

BAB II

TINJAUAN PUSTAKA

II.1. Sistem Pendukung Keputusan

Konsep Sistem Pendukung Keputusan atau *Decision Support System* pertama kali diungkapkan pada awal tahun 1970-an oleh Michael S. Scott Morton dengan istilah *Management Decision System*. Definisi dari Sistem Pendukung Keputusan adalah suatu sistem berbasis komputer yang ditujukan untuk membantu pengambilan keputusan dalam memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. (Faisal,dkk; Jurnal Teknik Informasi dan Ilmu Komputer (JTIK):April 2015:12)

Sistem pendukung keputusan biasanya dibangun untuk mendukung solusi atas suatu masalah atau untuk mengevaluasi suatu peluang. Sistem pendukung keputusan yang seperti itu disebut aplikasi Sistem pendukung keputusan. Aplikasi Sistem pendukung keputusan digunakan dalam pengambilan keputusan. Aplikasi menggunakan CBIS (*Computer Based Information System*) yang fleksibel, interaktif, dan dapat diadaptasi,yang dikembangkan untuk mendukung solusi atas masalah manajemen spesifik yang tidak terstruktur.

Aplikasi Sistem pendukung keputusan menggunakan data, memberikan antar muka pengguna yang mudah,dan dapat menggabungkan pemikiran pengambilan keputusan. Sistem pendukung keputusan lebih ditujukan untuk mendukung manajemen dalam melakukan pekerjaan yang bersifat analitis dalam situasi yang kurang terstruktur dan dengan kriteria yang kurang jelas.

Sistem pendukung keputusan tidak dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia. Tujuan dari DSS adalah :

1. Membantu manajer dalam pengambilan keputusan atas masalah semistuktur.
 2. Memberikan dukungan atas pertimbangan manajer dan bukannya dimaksudkan untuk menggantikan fungsi manajer.
 3. Meningkatkan efektifitas keputusan yang diambil lebih daripada perbaikan efisiensinya.
 4. Kecepatan komputasi. Komputer memungkinkan para pengambil keputusan untuk melakukan banyak komputansi secara cepat dengan biaya rendah.
 5. Peningkatan produktivitas.
 6. Dukungan kualitas.
 7. Berdaya saing.
 8. Mengatasi keterbatasan kognitif dalam pemrosesan dan penyimpanan.
- (Sylvia Hartati Saragih; Pelita Informatika Budi Darma: Agustus 2013:83)

II.1.1. Hakikat Sistem Penunjang Keputusan

Pada dasarnya pengambilan keputusan adalah suatu pendekatan pendekatan sistematis pada hakekat suatu masalah, pengumpulan fakta-fakta,

penentuan yang matang dari alternatif yang dihadapi, dan pengambilan tindakan yang menurut perhitungan merupakan tindakan yang paling tepat. Pada sisi lain, pembuat keputusan

kerap kali dihadapkan pada kerumitan dan lingkup pengambilan keputusan dengan datayang begitu banyak. Untuk kepentingan itu, sebagian besar pembuat keputusan mempertimbangkan rasio biaya atau manfaat, dihadapkan pada suatu keharusan untuk mengandalkan seperangkat sistem yang mampu memecahkan masalah secara efisien dan efektif, yang kemudian disebut Sistem Pendukung Keputusan. (Iwan Rijayana,dkk; Seminar Nasional Informatika 2012(*semnasIF 2012*):C-49)

II.2. Sekolah

Sekolah merupakan lembaga pendidikan formal yang sistematis melaksanakan program bimbingan, pengajaran, dan latihan dalam rangka membantu siswa agar mampu megembangkan potensinya baik yang menyangkut aspek moral, spiritual, intelektual, emosional maupun sosial. Dalam pemilihan sekolah menengah pertama yang baik dan benar bagi siswa Sekolah Dasar yang akan melanjutkan pendidikannya, banyak sekali kriteria-kriteria yang harus dilihat dari banyaknya sekolah Negeri dan sekolah Swasta yang beredar di pasaran saat ini, yaitu salah satu faktor menentukan kualitas dari sekolah tersebut, lokasi sekolah tersebut. (Bardansyah: Pelita Informatika Budi Darma: April 2014:21)

II.2.1. Guru

Guru adalah seseorang yang memiliki tugas sebagai fasilitator agar siswa dapat belajar dan atau mengembangkan potensi dasar dan kemampuannya secara optimal, melalui lembaga pendidikan sekolah, baik yang didirikan oleh pemerintah maupun

masyarakat atau swasta. Dengan demikian, dalam pandangan umum, guru tidak hanya dikenal secara formal sebagai pendidik, pengajar, pelatih, dan pembimbing tetapi juga sebagai “ *social agent hired by society to help facilitate members of society who attend schools*”, atau agen sosial yang diminta oleh masyarakat untuk memberikan bantuan kepada warga masyarakat yang akan dan sedang berada di bangku sekolah. (Drs. Suparlan, M.Ed:10)

II.2.2. Visi, Misi, dan Moto

Dalam dunia pendidikan di butuhkan suatu acuan yang dapat mengembangkan organisasi tersebut agar lebih maju dan berkembang, dalam hal ini visi, misi, dan moto setiap sekolah selalu ada, berikut pengertian dari hal tersebut :

1. Visi adalah suatu pandangan jauh kedepan, mengenai apa yang harus dilakukan oleh perusahaan, tujuan perusahaan dan tindakan apa yang harus dilakukan untuk mewujudkan hal tersebut.

Contoh visi “Unggul dalam berprestasi, beriman, terdidik dan berbudaya”.

2. Misi merupakan suatu hal (pernyataan) yang dilakukan oleh suatu lembaga tertentu dalam mewujudkan Visi.

Contoh misi :

- a. Membina dan melatih peserta didik dengan keimanan dan ketaqwaan menurut Agama dan kepercayaan masing-masing.
 - b. Meningkatkan prestasi secara optimal melalui kegiatan bimbingan belajar pelatihan secara efektif.
 - c. Mengawasi dan mengembangkan sistem pembelajaran yang mengikuti perkembangan metode pendidikan bertaraf nasional.
 - d. Menetapkan budaya bekerjasama dan sama – sama bekerja untuk meningkatkan kualitas pendidikan.
 - e. Menerapkan manajemen partisipasi dengan prinsip kemintraan seluruh warga sekolah dengan komite sekolah, orang tua murid, alumni, lintas sektoral dan pihak sekolah.
3. Moto atau dalam bahasa Inggris dituliskan dengan *Motto*, merupakan sebuah kalimat ataupun kata yang dijadikan sebagai prinsip dan semboyan dalam kehidupan. Contoh Moto : “Sekolahku harapan masa depanku”. (SMP Negeri 25 Medan)

II.2.3. Struktur Organisasi

Apabila suatu organisasi hanya terdiri atas dua orang dan tujuan yang akan dicapai juga hanya sederhana, maka belum diperlukan struktur organisasi.

Jika kelompok yang bekerjasama berjumlah besar, dan tujuan yang akan dicapai luas, maka struktur organisasi yang tersusun rapi mutlak perlu.

Struktur organisasi ialah suatu kerangka yang menunjukkan semua tugas kerja untuk mencapai tujuan organisasi, hubungan antara fungsi-fungsi tersebut, serta wewenang dan tanggung jawab setiap anggota organisasi yang melakukan tiap-tiap tugas kerja tersebut.

Struktur organisasi diperlukan untuk memberi wadah tujuan, misi, tugas pokok dan fungsi. Jika fungsi yang diselenggarakan berlangsung secara terus menerus, maka harus dilembagakan agar memungkinkan berlakunya fungsionalisasi yang menjadi landasan peningkatan efisiensi dan efektivitas organisasi. (Darmono:2001:25)

II.3. *Analytical Hierarchy Process (AHP)*

Analytical Hierarchy Process (AHP) merupakan suatu model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks menjadi suatu hirarki, menurut Saaty, hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multi level dimana level pertama adalah tujuan, yang diikuti level faktor, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terakhir dari alternatif. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis. sering digunakan

sebagai metode pemecahan masalah dibanding dengan metode yang lain karena alasan-alasan sebagai berikut:

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam.
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh pengambil keputusan. Memperhitungkan daya tahan *output* analisis sensitivitas pengambilan keputusan. (Sylvia Hartati Saragih; Pelita Informatika Budi Darma: Agustus 2013:83)

II.3.1. Tahapan Metode AHP

Dalam metode *Analytical Hierarchy Process* dilakukan langkah-langkah sebagai berikut :

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan
Dalam tahap ini penulis berusaha menentukan masalah yang akan penulis pecahkan secara jelas, detail dan mudah dipahami. Dari masalah yang ada penulis coba tentukan solusi yang mungkin cocok bagi masalah tersebut. Solusi dari masalah mungkin berjumlah lebih dari satu. Solusi tersebut nantinya penulis kembangkan lebih lanjut dalam tahap berikutnya.

2. Membuat struktur hierarki yang diawali dengan tujuan utama

Setelah menyusun tujuan utama sebagai level teratas akan disusun level hirarki yang berada di bawahnya yaitu kriteria-kriteria yang cocok untuk mempertimbangkan atau menilai alternatif yang penulis berikan dan menentukan alternatif tersebut. Tiap kriteria mempunyai intensitas yang berbeda-beda. Hirarki dilanjutkan dengan subkriteria (jika mungkin diperlukan).

3. Membuat matrik perbandingan berpasangan yang menggambarkan kontribusi relatif atau pengaruh setiap elemen terhadap tujuan atau kriteria yang setingkat di atasnya. Matriks yang digunakan bersifat sederhana, memiliki kedudukan kuat untuk kerangka konsistensi, mendapatkan informasi lain yang mungkin dibutuhkan dengan semua perbandingan yang mungkin dan mampu menganalisis kepekaan prioritas secara keseluruhan untuk perubahan pertimbangan. Pendekatan dengan matriks mencerminkan aspek ganda dalam prioritas yaitu mendominasi dan didominasi. Perbandingan dilakukan berdasarkan judgment dari pengambil keputusan dengan menilai tingkat kepentingan suatu elemen dibandingkan elemen lainnya. Untuk memulai proses perbandingan berpasangan dipilih sebuah kriteria dari level paling atas hirarki misalnya K dan kemudian dari level di bawahnya diambil elemen yang akan dibandingkan misalnya E1,E2,E3,E4,E5.

4. Melakukan Mendefinisikan perbandingan berpasangan sehingga diperoleh jumlah penilaian seluruhnya sebanyak $n \times [(n-1)/2]$ buah, dengan n adalah banyaknya elemen yang dibandingkan. Hasil perbandingan dari masing-masing elemen akan berupa angka dari 1 sampai 9 yang menunjukkan perbandingan tingkat kepentingan suatu elemen. Apabila suatu elemen dalam matriks dibandingkan dengan dirinya sendiri maka hasil perbandingan diberi nilai 1. Skala 9 telah terbukti dapat diterima dan bisa membedakan intensitas antar elemen. Hasil perbandingan tersebut diisikan pada sel yang bersesuaian dengan elemen yang dibandingkan. Skala perbandingan perbandingan berpasangan dan maknanya yang diperkenalkan oleh Saaty bisa dilihat di bawah. Intensitas Kepentingan:
 - a. 1 berarti kedua elemen sama pentingnya, Dua elemen mempunyai pengaruh yang sama besar
 - b. 3 berarti elemen yang satu sedikit lebih penting daripada elemen yang lainnya, Pengalaman dan penilaian sedikit menyokong satu elemen dibandingkan elemen yang lainnya
 - c. 5 berarti elemen yang satu lebih penting daripada yang lainnya, Pengalaman dan penilaian sangat kuat menyokong satu elemen dibandingkan elemen yang lainnya
 - d. 7 berarti satu elemen jelas lebih mutlak penting daripada elemen lainnya, Satu elemen yang kuat disokong dan dominan terlihat dalam praktek.

- e. 9 berarti satu elemen mutlak penting daripada elemen lainnya, Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat penegasan tertinggi yang mungkin menguatkan.
- f. 2,4,6,8 berarti nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan, Nilai ini diberikan bila ada dua kompromi di antara 2 pilihan Kebalikan = Jika untuk aktivitas i mendapat satu angka dibanding dengan aktivitas j , maka j mempunyai nilai kebalikannya dibanding dengan i
5. Menghitung nilai eigen dan menguji konsistensinya. Jika tidak konsisten maka pengambilan data diulangi.
 6. Mengulangi langkah 3,4, dan 5 untuk seluruh tingkat hirarki.
 7. Menghitung vektor eigen dari setiap matriks perbandingan berpasangan yang merupakan bobot setiap elemen untuk penentuan prioritas elemen- elemen pada tingkat hirarki terendah sampai mencapai tujuan. Penghitungan dilakukan lewat cara menjumlahkan nilai setiap kolom dari matrik, membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks, dan menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan rata-rata.
 8. Memeriksa konsistensi hirarki. Adapun yang diukur dalam Analytical Hierarchy Process adalah rasio konsistensi dengan melihat index konsistensi. Konsistensi yang diharapkan adalah yang mendekati sempurna agar menghasilkan keputusan yang mendekati valid.

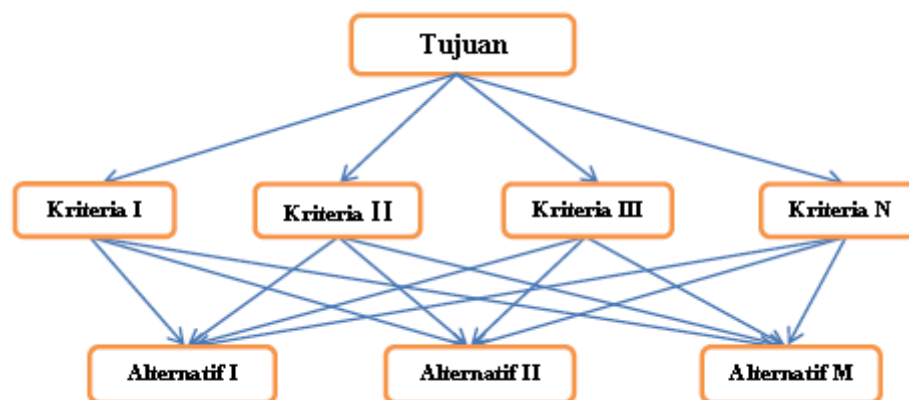
9. Walaupun sulit untuk mencapai yang sempurna, rasio konsistensi diharapkan kurang dari atau sama dengan 10 %. (Sylvia Hartati Saragih; Pelita Informatika Budi Darma: Agustus 2013:83)

II.3.2. Prosedur *Analytical Hierarchy Prosecess* (AHP)

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, di antaranya adalah sebagai berikut :

1. *Decomposition* (membuat hierarki)

Sistem yang kompleks bisa dipahami dengan memecahkannya menjadi elemen-elemen yang lebih kecil dan mudah dipahami, ditunjukkan oleh gambar II.1 dibawah ini :



Gambar II.1. Struktur Hirarki

(Sumber : Dalu Nulul Kirom, dkk; Jurnal Teknik ITS: Sept 2012:A-155)

2. *Comparative judgment* (penilaian kriteria alternatif)

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat

kualitatif dari skala perbandingan Saaty dapat diukur menggunakan tabel analisis seperti yang ditunjukkan oleh Tabel II.1. berikut ini.

Tabel II.1 Skala Penilaian Perbandingan Berpasangan

Tingkat Kepentingan	Definisi	Keterangan
1	Sama Pentingnya	Kedua elemen mempunyai pengaruh yang sama.
3	Agak lebih penting yang satu atas lainnya	Pengalaman dan penilaian sangat memihak satu elemen dibandingkan dengan pasangannya.
5	cukup penting	Pengalaman dan keputusan menunjukkan kesukaan atas satu aktifitas lebih dari yang lain
7	Sangat penting	Pengalaman dan keputusan menunjukkan kesukaan yang kuat atas satu aktifitas lebih dari yang lain
9	Mutlak lebih penting	Satu elemen mutlak lebih disukai dibandingkan dengan pasangannya, pada tingkat keyakinan tertinggi.
2,4,6,8	Nilai tengah diantara dua nilai berdekatan	Bila kompromi dibutuhkan

(Sumber : Dalu Nulul Kirom, dkk; Jurnal Teknik ITS: Sept 2012:A-155)

3. *Synthesis of priority* (Menentukan Prioritas)

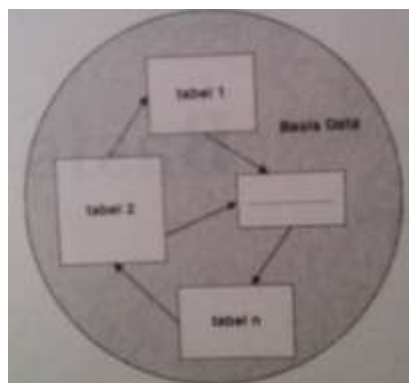
Menentukan prioritas dari elemen-elemen kriteria dapat dipandang sebagai bobot/kontribusi elemen tersebut terhadap tujuan pengambilan keputusan. Prioritas ini ditentukan berdasarkan pandangan para pakar dan pihak-pihak yang berkepentingan terhadap pengambilan keputusan, baik secara langsung (diskusi) maupun secara tidak langsung (kuisisioner).

4. Logical Consistency (konsistensi logis)

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antar objek yang didasarkan pada kriteria tertentu. (Dalu Nuzlul Kirom, dkk; Jurnal Teknik ITS: Sept 2012:a-155)

II. 4. Basis Data

Sistem Basis data adalah sistem tekomputerisasi yang tujuan utamanya adalah memelihara data yang sudah di olah atau informasi dan membuat informasi tersedia saat dibutuhkan. Pada intinya basis data adalah media untuk menyimpan data agar dapat diakses dengan mudah dan cepat yang diimplementasikan dengan tabel-tabel yang saling memiliki relasi seperti pada gambar berikut :



Gambar II.2 Ilustrasi Basis Data
(Sumber : Rosa A.S, dkk: Juli 2015:44)

Sistem informasi tidak dapat dipisahkan dengan kebutuhan akan basis data apapun bentuknya, entah berupa *file* teks ataupun *Database Menagement System* (DBMS). Kebutuhan basis data dalam sistem informasi meliputi :

- a. Memasukkan, menyimpan, dan mengambil data
- b. Membuat laporan berdasarkan data yang telah disimpannya

Tujuan dari dibuatnya tabel-tabel di sini adalah untuk menyimpan data ke dalam tabel-tabel agar mudah diakses. Oleh karena itu, untuk merancang tabel-tabel yang akan dibuat maka dibutuhkan pola pikir penyimpanan data nantinya jika bentuk baris-baris data (*record*) dimana setiap baris terdiri dari beberapa kolom.

II.4.1. DBMS (*Database Management System*)

DBMS (*Database Management System*) atau dalam bahasa Indonesia sering disebut sebagai sistem manajemen basis data adalah suatu sistem aplikasi yang digunakan untuk menyimpan, mengelola, dan menampilkan data. Suatu sistem aplikasi disebut DBMS jika memenuhi persyaratan minimal sebagai berikut :

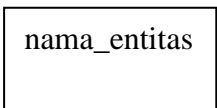
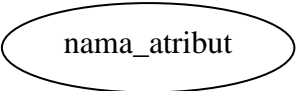
- a. Menyediakan fasilitas untuk mengelola akses data
- b. Mampu menangani integritas data
- c. Mampu menangani akses data
- d. Mampu menangani *backup* data

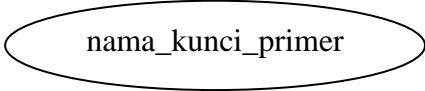
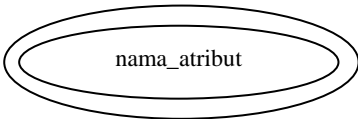
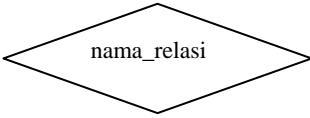
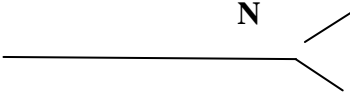
Karena pentingnya data bagi suatu organisasi/perusahaan, maka hampir sebagian besar perusahaan memanfaatkan DBMS dalam mengelola data yang mereka miliki. Pengelolaan DBMS sendiri biasanya ditangani oleh tangan ahli yang spesialis menangani DBMS yang disebut sebagai DBA (*Database Administrator*).

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

Pemodelan awal basis data yang paling banyak digunakan adalah menggunakan *Entity Relationship Diagram* (ERD). ERD dikembangkan berdasarkan teori himpunan dalam bidang matematika. ERD digunakan untuk pemodelan basis data relasional. Sehingga jika menyimpan basis data menggunakan OODBMS maka perancangan basis data tidak perlu menggunakan ERD. ERD memiliki beberapa aliran notasi seperti notasi chen (dikembangkan oleh Peter Chen). Barker (dikembangkan oleh Richard Berker, Ian Palmer, Harry Ellis), notasi Crows Foot, dan beberapa notasi lain. Namun yang banyak digunakan adalah notasi dari Chen. Berikut adalah simbol-simbol yang digunakan pada ERD dengan notasi Chen :

Tabel II.2 Simbol-simbol ERD (*Entity Relationship Diagram*)

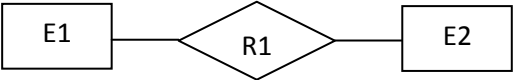
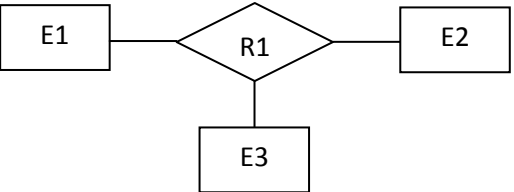
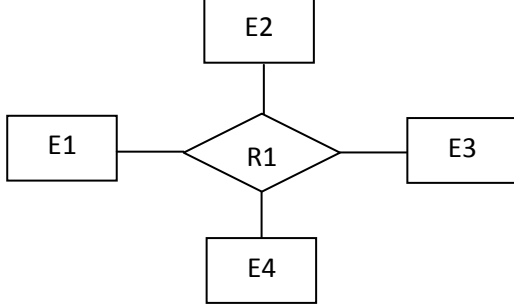
Simbol	Deskripsi
Entitas / <i>Entity</i> 	Entitas merupakan data inti yang akan disimpan; bakal tabel pada basis data; benda yang memiliki data dan harus disimpan datanya agar dapat diakses oleh aplikasi komputer; penamaan entitas biasanya lebih ke kata benda dan belum merupakan nama tabel
Atribut 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas

Simbol	Deskripsi
Atribut kunci primer 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas dan digunakan sebagai kunci akses record yang diinginkan; biasanya berupa id; kunci primer dapat lebih dari satu kolom, asalkan kombinasi dari beberapa kolom tersebut dapat bersifat unik (berbeda tanpa ada yang sama)
Atribut multi nilai / <i>multivalued</i> 	<i>Field</i> atau kolom data yang butuh disimpan dalam suatu entitas yang dapat memiliki nilai lebih dari satu
Relasi 	Relasi yang menghubungkan antar entitas; biasanya diawali dengan kata kerja
Asosiasi / <i>association</i> 	Perhubungan antar relasi dan entitas dimana di kedua ujungnya memiliki multiplicity kemungkinan jumlah pemakaian. Kemungkinan jumlah maksimum keterhubungan antar entitas yang lain disebut dengan kardinalitas. misalkan ada kardinalitas 1 ke N atau sering disebut dengan <i>one to many</i> menghubungkan entitas A dan entitas B

(Sumber : Rosa A.S, dkk: Juli 2015:50-51)

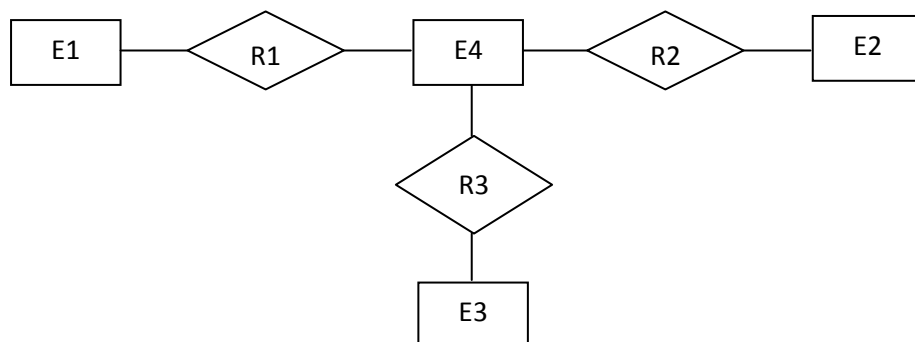
ERD biasanya memiliki hubungan *binary* (satu relasi menghubungkan dua buah entitas). Beberapa metode perancangan ERD menoleransi hubungan relasi *ternary* (satu relasi menghubungkan tiga buah relasi) atau *N-ary* (satu relasi menghubungkan banyak entitas), tapi banyak metode perancangan ERD yang tidak mengizinkan hubungan *ternary* atau *N-ary*. Berikut adalah contoh bentuk hubungan relasi dalam ERD :

Tabel II.3 Hubungan Relasi Dalam ERD

Nama	Gambar
<i>Binary</i>	
<i>Ternary</i>	
<i>N-ary</i>	

(Sumber : Rosa A.S, dkk: Juli 2015:52)

Beberapa metode perancangan ERD memberikan solusi untuk perancangan ERD yang tidak *binary* diubah sebagai berikut :



Gambar II.3 Cara Menghindari Relasi Ternary

(Sumber : Rosa A.S, dkk: Juli 2015:52)

Entitas E4 berasal dari relasi R1 yang dijadikan entitas, karena banyak metode perancangan ERD yang menyatakan bahwa jika terjadi relasi *ternary*, maka sebenarnya relasinya lebih layak di jadikan entitas dibandingkan menjadi relasi.

II.5.1. Kamus Data

Kamus data (*data binary*) dipergunakan untuk meperjelas 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 dalam implementasi program dapat menjadi parameter masukan atau keluaran dari sebuah fungsi atau prosedur. Kamus data biasanya berisi:

- a. Nama-nama dari data
- b. Digunakan pada merupakan proses-proses yang terkait data
- c. Deskripsi merupakan deskripsi data, nilai data, batas nilai data, dan komponen yang membentuk data.

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

Tabel II.4 Simbol Informasi Tambahan

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

(Sumber : Rosa A.S, dkk: Juli 2015:74)

Kamus data pada pada DFD nanti harus dapat dipetakan dengan hasil perancangan basis data yang dilakukan sebelumnya. Jika ada kamus data yang tidak dapat dipetakan pada tabel hasil perencanaan basis dengan DFD masih belum sesuai, sehingga harus ada yang diperbaiki baik perancangan basis datanya, perancangan DFD-nya, atau keduanya.

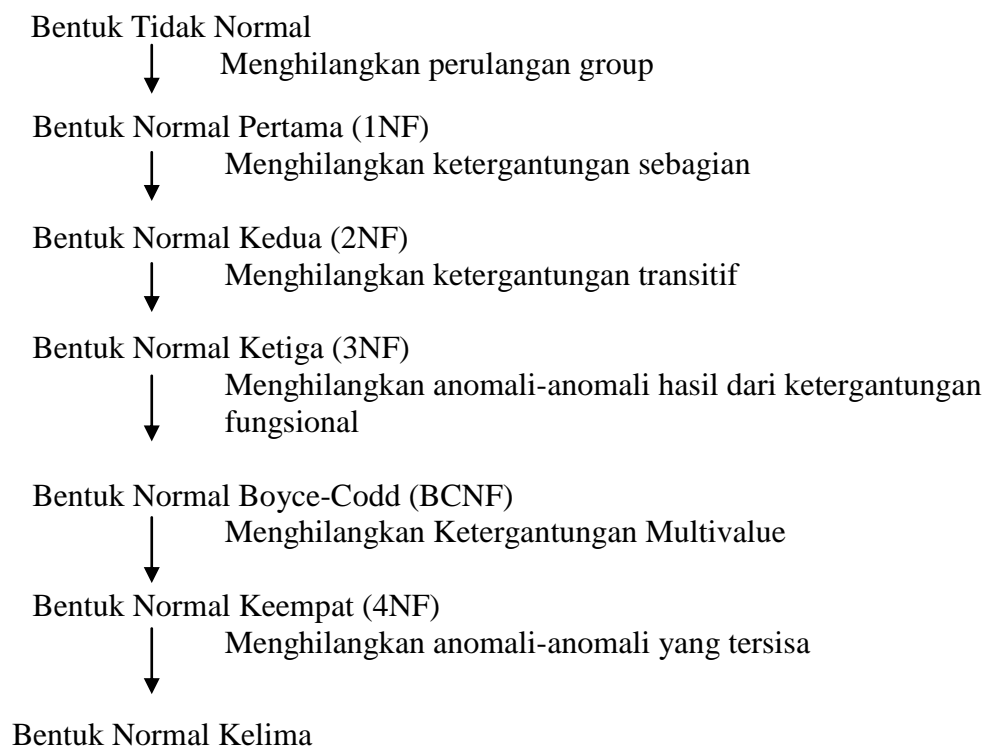
II.5.2. Normalisasi

Beberapa pengertian mengenai normalisasi :

- a. Istilah Normalisasi berasal dari E. F.Codd, salah seorang perintis teknologi basis data. selain dipakai sebagai metodologi tersendiri untuk menciptakan struktur tabel (9 relasi) dalam basis data (dengan tujuan untuk mengurangi kemubaziran data) , normalisasi terkadang hanya dipakai sebagai perangkat verifikasi terhadap tabel-tabel yang dihasilkan oleh metodologi lain (misalnya E -R). Normalisasi memberikan panduan yang sangat membantu bagi pengembang untuk mencegah penciptaan struktur tabel yang kurang fleksibel atau mengurangi kefleksibelan.
- b. Normalisasi merupakan sebuah teknik dalam logical desain sebuah basis data / database, teknik pengelompokkan atribut dari suatu relasi sehingga membentuk struktur relasi yang baik (tanpa redundansi).
- c. Normalisasi adalah suatu proses memperbaiki / membangun dengan model data relasional, dan secara umum lebih tepat dikoneksikan dengan model data logika.

Proses normalisasi adalah proses pengelompokan data elemen menjadi tabel-tabel yang menunjukkan entity dan relasinya. Pada proses normalisasi dilakukan pengujian pada beberapa kondisi apakah ada kesulitan pada saat menambah/menyisipkan, menghapus, mengubah dan mengakses pada suatu basis data. Bila terdapat kesulitan pada pengujian tersebut maka perlu dipecahkan relasi pada beberapa tabel lagi atau dengan kata lain perancangan basis data belum optimal. Tujuan dari normalisasi itu sendiri adalah untuk menghilangkan kerangkapan data, mengurangi kompleksitas, dan untuk mempermudah pemodifikasian data.

II.5.3. Jenis-jenis Normalisasi



Penjelasan dari alur diatas adalah sebagai berikut:

a. Bentuk Normal Kesatu (1NF/FirstNormalForm)

Bentuk Bentuk Normal Kesatu mempunyai ciri yaitu setiap data dibentuk dalam file flat, data dibentuk dalam satu record demi satu record dan nilai dari field berupa “atomic value”. Tidak ada set atribut yang berulang ulang atau atribut bernilai ganda (multi value). Tiap field hanya satu pengertian, bukan merupakan kumpulan data yang mempunyai arti mendua. Hanya satu arti saja dan juga bukanlah pecahan kata kata sehingga artinya lain.

Contoh :

Kode Kelas	Nama Kelas	Pengajar
1111	Basis Data	Muhamad Ali
2222	Riset Pemasaran	Ahmad Yunani
3333	Pemograman	Suryo Pratolo

Mahasiswa (NPM, Nama, Dosen Wali, Semester1, Semester2 Semester3) Mahasiswa yang punya NPM, Nama, Dosen Wali mengikuti 3 mata kuliah. Di sini ada perulangan semester sebanyak 3 kali. Bentuk seperti ini bukanlah 1NF.

Contoh Data :

NPM	Nama	Dosen Wali	Sem1	Sem2	Sem3
1000	Sally Fatimah	Dedy S	1234		3100
1001	Inul Daratista	Ruslan	1234	2109	
1002	Putri Patricia	Denmas		2100	3122

Bentuk 1 NF dari bentuk diatas adalah sebagai berikut :

NPM	Nama	Dosen Wali	Semester
1000	Sally Fatimah	Dedy S	1234
1000	Sally Fatimah	Dedy S	3100
1001	Inul Daratista	Ruslan	1234
1001	Inul Daratista	Ruslan	2109
1002	Putri Patricia	Denmas	2100
1002	Putri Patricia	Denmas	3122

b. Bentuk Normal Kedua (2NF)

Bentuk Normal kedua mempunyai syarat yaitu bentuk data telah memenuhi kriteria bentuk Normal Kesatu. Atribut bukan kunci haruslah bergantung secara fungsi pada kunci utama, sehingga untuk membentuk Normal Kedua haruslah sudah ditentukan kunci-kunci field. Kunci field harus unik dan dapat mewakili atribut lain yang menjadi anggotanya. Dari contoh relasi mahasiswa pada bentuk Normal Kesatu, terlihat bahwa kunci utama adalah NPM. Nama Mahasiswa dan Dosen Wali bergantung pada NPM, Tetapi Kode Semester bukanlah fungsi dari Mahasiswa maka file siswa dipecah menjadi 2 relasi yaitu :

Relasi Mahasiswa

NPM	Nama	Dosen Wali
1000	Sally Fatimah	Dedy S
1001	Inul Daratista	Ruslan
1002	Putri Patricia	Denmas

Dan relasi ambil kuliah

NPM	Kode Kuliah
1000	1234
1000	3100
1001	1234
1001	2109
1001	2100
1001	3122

c. Bentuk Normal Ketiga (3NF)

Untuk menjadi bentuk Normal Ketiga maka relasi haruslah dalam bentuk Normal Kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Artinya setiap atribut bukan kunci harus bergantung hanya pada kunci primer secara menyeluruh. Contoh pada bentuk Normal kedua di atas termasuk juga bentuk Normal Ketiga karena seluruh atribut yang ada di situ bergantung penuh pada kunci primernya.

d. *Boyce-Codd Normal Form* (BCNF)

Boyce-Codd Normal Form mempunyai paksaan yang lebih kuat dari bentuk Normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk Normal Kesatu dan setiap atribut dipaksa bergantung pada fungsi pada atribut super key. Pada contoh dibawah ini terdapat relasi Seminar, Kunci Primer adalah NPM + Seminar. Siswa boleh mengambil satu atau dua seminar. Setiap seminar membutuhkan 2 pembimbing dan setiap siswa dibimbing oleh salah satu diantara 2 pembimbing seminar tersebut. Setiap pembimbing hanya boleh mengambil satu seminar saja. pada contoh ini NPM dan Seminar menunjukkan seorang Pembimbing.

Relasi seminar:

NPM	Seminar	Pembimbing
1000	S100	Siska
1001	S100	Sinta
1002	S101	Sukma
1001	S101	Sukma
1003	S101	Akbar

Bentuk Relasi Seminar adalah bentuk Normal Ketiga, tetapi tidak BCNF karena Kode Seminar masih bergantung fungsi pada Pembimbing, jika setiap Pembimbing dapat mengajar hanya satu seminar. Seminar bergantung pada satu atribut bukan super key seperti yang disyaratkan oleh BCNF. Maka relasi Seminar harus dipecah menjadi dua yaitu :

Relasi Pembimbing

Pembimbing	Seminar
Siska	S100
Sinta	S100
Sukma	S101
Akbar	S101

Relasi Seminar-Pembimbing

NPM	Pembimbing
1000	Siska
1001	Sinta
1002	Sukma
1001	Sukma
1003	Akbar

(Sumber : I Wayan Susena; Sistem Komputer STIKOM BALI:2011:6-11)

II.6. Database

Database merupakan kumpulan data yang pada umumnya menggambarkan aktifitas-aktifitas dan pelakunya dalam suatu organisasi. Sistem database adalah sistem komputer yang digunakan untuk menyimpan dan mengelola data tersebut. Sistem database memiliki empat komponen penting, yakni :

1. *Data*, merupakan informasi yang disimpan dalam suatu struktur tertentu yang terintegrasi.
2. *Hardware*, merupakan perangkat keras berupa komputer dengan media penyimpanan sekunder yang digunakan untuk menyimpan data karena pada umumnya database memiliki ukuran yang besar.
3. *Software*, merupakan perangkat lunak yang digunakan untuk melakukan pengelolaan data. Perangkat lunak ini sering disebut sebagai *Database Management System (DBMS)*.
4. *User*, merupakan orang yang menggunakan data yang tersimpan dan dikelola. *User* dapat berupa seorang yang mengelola database tersebut, yang disebut dengan database *administrator (DBA)*, bisa juga end user yang mengambil hasil dari pengelolaan database melalui bahasa *query*. *User* juga dapat seorang programmer yang membangun aplikasi yang terhubung ke database dengan menggunakan bahasa pemrograman.
(Yuliandri Priyo Nugroho:2012:13)

II.6.1. SQL (*Structured Query Language*)

SQL (*Structured Query Language*) adalah bahasa yang digunakan untuk mengelola data pada RDBMS. SQL awalnya dikembangkan berdasarkan teori aljabar relasional dan kalkulus. SQL mulai berkembang pada tahun 1970AN. SQL mulai digunakan sebagai standar yang resmi pada tahun 1986 oleh ANSI (*American National Standards Institute*) dan pada tahun 1987 oleh ISO (*International Organization For Standardization*) dan disebut sebagai SQL-86.

Pada perkembangannya, SQL beberapa kali dilakukan revisi. Berikut ini sejarah SQL sampai saat ini :

Tabel II.5 Sejarah SQL

No	Tahun	Nama
1	1986	SQL-86
2	1989	SQL-89
3	1992	SQL-89
4	1999	SQL:1999
5	2003	SQL:2003
6	2006	SQL:2006
7	2008	SQL:2008
8	2011	SQL:2011

(Sumber : Rosa A.S, dkk: Juli 2015:46)

II.7. Visual Basic 2010

Aplikasi pemrograman visual adalah *software* yang berguna untuk membuat *software* dengan antar muka grafis (*graphical user interface / GUI*). Sebetulnya aplikasi ini membuat *software text-based*, hanya saja lebih sering digunakan untuk membuat *software* dengan antar muka grafis. Saat ini ada banyak aplikasi pemrograman visual yang beredar. Borland membuat Delphin, C++ bulder, dan J Builder. Microsoft mengeluarkan VB, VB.net, C#.net, F#.ne, maupun C++.net.

Pada intinya, Visual Basic.Net adalah salah satu dari kumpulan *tools* pemrograman yang terdapat pada paket Visual Studio.Net Pada Visual Studio terdapat beberapa *tools* pemrograman lain seperti: Visual C++.Net, Visual C#.Net, dan Visual J#.Net. (Priyanto Hidayatullah:2015:3)

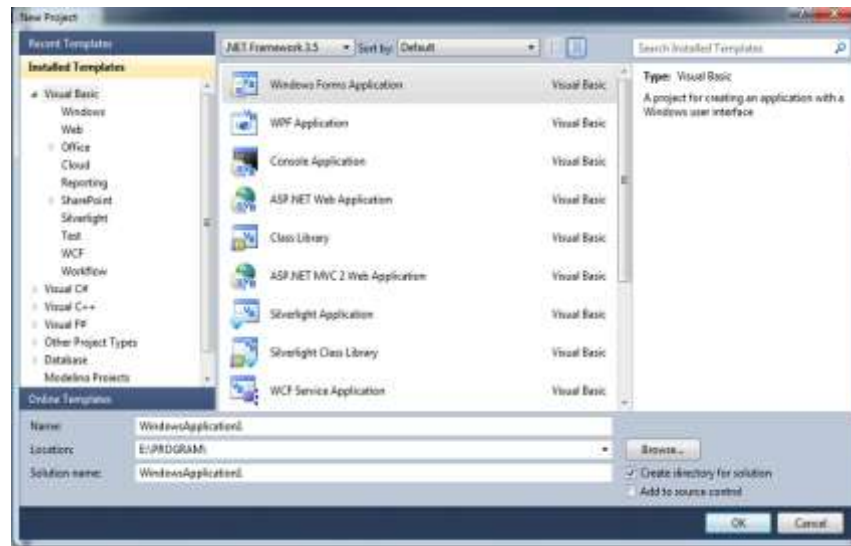
II.7.1. Pengenalan Visual Basic 2010

Pada waktu *Visual Basic* 2010 dijalankan, akan tampil sebuah *start page* dari seperti terlihat pada gambar II.5 berikut ini :

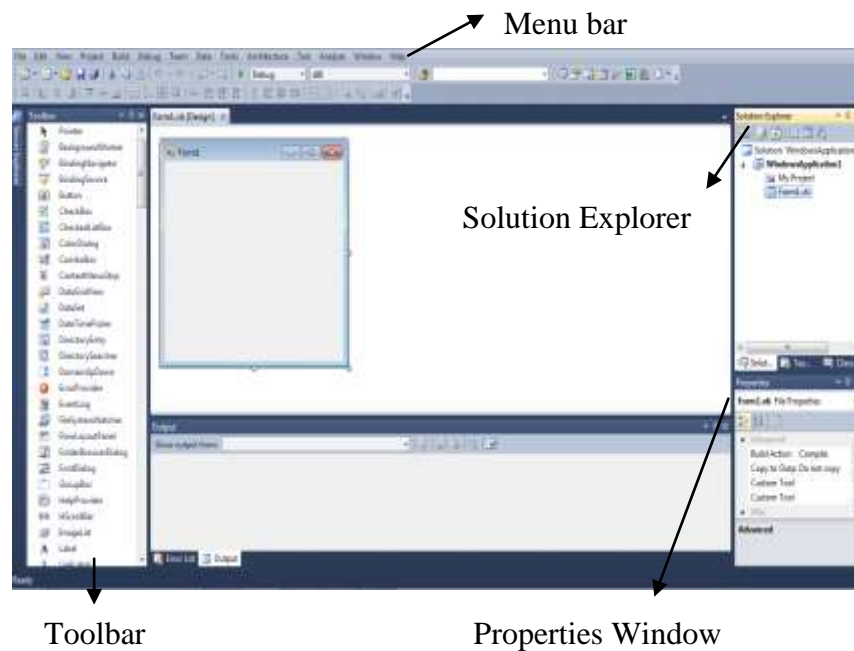


Gambar II. 4 *Start Page* dari *Visual Studio* 2010
(Sumber: Priyanto Hidayatullah:2015:25)

Pada IDE *Visual Studio* 2010 untuk *Windows Application* secara *default* telah terdapat sebuah *form*. *Form* tersebut bernama *Form1*. Pada *form* inilah tempat meletakkan kontrol-kontrol atau komponen-komponen untuk membuat sebuah aplikasi *Windows Form* dan control-kontrol dari aplikasi inilah yang biasanya disebut dengan GUI (*Graphical User Interface*). Jadi *user* akan berinteraksi dengan sebuah program aplikasi melalui GUI. Pada IDE *Visual Studio* 2010 terdapat *menu bar*, *toolbar*, *toolbox*, *solution explorer*, dan *properties window*.



Gambar II.5 Kotak Dialog *New Project*
 (Sumber: Priyanto Hidatatullah:2015:26)



Gambar II.6 IDE Visual Studio 2010
 (Sumber: Priyanto Hidatatullah:2015:27)

II.8. UML (*Unified Modeling Language*)

Pada perkembangan teknik pemrograman berorientasi objek, muncullah sebuah standarisasi bahasa pemodelan untuk pembangunan perangkat lunak yang dibangun dengan menggunakan teknik pemrograman berorientasi objek, yaitu *Unified Modelling Language* (UML). UML muncul karena adanya kebutuhan pemodelan visual untuk menspesifikasikan, menggambarkan, membangun, dan dokumentasi dari sistem perangkat lunak. UML merupakan bahasa visual untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. (Rosa A.S,dkk:2014:137)

Dalam pembuatan skripsi ini penulis menggunakan beberapa model diagram yang terdapat di dalam UML, yaitu : *Use case Diagram*, *Class Diagram*, *Sequence Diagram*, *Activity Diagram*. Adapun maksud dari model diagram tersebut diterangkan di bawah ini.

1. *Use case Diagram*

Use case atau diagram *use case* merupakan pemodelan untuk melakukan kelakuan (*behavior*) sistem informasi yang akan dibuat. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. Ada dua hal utama pada *use case* yaitu pendefinisian apa yang disebut aktor dan *use case*.

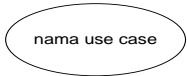



- a. Aktor merupakan orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun

simbol dari aktor adalah gambar orang, tapi aktor belum tentu orang.

- b. *Use case* merupakan fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor. (Rosa A.S, M. Shalahuddin:2014:137)

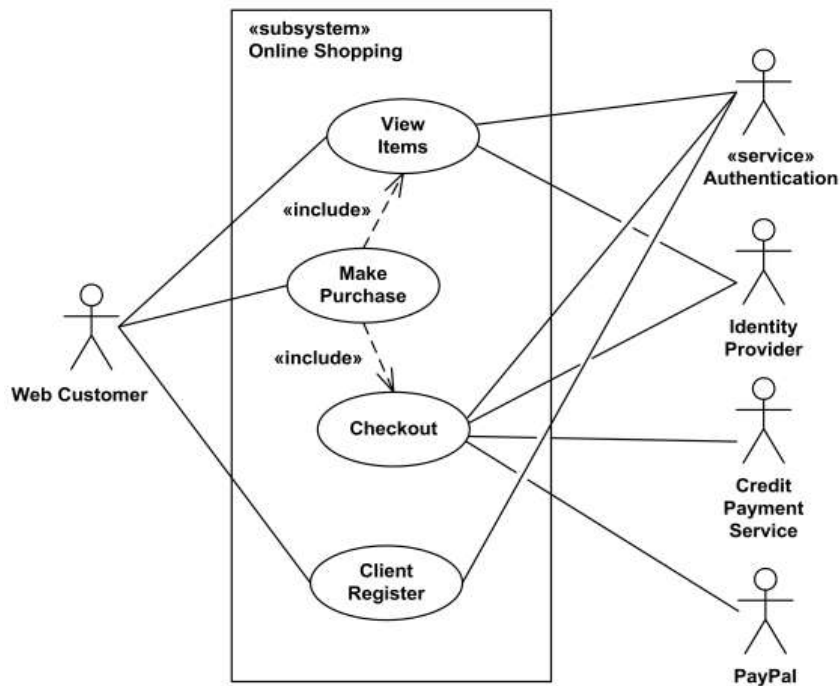
Berikut ini simbol-simbol yang ada pada diagram use case:

Tabel. II.6. Simbol-simbol *use case* diagram

Simbol	Deskripsi
<p><i>Use case</i></p> 	<p>Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau actor; biasanya dinyatakan dengan menggunakan kata kerja di awal di frase nama <i>Use case</i>.</p>
<p>Aktor /actor</p> 	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu orang; biasanya dinyatakan menggunakan kata benda di awal frase nama actor.</p>
<p>Asosiasi / <i>association</i></p> 	<p>Komunikasi antar actor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan actor</p>
<p>Ekstensi / <i>extend</i></p> 	<p>Relasi <i>use case</i> tambahan ke sebuah <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walaupun tanpa <i>use case</i> tambahan itu; mirip dengan prinsip <i>inheritance</i> pada pemograman berorientasi objek; biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang di tambahkan.</p>

(Rosa A.S,dkk:2014:156)

Contoh *use case* diagram





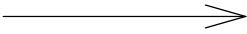
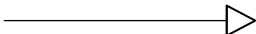


Gambar II.7 Contoh Use Case Diagram

2. Class Diagram

Diagram kelas atau *class diagram* menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Diagram kelas dibuat agar pembuat program atau *programmer* membuat kelas-kelas sesuai rancangan di dalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron. (Rosa A.S, M. Shalahuddin:2014:141,142,146)

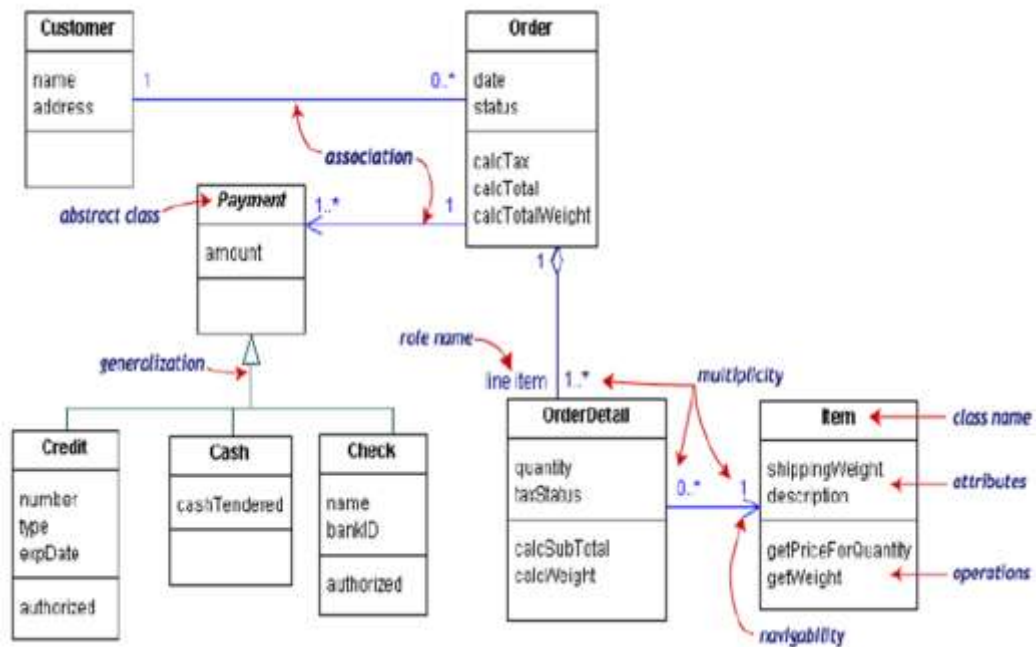
Berikut adalah symbol-simbol yang ada pada diagram kelas :

Tabel.II.7 Simbol-simbol *Class Diagram*

Simbol	Deskripsi			
<p>Kelas</p> <table border="1"> <tr> <td>Nama_kelas</td> </tr> <tr> <td>+atribut</td> </tr> <tr> <td>-operasi()</td> </tr> </table>	Nama_kelas	+atribut	-operasi()	Kelas pada struktur sistem
Nama_kelas				
+atribut				
-operasi()				
<p>Antarmuka / <i>interface</i></p>  <p>nama_interface</p>	Sama dengan konsep <i>interface</i> dalam pemrograman berorientasi objek			
<p>Asosiasi / <i>association</i></p> 	Relasi antar kelas dengan makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i>			
<p>Asosiasi berarah / <i>directed association</i></p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i>			
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum khusus)			
<p>Kebergantungan / <i>dependency</i></p> 	Ketergantungan antar kelas			
<p>Agregasi / <i>aggregation</i></p> 	Relasi antar kelas dengan makna semua-bagian (<i>whole-part</i>)			

(Rosa A.S,dkk:2014:146-147)

Contoh Class Diagram



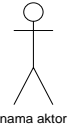
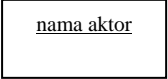

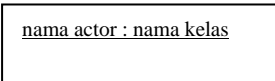

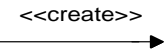
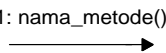
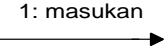
Gambar II.8 Contoh Class Diagram

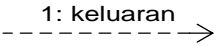
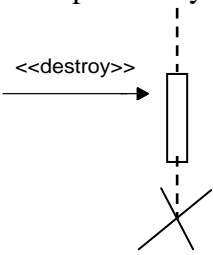
3. Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan *message* yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah *use case* beserta metode-metode yang dimiliki kelas yang diinstansi menjadi objek itu. Membuat diagram sekuen juga dibutuhkan melihat skenario yang ada pada *use case*. (Rosa A.S,dkk:2014:165-167)

Berikut adalah symbol-simbol yang ada pada diagram sekuen :

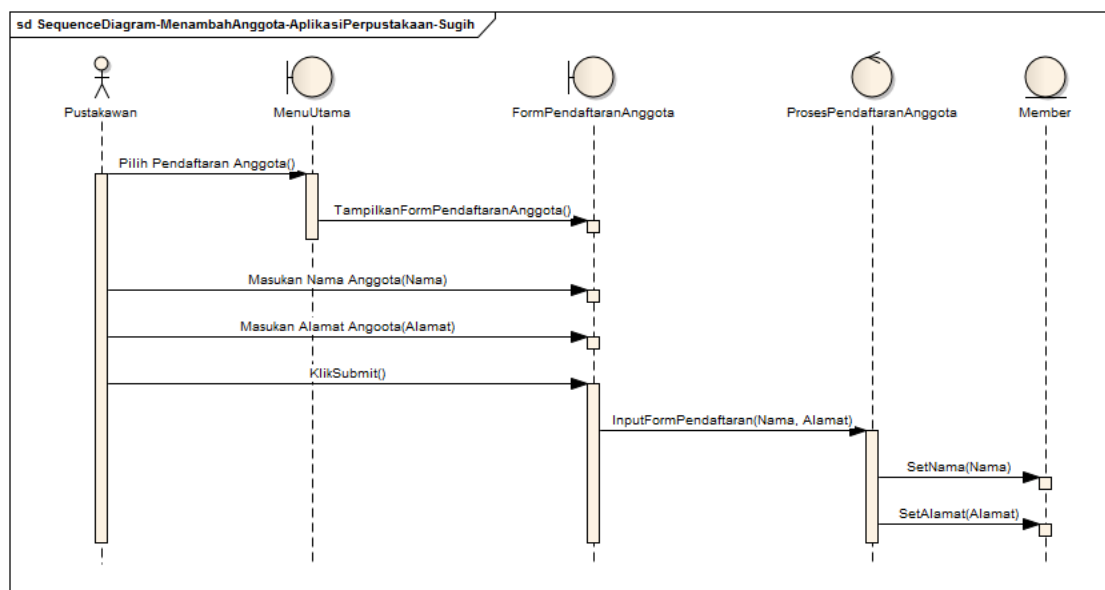
Tabel. II.8 Simbol-simbol *Sequence Diagram*

Simbol	Deskripsi
<p>Actor</p>  <p>atau</p>  <p>tanpa waktu aktif</p>	<p>Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun symbol dari actor adalah gambar orang, tapi actor belum tentu merupakan orang; biasanya dinyatakan menggunakan kata benda di awal frase nama aktor</p>
<p>Garis hidup / <i>lifeline</i></p> 	<p>Menyatakan kehidupan suatu objek</p>
<p>Objek</p> 	<p>Menyatakan objek yang berinteraksi pesan</p>
<p>Waktu aktif</p> 	<p>Menyatakan objek dalam keadaan aktif dan beriteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya</p>
<p>Pesan tipe create</p> 	<p>Menyatakan suatu objek membuat objek yang lain, arah panah mengarah pada objek yang dibuat</p>
<p>Pesan tipe call</p> 	<p>Menyatakan suatu objek memanggil operasi / metode yang ada pada objek lain atau dirinya sendiri,</p>
<p>Pesan tipe send</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>

<p>Pesan tipe return</p> 	<p>Menyatakan bahwa suatu objek mengirimkan data/masukan/informasi ke objek lainnya, arah panah mengarah pada objek yang dikirim</p>
<p>Pesan tipe destroy</p> 	<p>Menyatakan suatu objek mengakhiri objek yang lain, arah panah mengarah pada objek yang diakhiri, sebaiknya jika ada create maka ada destroy</p>

(Rosa A.S,dkk:2014:166-167)

Contoh *sequence diagram*





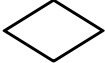


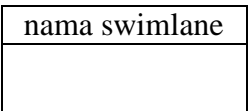
Gambar. II.9 Contoh *Sequence Diagram*

4. Activity Diagram

Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Yang perlu diperhatikan disini adalah bukan apa yang dilakukan actor, jadi aktivitas yang dapat dilakukan oleh sistem. (Rosa A.S, dkk:2014:161-163)

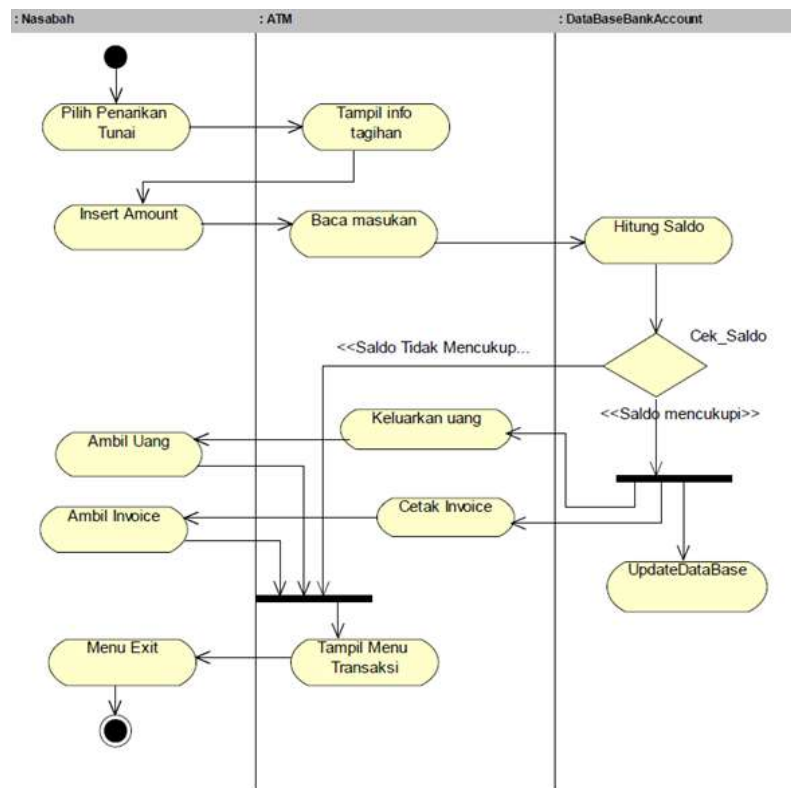
Berikut adalah simbol-simbol yang ada pada diagram aktivitas:

Tabel. II. 9 Simbol-simbol Activity Diagram

Simbol	Deskripsi
Status awal 	Status awal aktivitas 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 aktifitas digabungkan menjadi satu
Status akhir 	Status akhir yang dilakukan sistem, sebuah diagram aktifitas memiliki sebuah status akhir
Swimlane 	Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi

(Rosa A.S, dkk:2014:162)

Contoh Activity Diagram



Gambar II. 10 Contoh Activity Diagram