databse Management System* (RDMS).
3. Perangkat lunak MySQL didistribusikan secara open source.
4. Database server MySQL sangat cepat, dapat diperoleh dan mudah digunakan.
5. Database server MySQL bekerja dalam lingkungan client/server.
6. Dukungan terhadap perangkat lunak MySQL tersebar luas dan mudah ditemukan. (*Sumber : Didik Dwi Prasetyo; 1-2*).

MySQL adalah Sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual di bawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Tidak sama dengan proyek-proyek seperti Apache, dimana perangkat lunak dikembangkan oleh komunitas umum, dan hak cipta untuk kode sumber dimiliki oleh penulisnya masing-masing, MySQL dimiliki dan disponsori oleh sebuah perusahaan komersial Swedia MySQL AB, dimana memegang hak cipta hampir atas semua kode sumbernya. Kedua orang Swedia dan satu orang Finlandia yang mendirikan MySQL AB adalah *David Axmark, Allan Larsson, dan Michael "Monty" Widenius*. MySQL adalah sebuah implementasi dari sistem manajemen basis data relasional (RDBMS) yang didistribusikan secara gratis di bawah lisensi GPL (General Public License).

Setiap pengguna dapat secara bebas menggunakan MySQL, namun dengan batasan perangkat lunak tersebut tidak boleh dijadikan produk turunan yang bersifat komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam basis data yang telah ada sebelumnya ; SQL (Structured Query Language).

SQL adalah sebuah konsep pengoperasian basis data, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Kehandalan suatu sistem basis data (DBMS) dapat diketahui dari cara kerja pengoptimasi-nya dalam melakukan proses perintah-perintah SQL yang dibuat oleh pengguna maupun program-program aplikasi yang memanfaatkannya. Sebagai peladen basis data, MySQL mendukung operasi basis data transaksional maupun operasi basis data non-transaksional.

Pada modus operasi non-transaksional, MySQL dapat dikatakan unggul dalam hal unjuk kerja dibandingkan perangkat lunak peladen basis data kompetitor lainnya. Namun demikian pada modus non-transaksional tidak ada jaminan atas reliabilitas terhadap data yang tersimpan, karenanya modus non-transaksional hanya cocok untuk jenis aplikasi yang tidak membutuhkan reliabilitas data seperti aplikasi blogging berbasis web (Wordpress), CMS, dan sejenisnya. Untuk kebutuhan sistem yang ditujukan untuk bisnis sangat disarankan untuk menggunakan modus basis data transaksional, hanya saja sebagai konsekuensinya unjuk kerja MySQL pada modus transaksional tidak secepat unjuk kerja pada modus non-transaksional.

MySQL memiliki beberapa keistimewaan, antara lain :

1. Portabilitas.

MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac Os X Server, Solaris, Amiga, dan masih banyak lagi.

2. Perangkat lunak sumber terbuka.

MySQL didistribusikan sebagai perangkat lunak sumber terbuka, di bawah lisensi GPL sehingga dapat digunakan secara gratis.

3. Multi-user.

MySQL dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. Performance tuning.

MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. Ragam tipe data.

MySQL memiliki ragam tipe data yang sangat kaya, seperti signed / unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. Perintah dan Fungsi.

7. MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam perintah (query).

8. Keamanan.

MySQL memiliki beberapa lapisan keamanan seperti level subnet mask, nama host, dan izin akses user dengan sistem perizinan yang mendetail serta sandi terenkripsi.

9. Skalabilitas dan Pembatasan.

MySQL mampu menangani basis data dalam skala besar, dengan jumlah rekaman (records) lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

10. Konektivitas.

MySQL dapat melakukan koneksi dengan klien menggunakan protokol TCP/IP, UNIX soket (UNIX), atau Named Pipes (NT).

11. Lokalisasi.

MySQL dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari dua puluh bahasa. Meski pun demikian, bahasa Indonesia belum termasuk di dalamnya.

12. Antar Muka.

MySQL memiliki antar muka (interface) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming Interface).

13. Klien dan Peralatan.

MySQL dilengkapi dengan berbagai peralatan (tool) yang dapat digunakan untuk administrasi basis data, dan pada setiap peralatan yang ada disertakan petunjuk online.

14. Struktur tabel.

MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER TABLE, dibandingkan basis data lainnya. (*Sumber Adityo Pratama ; 2013 : 1-3*)

3. Kamus Data

Karena DBMS menyimpan kumpulan beberapa item data yang terpisah yang dapat digunakan pemakai pada beberapa aplikasi secara bersama-sama adalah penting beberapa mekanisme digunakan untuk menyediakan informasi mengenai beberapa item data bersangkutan.

Menurut Zulkifly Amsyah (2005:382), kamus data adalah suatu file yang terpisah yang menyimpan informasi seperti :

1. Nama setiap item/jenis/kolom data.
2. Struktur data untuk tiap item
3. Program yang menyediakan tiap item
4. Tingkat keamanan untuk tiap item.

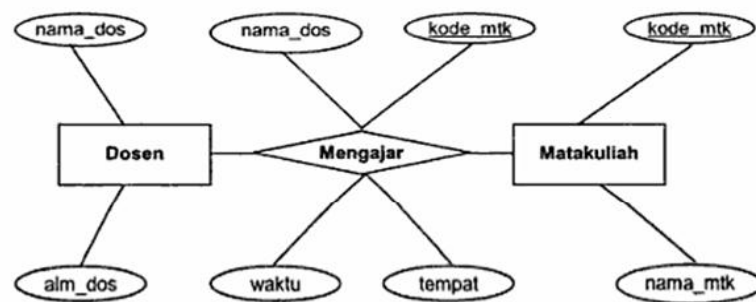
Kamus data berguna khusus bagi perlindungan timbulnya kelebihan data. Kamus data juga mengelola daftar password yang mengawasi akses ke sistem atau computer. (*Sumber : Zulkifly Amsyah ; 2005:382*)

4. ERD (*Entity Relationship Diagram*)

ERD digunakan untuk menggambarkan secara sistematis hubungan antar entity-entity yang ada dalam suatu sistem database menggunakan simbol-simbol sehingga mudah dipahami (Yuhefizard, 2008:17). Simbol yang digunakan adalah:

1. Persegi panjang, berfungsi untuk menyatakan suatu entity.
2. Elips, berfungsi untuk menyatakan attribute. Jika diberi garis bawah menandakan bahwa attribute tersebut merupakan attribute/field kunci.
3. Belah ketupat, menyatakan jenis relasi.
4. Garis, penghubung antara relasi dengan entity dan antara entity dengan attribute.

Contohnya hubungan antara entity dosen dengan entity matakuliah, sebagai berikut ini :



Gambar II.11. Contoh ERD

Sumber : Yuhefizard, 2008:17

5. Model Database Relasi (*Relationship Database Model*)

Pada relasional database dapat didefinisikan hubungan antar table yaitu sebagai berikut :

1. Satu Ke Satu (One To One).

Merupakan hubungan antar tabel dimana suatu tabel hanya memiliki suatu data pada tabel yang direferensikan.

2. Satu Ke Banyak (One To Many).

Merupakan hubungan antar tabel dimana suatu tabel memiliki lebih dari satu data pada tabel yang direferensikan.

3. Banyak Ke Banyak (Many To Many).

Merupakan hubungan antar tabel dimana suatu tabel memiliki lebih dari satu data yang direferensikan pada masing-masing tabel.

II.7. Normalisasi

Normalisasi merupakan cara pendekatan dalam membangun desain logika basis data relasional yang tidak secara langsung berkaitan dengan model data, tetapi dengan menerapkan sejumlah aturan dan kriteria standart untuk menghasilkan struktur tabel yang normal. Pada dasarnya desain logika basis data relasional dapat menggunakan prinsip normalisasi maupun transformasi dari model E-R ke bentuk fisik.

Dalam perspektif normalisasi sebuah database dikatakan baik jika semua tabel yang membentuk basis data sudah berada dalam keadaan normal. Suatu tabel dikatakan normal apabila :

1. Ada dekomposisi/penguraian tabel, maka dekomposisinya dijamin aman (lossless-join decomposition).
2. Terpeliharanya ketergantungan functional pada saat perubahan data (dependency preservation).
3. Tidak melanggar Boyce Code Normal Form (BCNF), jika tidak bisa minimal tidak melanggar bentuk normal ketiga.

Kondisi-kondisi yang diujikan pada proses normalisasi adalah sebagai berikut :

1. Menambah data.
2. Mengedit.
3. Menghapus.
4. Membaca.

(Kusrini, 2007:40)

1. Bentuk- Bentuk Normalisasi

Menurut *Kusrini (2007: 41)*, bentuk-bentuk normalisasi adalah sebagai berikut ini :

1. Bentuk tidak normal.

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan mengikuti format tertentu, dapat saja tidak lengkap dan terduplikasi.

2. Bentuk normal tahap pertama (1st normal form).

Sebuah tabel dapat dikatakan 1NF jika:

- a. Tidak ada baris yang duplikat dalam tabel tersebut.
- b. Masing-masing cell bernilai tunggal.

3. Bentuk normal tahap kedua (2nd normal form).

Bentuk normal kedua (2NF) terpenuhi jika pada sebuah tabel semua attribute yang tidak termasuk dalam primary key memiliki ketergantungan fungsional pada primary key secara utuh. Sebuah tabel tidak memenuhi 2NF, jika ketergantungannya hanya bersifat parsial atau tergantung pada sebagian primary key.

4. Bentuk normal tahap ketiga (3rd normal form).

Sebuah tabel dikatakan memenuhi bentuk normal ketiga (3NF), jika untuk setiap ketergantungan fungsional dengan notasi $X \rightarrow A$, dimana A mewakili semua atribut tunggal didalam tabel yang tidak ada didalam X, maka :

- a. X haruslah superkey pada tabel tersebut.
- b. Atau A merupakan bagian dari primary key pada tabel tersebut.

5. Bentuk normal tahap keempat dan kelima.

Bentuk normal keempat berkaitan dengan sifat ketergantungan banyak nilai (multivalued dependency) pada suatu tabel yang merupakan pengembangan dari ketergantungan fungsional. Adapun bentuk normal kelima merupakan nama lain dari Project Join Normal Form (PJNF).

6. Boyce code normal form (BCNF).

Suatu tabel dikatakan BNCF jika memenuhi 1NF dan relasi harus bergantung fungsi pada attribut superkey. (Sumber ; Kusrini, 2007:41)

2. PHP

Menurut *Eko Priyo Utomo, ST (hal 45)*, PHP merupakan bahasa *server-side* yang cukup handal, yang akan disatukan dengan HTML dan berada di server. Artinya, sintaks dan perintah diberikan akan sepenuhnya dijalankan di *server* sebelum dikirim ke komputer klien. Salah satu kelebihan PHP adalah kemampuan untuk dapat melakukan koneksi dengan berbagai *database*, seperti MySQL, PostgreSQL, Access sehingga sangat cocok jika digunakan untuk aplikasi *e-commerce* seperti *shopping online*.

PHP (*PHP: Hypertext Preprocessor*) adalah sebuah HTML-embedded scripting language, yaitu scripting language yang ‘ditempelkan’ dalam dokumen HTML, seperti halnya JavaScript atau VBScript. Tujuannya kurang lebih juga sama, yaitu untuk menciptakan halaman web yang interaktif dan dinamis. Lalu apa bedanya PHP dengan JavaScript atau VBScript ? Perbedaannya adalah tempat dimana script tersebut dieksekusi. JavaScript dan VBScript merupakan client-side scripting language yang akan dieksekusi di sisi klien (browser), sedangkan PHP adalah server-side scripting language yang akan dieksekusi di dalam web server ketika script-nya dipanggil. Begini untuk lebih jelasnya, Pada client-side scripting, ketika klien meminta sebuah dokumen yang mengandung script, VBScript atau JavaScript, script tersebut akan di download dan dieksekusi

di dalam browser yang bersangkutan. Sedangkan pada server-side scripting, dokumen yang diminta tetap berada di server, dijalankan di server, dan hasilnya yang berupa html biasa dikirimkan ke browser untuk ditampilkan. Jadi dari segi keamanan, jelas lebih aman dengan server-side scripting, sebab klien tidak akan dapat melihat script atau source code asli yang ada di server, sebab yang ditampilkan di browser adalah hasil eksekusi script tersebut.

Sintaks-sintaks yang digunakan dalam PHP ini kebanyakan diambil dari bahasa-bahasa pemrograman yang populer seperti C, Perl, dan Java dengan penambahan beberapa kelebihan dan keunikan yang dimiliki oleh PHP. Jadi jika Anda adalah programmer yang biasa menggunakan C, Perl, atau Java, Anda pasti tidak akan asing menggunakan fungsi-fungsi yang ada pada PHP. Namun jika Anda belum pernah bersinggungan sama sekali dengan bahasa-bahasa pemrograman tersebut, jangan berkecil hati, sebab banyak yang telah membuktikan bahwa mempelajari PHP tidaklah terlalu sulit, cukup bermodalkan kemauan dan tentunya referensi yang tepat J.

PHP memiliki beberapa kemampuan yang sangat mendukung dalam pembuatan halaman web yang interaktif dan menarik. Kemampuan tersebut antara lain : perhitungan matematis, informasi jaringan, mail, regular expression, dan yang paling menonjol adalah kemampuan PHP dalam menyediakan antarmuka dengan beberapa server database yang populer di pasaran, seperti MySQL, Oracle, Sybase, PostgreSQL, mSQL, dan lainnya.

Berikut beberapa fungsi yang sering digunakan dalam program PHP, yaitu ;

1. **Print (String Output)**. Fungsi ini menampilkan *Output* ke layar *browser*.
2. **Printf (String format,...)**. Fungsi ini untuk menampilkan ke layar *browser* dengan format yang dapat diatur.
3. **include (String filename)**. Menjalankan *file* yang disebutkan di dalam *argument*.
4. **boolean is_executable (String Filename)**. Fungsi ini digunakan untuk memeriksa apakah sebuah *file* bisa di eksekusi atau tidak.
5. **boolean is_file (string filename)**. Fungsi ini digunakan untuk mengetahui apakah *argument* yang ada berupa *file* atau tidak.
6. **boolean phpinfo()**. Fungsi ini berguna untuk mengetahui informasi tentang PHP version, sistem operasi pada *web server*, konfigurasi variabel, dan sebagainya.
7. **String phpversion ()**. Memberikan informasi mengenai versi php yang digunakan.
8. **string gethostbyaddr (string ip_address)**. Fungsi ini akan memberikan nama dari sebuah *host* yang *IP address*-nya disebut dalam *argument*.
9. **string gethostbyname (string hostname)**. Kebalikan dari fungsi di atasnya, dengan mengetahui nama dari sebuah *host*, maka Anda bisa mengetahui *IP address*-nya.
10. **integer strlen (string text)**. Mengembalikan panjang string dalam *argument*.
11. **string htmlentities (string text)**. Berguna untuk mengenali semua skrip yang ditulis dalam HTML sebagai format teks biasa.

12. **string strtolower (string text).** Mengubah karakter menjadi huruf kecil semua.
13. **string strtoupper (string text).** Mengubah karakter menjadi huruf besar semua. (*Submber ; Eko Priyo Utomo, ST (hal 45)*)