

BAB II

TINJAUAN PUSTAKA

II.1 Sistem

Mempelajari suatu sistem akan lebih mudah mengenal bila mengetahui terlebih dahulu apakah sistem itu. pengertian tentang sistem pertama kali dapat diperoleh dari definisi sistem itu sendiri. jika kita perhatikan dengan seksama, diri kita juga terdiri dari berbagai sistem yang berfungsi untuk mengantar kita kepada tujuan hidup kita. sudah banyak ahli yang mengungkapkan berbagai sistem yang bekerja dalam diri manusia, misalnya sistem kekebalan tubuh untuk menghadapi penyakit cacar dan diptheri. namun masih banyak pula berbagai sistem yang belum dapat diungkapkan dengan teknologi yang sekarang telah dimiliki oleh manusia, misalnya sistem kekebalan tubuh untuk menghadapi penyakit AIDS. contoh sistem lain dalam diri manusia adalah sistem pernapasan, yang berfungsi untuk menyediakan oksigen bagi tubuh dan mengeluarkan zat asam arang yang merupakan sampah hasil pembakaran didalam tubuh. sistem pernapasan terdiri dari unsur - unsur yang membentuk struktur sistem pernapasan dan setiap unsur struktur sistem tersebut bekerja dengan suatu pola tertentu untuk memenuhi tujuan sistem seperti yang telah disebutkan diatas.

Pendekatan sistem memberikan banyak manfaat dalam memahami lingkungan. Pendekatan sistem berusaha untuk menjelaskan sesuatu yang dipandang dari sudut pandang sistem serta berusaha menemukan struktur sistem yang membentuk sistem tersebut. Dengan memahami struktur sistem proses

sistem, seorang akan dapat menjelaskan mengapa tujuan suatu sistem tidak tercapai.

Dari uraian diatas, dapat diambil kesimpulan bahwa suatu sistem pada dasarnya adalah sekelompok unsur yang erat hubungannya satu dengan yang lain, yang berfungsi bersama-sama untuk mencapai tujuan tertentu (Sutabri, 2012:4).

II.2 Subsistem

Suatu sistem terdiri dari bagian-bagian sistem atau subsistem. Seperti sistem komputer dapat terdiri dari subsistem perangkat keras dan subsistem perangkat lunak. Masing-masing subsistem terdiri dari subsistem-subsistem lagi atau terdiri dari komponen-komponen pendukung sistem itu sendiri. Subsistem perangkat keras dapat terdiri dari alat masukan, alat pemroses dan alat keluaran serta media penyimpanan. Subsistem-subsistem itu saling terkait dan saling berhubungan untuk membentuk satu kesatuan sehingga tujuan atau sasaran sistem tersebut tercapai. Subsistem tersebut terintegrasi sedemikian rupa sehingga tercapai satu kesatuan yang terpadu dan terintegrasi.

Konsep sebuah sistem menuntut perancangannya untuk mempertimbangkan sistem sebagai suatu keseluruhan. Akan tetapi keseluruhan sistem mungkin terlalu besar untuk dianalisis secara terperinci. Oleh karena itu sistem dibagi atau diuraikan atas beberapa bagian subsistem.

Norman L. Enger mengatakan dalam bukunya bahwa subsistem adalah serangkaian kegiatan yang dapat ditentukan identitasnya yang berhubungan dalam sesuatu sistem. Batasan dan penghubung atau antarmuka di dalam satu sistem ditelaah secara cermat untuk menjamin bahwa hubungan hubungan

antarsubsystem didefinisikan secara jelas dan bahwa jumlah semua subsystem merupakan keseluruhan sistem. Proses menjadikan faktor-faktor suatu sistem ke subsystem-subsystem merupakan hal yang penting guna menyederhakan masalah. Sistem yang menentukan sederhana adalah sistem yang mempunyai sedikit subsystem dan antarmubungan, yang menunjukkan secara lengkap perilaku yang dapat diramalkan. Subsystem yang sederhana juga dapat bersifat memungkinkan. Misalnya suatu sistem pengendalian kualitas yang meramalkan banyaknya kerusakan dalam suatu populasi merupakan suatu sistem yang sederhana, tetapi juga bersifat memungkinkan.

Penggunaan subsystem dapat juga disebut konsep modular. Konsep subsystem digunakan dalam manajemen proyek dan juga dalam perancangan sistem. Pembatasan subsystem penting bagi pemeliharaan sistem. Apabila suatu subsystem mempunyai suatu perubahan atau pembetulan dapat dilakukan dengan mudah daripada apabila subsystem tersebut disimpan dalam proses yang lebih besar (Sutabri, 2012:9).

II.3 Karakteristik Sistem

Model umum sebuah sistem terdiri dari input, proses, dan output. hal ini merupakan konsep sebuah sistem yang sangat sederhana mengingat sebuah sistem dapat mempunyai beberapa masukan dan keluaran sekaligus. Selain itu, sebuah sistem juga memiliki karakteristik atau sifat-sifat tertentu, yang mencirikan bahwa hal tersebut bisa dikatakan sebagai suatu sistem. Adapun karakteristik yang dimaksud adalah sebagai berikut:

1. Komponen Sistem

Suatu sistem terdiri dari sejumlah komponen yang saling berinteraksi, yang bekerja sama membentuk satu kesatuan. Komponen-komponen sistem tersebut dapat berupa suatu bentuk subsistem. Setiap subsistem memiliki sifat-sifat sistem yang menjalankan suatu fungsi tertentu dan mempengaruhi proses sistem secara keseluruhan. Satu sistem dapat mempunyai sistem yang lebih besar yang disebut dengan supra-sistem.

2. Batasan Sistem

Ruang lingkup suatu sistem merupakan daerah yang membatasi antara sistem dengan sistem yang lainnya atau sistem dengan lingkungan luarnya. Batasan sistem ini memungkinkan suatu sistem dipandang sebagai satu kesatuan yang tidak dapat dipisah-pisahkan.

3. Lingkungan Luar Sistem

Bentuk apapun yang ada diluar ruang lingkup atau batasan sistem yang mempengaruhi operasi sistem tersebut dengan lingkungan luar sistem. Lingkungan luar sistem yang menguntungkan bagi sistem tersebut merupakan energi bagi sistem, yang dengan demikian lingkungan luar tersebut harus selalu dijaga dan dipelihara. Sedangkan lingkungan luar sistem yang merugikan harus dikendalikan karena akan mengganggu kelangsungan hidup sistem tersebut.

4. Penghubung Sistem (Antarmuka)

Media yang menghubungkan sistem dengan subsistem yang lain disebut dengan penghubung sistem atau antarmuka. Penghubung ini memungkinkan sumber-sumber daya mengalir dari satu subsistem yang lain. Keluaran dari

subsistem akan menjadi masukan untuk subsistem yang lain dengan melewati penghubung. Dengan demikian terjadi suatu integrasi yang membentuk satu kesatuan.

5. Masukan Sistem

Energi yang dimasukkan ke dalam sistem disebut sebagai masukan sistem, yang dapat berupa pemeliharaan dan sinyal. Sebagai contoh, di dalam satu unit sistem komputer, program adalah masukan pemeliharaan yang digunakan untuk mengoperasikan komputer. Sementara data adalah sinyal masukan yang akan diolah menjadi informasi.

6. Keluaran Sistem

Hasil dari energi yang diolah dan diklarifikasikan menjadi keluaran yang berguna. Keluaran ini merupakan masukan bagi subsistem-subsistem yang lain. Seperti subsistem informasi, keluaran yang dihasilkan adalah informasi, dimana informasi ini dapat digunakan sebagai masukan untuk pengambilan keputusan atau hal-hal lain yang merupakan input bagi subsistem lainnya.

7. Pengolah Sistem

Suatu sistem dapat mempunyai suatu proses yang akan mengubah masukan menjadi keluaran. Sebagai contoh, Sistem Akuntansi. Sistem ini akan mengolah data transaksi menjadi laporan-laporan yang dibutuhkan oleh pihak manajemen.

8. Sasaran Sistem

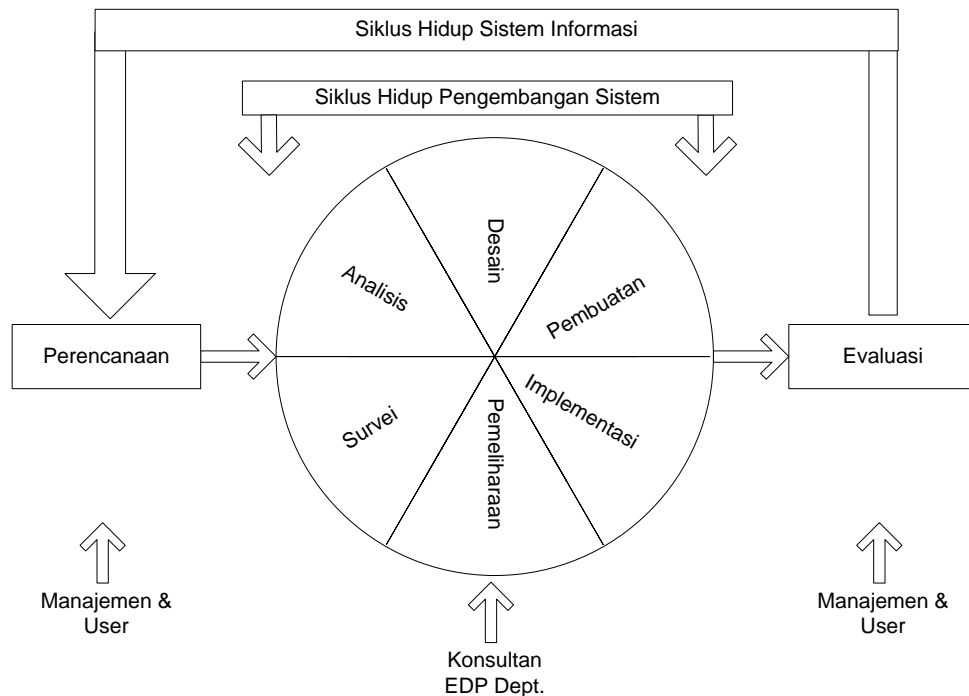
Suatu sistem memiliki suatu tujuan atau sasaran yang pasti dan bersifat deterministik. Kalau sistem tidak memiliki sasaran maka operasi sistem tidak

ada gunanya. Suatu sistem dikatakan berhasil jika mengenai sasaran atau tujuan yang telah direncanakan (Sutabri, 2012:13).

II.4 *System Development Life Cycle (SDLC)*

Siklus hidup pengembangan sistem merupakan suatu bentuk yang digunakan untuk menggambarkan tahapan utama dan langkah-langkah pada tahapan tersebut dalam proses pengembangan sistem. Siklus hidup pembangunan atau pengembangan sistem informasi menyajikan metodologi atau proses yang diorganisasikan guna membangun suatu sistem informasi. Dengan demikian, dalam membangun suatu sistem informasi, maka sejumlah tugas harus diselesaikan. Beberapa tugas harus dilaksanakan dalam suatu urutan tertentu. Sejumlah personil terlibat. Usaha-usaha mereka harus dikoordinasikan untuk mencapai suatu efektifitas dan efesiensi yang tinggi.

Siklus hidup sistem informasi dimulai dari fase perencanaan, fase pengembangan (investigasi, analisis, desain, implementasi) dan dievaluasi secara terus-menerus untuk menetapkan apakah sistem informasi tersebut masih layak diaplikasikan. Jika tidak, maka sistem informasi tersebut akan diganti dengan yang baru dan dimulai dari perencanaan kembali, seperti pada gambar II.1 di bawah ini:



Gambar II.1 Siklus Hidup Pengembangan Sistem
Sumber: Sutabri, 2012:57

II.5 Sistem Informasi

Sistem informasi bukan merupakan hal yang baru, yang baru adalah komputerisasinya. Sebelum ada komputer, teknik penyaluran informasi yang memungkinkan manajer merencanakan serta mengendalikan operasi yang ada. Komputer menambahkan satu atau dua dimensi, seperti kecepatan, ketelitian dan penyediaan data dengan volume yang lebih besar yang memberikan bahan pertimbangan yang lebih banyak untuk mengambil keputusan.

Suatu organisasi terdiri atas sejumlah unsur, orang-orang yang mempunyai berbagai peran, kegiatan atau tugas yang harus diselesaikan, tempat kerja, wewenang, serta hubungan komunikasi yang mengikat organisasi tersebut. Sistem informasi merupakan penerapan sistem di dalam organisasi untuk mendukung informasi yang dibutuhkan oleh semua tingkat manajemen. Telah diketahui bahwa

informasi merupakan yang sangat penting bagi manajemen di dalam pengambilan keputusan. Informasi dapat diperoleh dari sistem informasi.

Sistem informasi adalah suatu sistem di dalam suatu organisasi yang mempertemukan kebutuhan pengolahan transaksi harian yang mendukung fungsi operasi organisasi yang bersifat manajerial dengan kegiatan strategi dari suatu organisasi untuk mendapatkan penyediaan laporan-laporan yang diperlukan oleh pihak luar tertentu (Sutabri, 2012:38).

II.6 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) merupakan sistem informasi interaktif yang menyediakan informasi, pemodelan, dan pemanipulasian data. Sistem itu digunakan untuk membatu pengambilan keputusan dalam situasi yang semiterstruktur dan situasi yang tidak terstruktur, dimana tidak seorang pun tahu secara pasti bagaimana keputusan seharusnya dibuat. DSS biasanya dibangun untuk mendukung solusi atau suatu masalah atau untuk mengevaluasi suatu peluang. DSS yang seperti itu disebut aplikasi DSS. Aplikasi DSS digunakan dalam mengambil keputusan. Aplikasi DSS menggunakan CBIS (*Computer Based Infomation System*) 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 analisis dalam situasi yang kurang terstruktur dan dengan kriteria yang

kurang jelas. DSS dimaksudkan untuk mengotomatisasikan pengambilan keputusan, tetapi memberikan perangkat interaktif yang memungkinkan pengambilan keputusan untuk melakukan berbagai analisis menggunakan model-model yang tersedia (Kusrini, 2007: 15).

II.6.1. Arsitektur Sistem Pendukung Keputusan

Aplikasi sistem pendukung keputusan bisa terdiri dari beberapa subsistem, yaitu:

1. Subsistem manajemen data

Subsistem manajemen data memasukkan suatu basis data yang berisi data yang relevan untuk suatu situasi dan dikelola oleh perangkat lunak yang disebut Sistem Manajemen Basis Data. Subsistem manajemen data bisa diinterkoneksi dengan data *warehouse* perusahaan, suatu repositori untuk data perusahaan yang relevan dengan pengambilan keputusan.

2. Subsistem manajemen model

Merupakan paket perangkat lunak yang memasukkan model keuangan, statistik, ilmu manajemen, atau model kuantitatif lain yang memberikan kapabilitas analitik dan manajemen perangkat lunak yang tepat. Bahasa-bahasa pemodelan untuk membangun model-model kustom juga dimasukkan. Perangkat lunak itu sering disebut sistem manajemen basis model. Komponen tersebut bisa dikoneksikan ke penyimpanan korporat atau eksternal yang ada pada model.

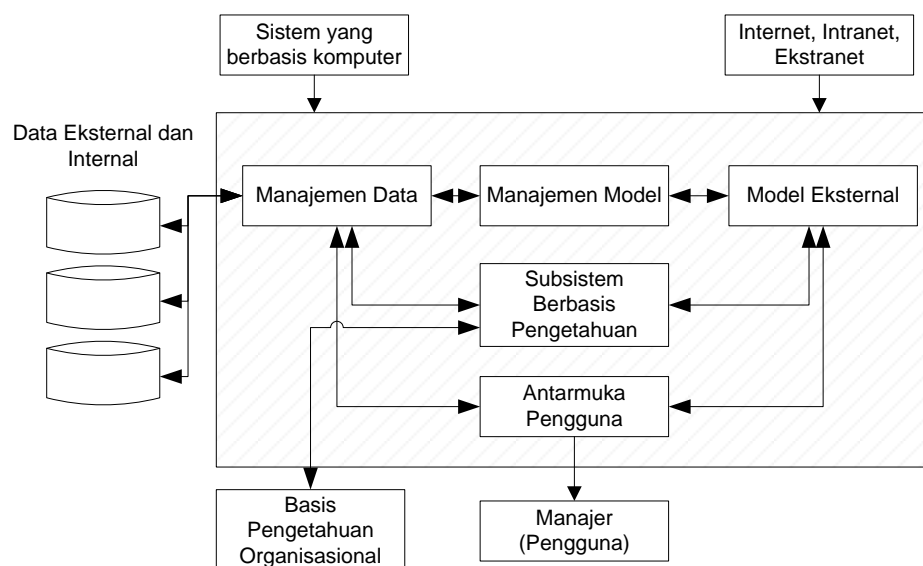
3. Subsistem antarmuka pengguna

Pengguna berkomunikasi dengan dan memerintahkan sistem pendukung keputusan melalui subsistem tersebut. Pengguna adalah bagian yang dipertimbangkan dari sistem. Para peneliti menegaskan bahwa beberapa kontribusi unik dari sistem pendukung keputusan berasal dari interaksi yang intensif antara komputer dan pembuat keputusan.

4. Subsistem manajemen berbasis-pengetahuan

Subsistem tersebut mendukung semua subsistem lain atau bertindak secara langsung sebagai suatu komponen independen dan bersifat opsional.

Berdasarkan definisi, sistem pendukung keputusan harus mencakup tiga komponen utama dari DBMS, MBMS, dan antarmuka pengguna. Arsitektur dari sistem pendukung keputusan ditunjukkan dalam gambar II.2 berikut:



Gambar II.2 Arsitektur SPK
Sumber: Kusri, 2007:26

II.7 Model *Analytical Hierarchy Process* (AHP)

Pada dasarnya, proses pengambilan keputusan adalah memilih suatu alternatif. Peralatan utama AHP adalah sebuah hierarki fungsional dengan input

utamanya persepsi manusia. Keberadaan hierarki memungkinkan dipecahnya masalah kompleks atau tidak terstruktur dalam sub-sub masalah, lalu menyusunnya menjadi suatu bentuk hierarki.

AHP juga mempunyai beberapa kelebihan dan kelemahan hasil penerapannya:

1. Kelebihan AHP

- a. Mampu membahas permasalahan yang kompleks dan tidak terstruktur secara detil.
- b. Memiliki kemampuan dalam melakukan sintesa pemikiran berbagai sudut pandang responden.
- c. Memperhitungkan konsistensi dai penilaian yang telah dilakukan dalam memperbandingkan faktor–faktor untuk memvalidasi.
- d. Memungkinkan untuk melakukan perencanaan kedepan (*forward*) atau sebaliknya, menjabarkan masa depan yang ingin dicapai kemas kini (*backward*).

2. Kelemahan AHP

- a. Dalam penerapannya harus melibatkan orang–orang yang memiliki pengetahuan yang cukup tentang permasalahan dan metode AHP sendiri.
- b. Untuk melakukan perbaikan keputusan, walaupun kecil maka harus dimulai lagi dari tahap awal dan memakan waktu yang relatif lama.
- c. AHP tidak dapat diterapkan pada suatu perbedaan sudut pandang yang sangat ekstrim di kalangan responden.

AHP memiliki banyak keunggulan dalam menjelaskan proses pengambilan keputusan. Salah satunya adalah dapat digambarkan secara grafis sehingga dapat dengan mudah dipahami oleh pihak yang terlibat dalam pengambilan keputusan (Kusrini, 2007:133).

II.7.1. Prinsip Dasar AHP

Dalam menyelesaikan permasalahan dengan AHP ada beberapa prinsip yang harus dipahami, diantaranya adalah:

1. Membuat hierarki

Sistem yang kompleks bisa dipahami dengan memecahkan menjadi elemen-elemen pendukung, menyusun elemen secara hirarki, dan menggabungkannya atau mensintesisnya.

2. Penilaian kriteria dan alternatif

Penilaian kriteria dan alternatif dilakukan dengan perbandingan berpasangan. Menurut Saaty (1988), untuk berbagai persoalan, skala 1-9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan defenisi pendapat kualitatif dari skala perbandingan Saaty bisa diukur dengan menggunakan tabel analisis seperti ditunjukkan pada tabel II.1 berikut:

Tabel II.1 Skala Penilaian Perbandingan Pasangan

Intensitas Kepentingan	Keterangan
1	Kedua elemen sama pentingnya ²
3	Elemen yang satu sedikit lebih penting daripada elemen lainnya.
5	Elemen yang satu lebih penting daripada elemen lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya.
9	Satu elemen mutlak penting daripada elemen lainnya.
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan.

Kebalikan	Jika aktivitas i mendapat satu angka dibandingkan dengan aktivitas j , maka j memiliki nilai kebalikannya dibandingkan dengan i .
-----------	---

Sumber: Kusrini, 2007:134

3. Menentukan Prioritas

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan (*pairwise comparison*). Nilai-nilai perbandingan relatif dari seluruh alternatif kriteria bisa disesuaikan dengan *judgement* yang telah ditentukan untuk menghasilkan bobot dan prioritas. Bobot dan prioritas dihitung dengan memanipulasi matriks atau melalui penyelesaian persamaan matematika.

4. Konsistensi Logis

Konsistensi memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut tingkat hubungan antarobjek yang didasarkan pada kriteria tertentu (Kusrini, 2007:133).

II.7.2. Prosedur AHP

Pada dasarnya, prosedur atau langkah-langkah dalam metode AHP meliputi:

1. Mendefinisikan masalah dan menentukan solusi yang diinginkan, lalu menyusun hirarki dari permasalahan yang dihadapi. Penyusunan hirarki adalah dengan menetapkan tujuan yang merupakan sasaran sistem secara keseluruhan pada level teratas.
2. Menentukan prioritas elemen

- a. Langkah pertama dalam menentukan prioritas elemen adalah dengan membuat perbandingan berpasangan, yaitu membandingkan elemen secara berpasangan sesuai dengan kriteria yang diberikan.
- b. Matriks perbandingan berpasangan diisi menggunakan bilangan untuk merepresetasikan kepentingan relatif dari suatu elemen terhadap elemen yang lain.

3. Sintesis

Pertimbangan-pertimbangan terhadap perbandingan berpasangan disintesis untuk memperoleh keseluruhan prioritas. Hal-hal yang dilakukan dalam langkah ini adalah:

- a. Menjumlahkan nilai-nilai dari setiap kolom pada matriks.
- b. Membagi setiap nilai dari kolom dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks.
- c. Menjumlahkan nilai-nilai dari setiap baris dan membaginya dengan jumlah elemen untuk mendapatkan nilai rata-rata.

4. Mengukur konsistensi

Dalam pembuatan keputusan, penting untuk mengetahui seberapa baik konsistensi yang ada karena kita tidak menginginkan keputusan berdasarkan pertimbangan dengan konsistensi yang rendah. Hal-hal yang dilakukan dalam langkah ini adalah:

- a. Kalikan setiap nilai pada kolom pertama dengan prioritas relatif elemen pertama, nilai pada kolom kedua dengan prioritas relatif elemen kedua, dan seterusnya.

- b. Jumlahkan setiap baris.
 - c. Hasil dari penjumlahan baris dibagi dengan elemen prioritas relatif yang bersangkutan.
 - d. Jumlahkan hasil bagi diatas dengan banyaknya elemen yang ada, hasilnya disebut lamda maksimum.
5. Hitung Consistency Index (CI) dengan rumus:
- $$CI = (\lambda_{\max} - n) / n$$
- Dimana n = banyaknya elemen.
6. Hitung Ratio Consistency (CR) dengan rumus:
- $$CR = CI / IR$$
- Dimana CR = Consistency Ratio
- CI = Consistency Index
- IR = Index Random Consistency
7. Memeriksa konsistensi hirarki. Jika nilainya lebih dari 10%, maka nilai data *judgment* harus diperbaiki. Namun jika Ratio Consistency (CI/IR) kurang atau sama dengan 0.1, maka hasil perhitungan bisa dinyatakan benar (Kusrini, 2007:135).

Daftar Indeks Random Konsistensi (IR) bisa dilihat pada tabel II.2 dibawah ini:

Tabel II.2 Daftar Indeks Random Konsistensi

Ukuran Matriks	Nilai IR
1,2	0.00
3	0.58
4	0.90

5	1.12
6	1.24
7	1.32
8	1.41
9	1.45
10	1.49
11	1.51
12	1.48
13	1.56
14	1.57
15	1.59

Sumber: Kusri, 2007:136

II.8 Bahasa Pemrograman Java

Menurut Hariyanto (2001:1), dalam *Esensi-Esensi Bahasa Java*, bahasa pemrograman Java merupakan karya Sun Microsystems Inc. Rilis resmi level *beta* dilakukan pada November 1995. Dua bulan berikutnya Netscape menjadi perusahaan pertama yang memperoleh lisensi bahasa Java dari Sun. Java adalah bahasa pemrograman berorientasi objek yang berukuran kecil, sederhana dan aman, diinterpretasi atau dioptimalisasi secara dinamis, ber-*bytecode*, arsitektur yang netral, mempunyai *garbage-collector*, *multithreading*, mempunyai mekanisme penanganan pengecualian, berbasis tipe untuk penulisan program mudah diperluas dinamis serta telah diperuntukan sistem tersebar.

Pada awal pembuatannya, Java dinamakan Oak, kemudian nama Oak dinilai kurang menjual sehingga pada Januari 1995 nama Oak diubah menjadi Java. Sebagai bahasa yang bersifat terbuka, Java didukung oleh banyak programmer dari seluruh dunia yang memberikan kontribusinya untuk mengembangkan bahasa Java.

II.8.1. *The Java 2 Platform Standard Edition (J2SE)*

The Java 2 Platform Standard Edition (J2SE) menyediakan lingkungan pengembangan yang kaya fitur, stabil, aman, dan *cross-platform*. Edisi ini mendukung konektivitas basis data, rancangan antarmuka pemakai, masukan/keluaran, dan pemrograman jaringan dan termasuk sebagai paket-paket dasar bahasa Java (Hariyanto, 2011:3).

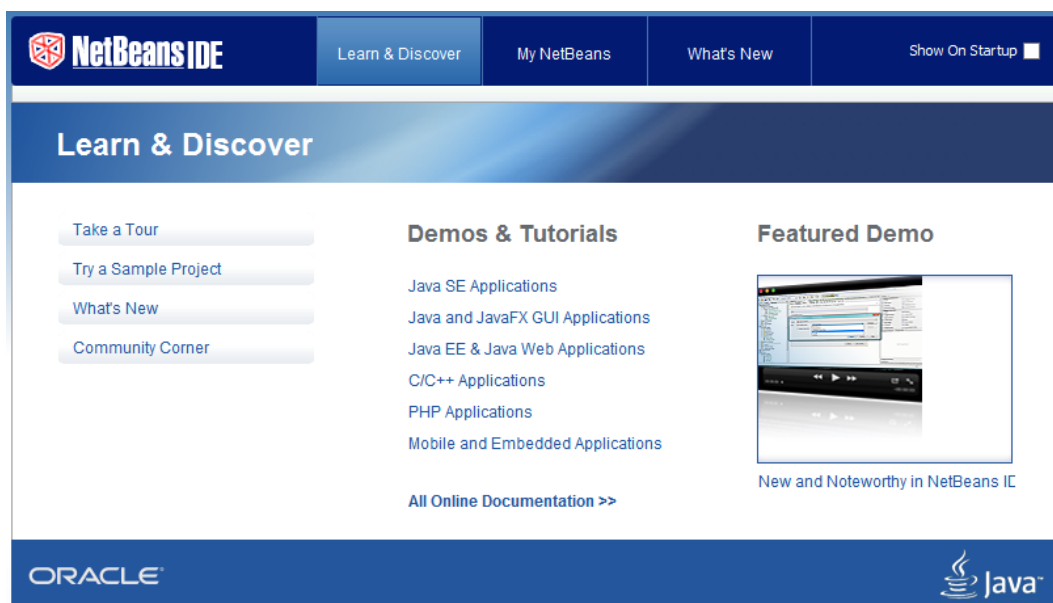
II.9. Perangkat Pengembangan Terintegrasi Netbeans 7.3

NetBeans merupakan sebuah proyek kode terbuka yang sukses dengan pengguna yang sangat luas, komunitas yang terus tumbuh, dan memiliki hampir 100 mitra (dan terus bertambah). Sun Microsystems mendirikan proyek kode terbuka NetBeans pada bulan Juni 2000 dan terus menjadi sponsor utama.

Saat ini terdapat dua produk, yaitu NetBeans IDE dan NetBeans *Platform*. NetBeans IDE adalah sebuah lingkungan pengembangan - sebuah kakas untuk pemrogram menulis, mengompilasi, mencari kesalahan dan menyebarkan program. Netbeans IDE ditulis dalam Java - namun dapat mendukung bahasa pemrograman lain. Terdapat banyak modul untuk memperluas Netbeans IDE. Netbeans IDE adalah sebuah produk bebas dengan tanpa batasan bagaimana digunakan. Tersedia juga NetBeans *Platform*; sebuah fondasi yang modular dan dapat diperluas yang dapat digunakan sebagai perangkat lunak dasar untuk membuat aplikasi desktop yang besar. Mitra ISV menyediakan plug-in bernilai tambah yang dapat dengan mudah diintegrasikan ke dalam Platform dan dapat juga digunakan untuk membuat kakas dan solusi sendiri.

Kedua produk adalah kode terbuka (*open source*) dan bebas (*free*) untuk penggunaan komersial dan non komersial. Kode sumber tersedia untuk guna ulang dengan lisensi *Common Development and Distribution License* (CDDL) (http://netbeans.org/index_id.html).

Tampilan awal Netbeans 7.3 terlihat seperti gambar II.3 di bawah ini:



Gambar II.3 Tampilan Awal Netbeans 7.3

Sumber: www.netbeans.org

Fitur-fitur terbaru yang tersedia pada Netbeans 7.3 untuk Java adalah sebagai berikut:

1. Fitur Breadcrumbs sebagai bar navigasi untuk menunjukkan area kode yang sedang aktif pada *editor* dan ditampilkan juga pada tampilan hirarki.
2. Sekumpulan petunjuk baru dan kemampuan *refactoring*.
3. Penyaringan untuk hasil fitur *Find Usages*.
4. Penambahan pengedit berkas xml POM yang efektif pada *tab* baru.
5. Penambahan peralatan pengujian Java *Persistence JPQL*.
6. Pengembangan tambahan untuk layanan REST.

7. Mendukung FXML/SceneBuilder untuk JavaFX.
8. Mendukung JDK 7 *Update* 6 pada Windows, Mac, dan Linux.
9. Pengeditan FXML sangat ditingkatkan dengan menyediakan pelengkapan kode yang akurat, penanda kesalahan dan lain-lain.
10. Penambahan pembangunan aksi *Controller* pada berkas FXML.

II.10 Basis Data

Menurut Kusrini (2007:2) dalam bukunya *Strategi Perancangan dan Pengelolaan Basis Data*, basis data adalah kumpulan data yang saling berelasi. Data sendiri merupakan fakta mengenai obyek, orang, dan lain-lain. Data dinyatakan dengan nilai baik berupa angka, deretan karakter maupun simbol.

Basis data dapat didefinisikan dalam berbagai sudut pandang seperti berikut:

1. Himpunan kelompok data yang saling berhubungan yang diorganisasikan sedemikian rupa sehingga kelak dapat dimanfaatkan dengan cepat dan mudah.
2. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa tanpa pengulangan atau redundansi yang tidak perlu, untuk memenuhi kebutuhan.
3. Kumpulan berkas/tabel/arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

II.10.1. Arsitektur Basis Data

Menurut Kusrini (2007:15) dalam bukunya *Strategi Perancangan dan Pengelolaan Basis Data*, arsitektur basis data memberikan kerangka kerja bagi para pengembang sistem basis data.

Ada tiga level arsitektur sistem basis data, yaitu:

1. Level Fisik/Internal

Level fisik merupakan level paling rendah yang menggambarkan bagaimana data disimpan secara fisik.

2. Level Konseptual/Logika

Level ini menggambarkan data apa yang disimpan dalam basis data dan hubungan relasi yang terjadi antara data dari keseluruhan basis data. Level ini memperhatikan data apa sebenarnya secara fungsional disimpan dalam basis data dan hubungannya dengan data lain. Pemakai tidak memperdulikan kerumitan dalam struktur level fisik lagi, penggambaran cukup dengan memakai kotak, garis, dan hubungan secukupnya. Pada level ini biasanya perancang basis data membuat rancangan dengan bentuk diagram.

3. Level Pandangna Pengguna/Eksternal

Level ini merupakan level abstraksi data tertinggi yang menggambarkan hanya sebagian saja yang dilihat dan dipakai dari keseluruhan basis data. Hal ini disebabkan beberapa pengguna basis data tidak membutuhkan semua isi basis data. Pengguna yang dimaksud adalah pemrogram, pengguna akhir, atau administrator basis data.

II.10.2. Entity Relationship Diagram (ERD)

Menurut Kusrini (2007:21) dalam bukunya *Strategi Perancangan dan Pengelolaan Basis Data*, perancangan basis data dengan menggunakan model relasi entitas adalah dengan menggunakan *Entity Relationship Diagram (ERD)*.

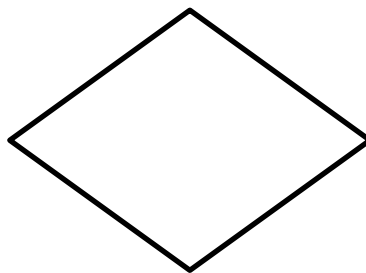
Terdapat tiga notasi dasar yang bekerja pada model E-R, yaitu *Entity Sets*, *Relationship Sets*, dan *Attributes*.

Entity adalah sebuah benda atau objek di dunia nyata yang dapat dibedakan dari semua objek lainnya. *Entity Sets* adalah sekumpulan entitas yang mempunyai tipe yang sama. Kesamaan tipe ini dapat dilihat dari atribut atau properti yang dimiliki oleh setiap entitas. *Entity Set* dilambangkan dengan bentuk persegi panjang seperti tampak pada gambar II.4 berikut:



Gambar II.4 Notasi *Entity Set*
Sumber: Kusrini, 2007:21

Relationship adalah hubungan diantara beberapa entitas. *Relationship Set* adalah sekumpulan relasi yang mempunyai tipe yang sama. *Relationship Set* digambarkan dengan belah ketupat seperti tampak pada gambar II.5 di bawah ini:



Gambar II.5 Notasi *Relationship*
Sumber: Kusrini, 2007:21

Attributes adalah sekumpulan properti yang dimiliki oleh suatu entitas. *Attributes* digambarkan dengan bentuk elips seperti terlihat pada gambar II.6 di bawah ini:



Gambar II.6 Notasi *Attributes*
Sumber: Kusrini, 2007:22

II.10.3. Kamus Data

Menurut Kusri (2007:29) dalam bukunya *Strategi Perancangan dan Pengelolaan Basis Data*, kamus data adalah suatu daftar data elemen yang terorganisir dengan definisi yang tetap dan sesuai dengan sistem, sehingga *user* dan analis sistem mempunyai pengertian yang sama tentang *input*, *output*, dan komponen *data store*.

Kamus data ini sangat membantu analis sistem dalam mendefinisikan data yang mengalir di dalam sistem, sehingga pendefinisian data itu dapat dilakukan dengan lengkap dan terstruktur. Pembentukan kamus data dilaksanakan dalam tahap analisis dan perancangan suatu sistem.

Pada tahap analisis, kamus data merupakan alat komunikasi antara *user* dan analis sistem tentang data yang mengalir di dalam sistem, yaitu tentang data yang masuk ke sistem dan tentang informasi yang dibutuhkan oleh *user*. Sementara itu, pada tahap perancangan sistem kamus data digunakan untuk merancang *input*, laporan dan *database*.

II.10.4 Normalisasi

Proses normalisasi merupakan proses pengelompokan elemen data menjadi tabel-tabel yang menunjukkan entitas dan relasinya. Proses ini selalu diuji pada beberapa kondisi seperti menambah, menghapus, mengubah, atau membaca. Bila terdapat kesulitan pada pengujian tersebut maka relasi dapat dipecah dalam beberapa tabel lagi. Pada normalisasi, proses dibagi menjadi beberapa tahap sebagai berikut:

1. Bentuk Tidak Normal

Bentuk ini merupakan kumpulan data yang akan direkam, tidak ada keharusan untuk mengikti suatu format tertentu, dapat saja data tidak lengkap atau terduplikasi. Data dikumpulkan apa adanya sesuai dengan kedatagannya.

2. Bentuk Normal Kesatu (1NF)

Bentuk normal kesatu mempunyai ciri yaitu setiap data dibentuk dalam berkas datar, data dibentuk dalam satu rekaman demi rekaman dan nilai dari kolom berupa nilai atomik. Tidak dibentuk set atribut yang berulang atau atribut bernilai ganda, tiap kolom hanya mempunyai satu pengertian, bukan merupakan kumpulan kata yang mempunyai arti ganda.

3. 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. Dengan demikian untuk membentuk normal kedua haruslah sudah ditentukan kunci utama.

4. Bentuk Normal Ketiga (3NF)

Untuk menjadi bentuk normal ketiga maka relasi harus dalam bentuk normal kedua dan semua atribut bukan primer tidak punya hubungan yang transitif. Dengan kata lain, setiap atribut bukan kunci harus bergantung hanya pada kunci utama.

5. Bentuk Normal *Boyce-Codd* (BCNF)

Bentuk *Boyce-Codd* mempunyai paksaan yang lebih kuat dari bentuk normal ketiga. Untuk menjadi BCNF, relasi harus dalam bentuk normal kesatu dan setiap atribut harus bergantung pada atribut kunci super (Sutabri, 2012:138).

II.11 Basis Data MySQL

Menurut Sutaji (2012 : 40), MySQL adalah DMBS yang didistribusikan secara gratis dibawah lisensi dari *General Public Lisence* (GPL), dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh untuk dijadikan program induk turunan yang bersifat tertutup atau komersial.

MySQL sebenarnya merupakan turunan dari salah satu konsep utama dalam basis data sejak lama, yaitu SQL. SQL adalah sebuah konsep pengoperasian basis data terutama untuk proses seleksi, pemasukan, perubahan dan penghapusan data yang dimungkinkan dapat dikerjakan dengan mudah dan otomatis.

Berikut adalah keunggulan dari MySQL, diantaranya:

1. Portabilitas, yaitu dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux, FreeBSD, Mac OS, Solaris dan Asigma.
2. Sumber Kode Terbuka yang berarti didistribusikan secara gratis dibawah lisensi dari GPL dimana setiap orang bebas untuk menggunakannya tetapi tidak boleh menjadikan MySQL sebagai induk turunan yang bersifat tertutup atau komersial.
3. Multi Pengguna yang berarti dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan.
4. Kinerja tinggi yang mempunyai kecepatan dalam menangani *query*.
5. Tipe kolom yang memiliki tipe data sangat kompleks, seperti *signed/unsigned integer, float, double, char, varchar, text, blob, date, time, datetime, timestamp, year* dan *enum*.

6. Perintah dan fungsi, memiliki operataor dan fungsi penuh yang mendukung *select* dan *where* dalam *query*.
7. Kemananan yang memiliki beberapa lapisan sekuritas seperti tingkat *subnet mask*, *hostname*, *privilage user* dengan sistem perijinan yang mendetail serta *password* yang terenkripsi.
8. Skalabilitas dan Batasan yang mampu menangani basis data dalam jumlah besar dengan jumlah *field* lebih dari 50 juta, 60 ribu tabel, 5 milyar *record* serta batas indeks mencapai 32 buah per tabel.
9. Lokalisasi yang dapat mendeteksi pesan kesalahan pada klien dengan menggunakan lebih dari 20 bahasa.
10. Keterhubungan yang dapat melakukan koneksi dengan klien mengguakan protokol TCP/IP, Unix Socket dan Named Pipes.
11. Antarmuka yang dimiliki terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API.
12. Klien dan Peralatan yang dilengkapi dengan berbagai peralatan yang dapat digunakan untuk administrasi basis data sekaligus dokumen petunjuk daring.
13. Struktur tabel yang dimiliki lebih fleksibel dalam menangani alter tabel dibandingkan dengan PostgreSQL dan Oracle.

Tipe data yang dikenal oleh MySQL sampai dengan versi 5 keatas adalah String, Number, Boolean dan Date dan secara detail didistribusikan dalam tabel II.3 berikut:

Tabel II.3 Tipe Data MySQL

Tipe Data	Keterangan
TINYINT	Integer yang sangat kecil.
SMALLINT	Integer yang kecil.

MEDIUMINT	Integer dengan ukuran menengah.
INT	Integer dengan ukuran normal.
INTEGER	Sama dengan INT.
BIGINT	Integer yang sangat besar.
FLOAT	Bilangan floating point yang kecil atau presisi tunggal
DOUBLE	Bilangan floating point dengan ukuran normal atau presisi ganda.
DOUBLE PRECISION	Sama dengan DOUBLE.
REAL	Sama dengan DOUBLE.
DECIMAL	Bilangan unpacked floating point (bilangan disimpan dalam String) yang tidak dapat dijadikan unsigned.
NUMERIC	Sama dengan DECIMAL.
DATE	Data berupa tanggal dengan format YYYY-MM-DD
DATETIME	Kombinasi dari tanggal dan waktu dengan format YYYY-MM-DD HH:MM:SS
TIME	Data berupa waktu dengan format HH:MM:SS
YEAR	Data bentuk tahun dengan format 2 atau 4
CHAR	Sebuah String dengan panjang tetap, sisa jumlah karakter yang belum diisi akan digantikan oleh spasi akan tetapi spasi akan dibuang saat data dipanggil
VARCHAR	Sebuah String dengan panjang berupa variabel.
TINYTEXT	Sebuah BLOB atau teks dengan panjang karakter sampai dengan 255 karakter.
TEXT	Sebuah BLOB atau teks dengan panjang karakter 65535 karakter
MEDIUMBLOB	Sebuah BLOB atau teks dengan panjang karakter maksimum 16777215 karakter.
LONGBLOB	Sebuah BLOB atau teks dengan panjang karakter maksimum 4294967295 karakter.
ENUM	Sebuah enumeration berupa obyek string yang hanya boleh memiliki salah satu nilai dari 'value1', 'value2', ..., NULL.
SET	Sebuah set dengan obyek string yang dapat memiliki nilai nol atau lebih yang diambil dari 'value1'

Sumber: Sutaji, 2012 : 41

II.12 Unified Modelling Language (UML)

UML adalah salah satu alat bantu yang sangat handal di dunia pengembangan sistem berorientasi objek. Hal ini dikarenakan UML menyediakan bahasa pemodelan visual yang memungkinkan pengembang sistem untuk membuat cetak biru atas visi mereka dalam bentuk yang baku, mudah dimengerti dan dilengkapi

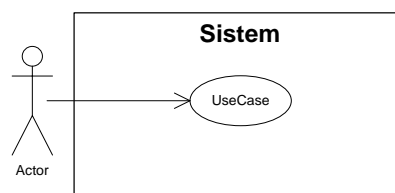
dengan mekanisme yang efektif untuk berbagi dan mengkomunikasikan rancangan sistem dengan pengembang lainnya (Munawar, 2005:17).

II.12.1. *Usecase Diagram*

Menurut Munawar (2005:63), *Usecase Diagram* adalah deskripsi fungsi dari sebuah sistem dengan perspektif pengguna. *Usecase* bekerja dengan mendeskripsikan tipikal interaksi antar pengguna sistem dengan sistem itu sendiri melalui skenario bagaimana sistem itu digunakan. Setiap skenario mendeskripsikan urutan kejadian dan setiap kejadian diinisialisasikan oleh pengguna, sistem, mesin atau urutan waktu. Dengan demikian secara singkat dapat dikatakan diagram *Usecase* adalah serangkaian skenario yang digabungkan bersama-sama oleh tujuan umum pengguna.

Dalam pembicaraan tentang *Usecase*, pengguna disebut dengan aktor, dimana aktor adalah sebuah peran yang dimainkan oleh pengguna dalam interaksinya dengan sistem.

Diagram *Usecase* menunjukkan tiga aspek dari sistem yaitu aktor, *Usecase* dan sistem/sub sistem *boundary*. Aktor mewakili peran orang, *usecase* mewakili aktivitas aktor dan sistem mewakili alat berkomunikasi antara aktor dengan *Usecase*. Notasi dan contoh diagram *Usecase* dapat dilihat pada gambar II.7 dibawah ini:








Gambar II.7 *Usecase Diagram*
Sumber: Munawar, 2005:64

II.12.2. Activity Diagram

Activity Diagram adalah teknik untuk mendiskripsikan logika prosedural, proses bisnis dan aliran kerja dalam banyak kasus. *Activity* Diagram mempunyai peran seperti halnya Flowchart, akan tetapi perbedaannya dengan Flowchart adalah *Activity* Diagram bisa mendukung perilaku paralel sedangkan Flowchart tidak bisa. Adapun simbol *Activity* Diagram dapat dilihat pada tabel II.4 :

Tabel II.4. Simbol Activity Diagram

Notasi	Keterangan
	Titik Awal
	Titik Akhir
	Activity
	Pilihan untuk pengambilan keputusan
	Fork digunakan menggabungkan kegiatan secara paralel dalam kegiatan

Sumber : Munawar, 2005:110

II.12.3. Class Diagram

Class Diagram adalah sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class* Diagram menggambarkan keadaan suatu 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* Diagram memiliki tiga area pokok :

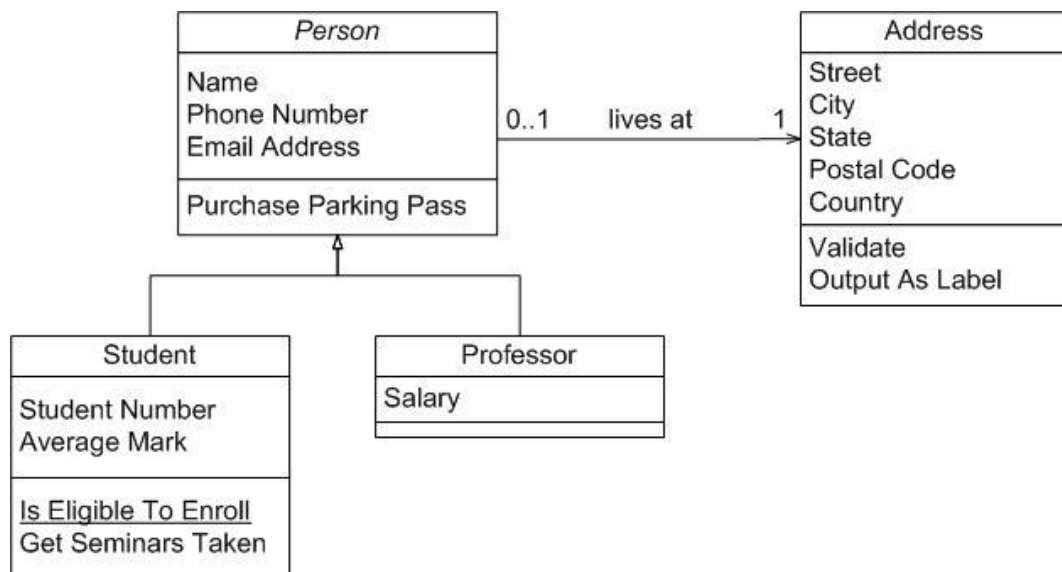
1. Nama kelas
2. Atribut

3. Metode

Atribut dan metode 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.
3. *Public*, dapat dipanggil oleh siapa saja.

Class dapat merupakan implementasi dari sebuah *interface*, yaitu *class* abstrak yang hanya memiliki metode. *Interface* tidak dapat langsung diinstansiasikan, tetapi harus diimplementasikan dahulu menjadi sebuah *class*. Contoh *Class Diagram* dapat dilihat pada gambar II.8 dibawah ini:



Gambar II.8 Class Diagram
Sumber: Munawar, 2005:220

II.12.4. Sequence Diagram

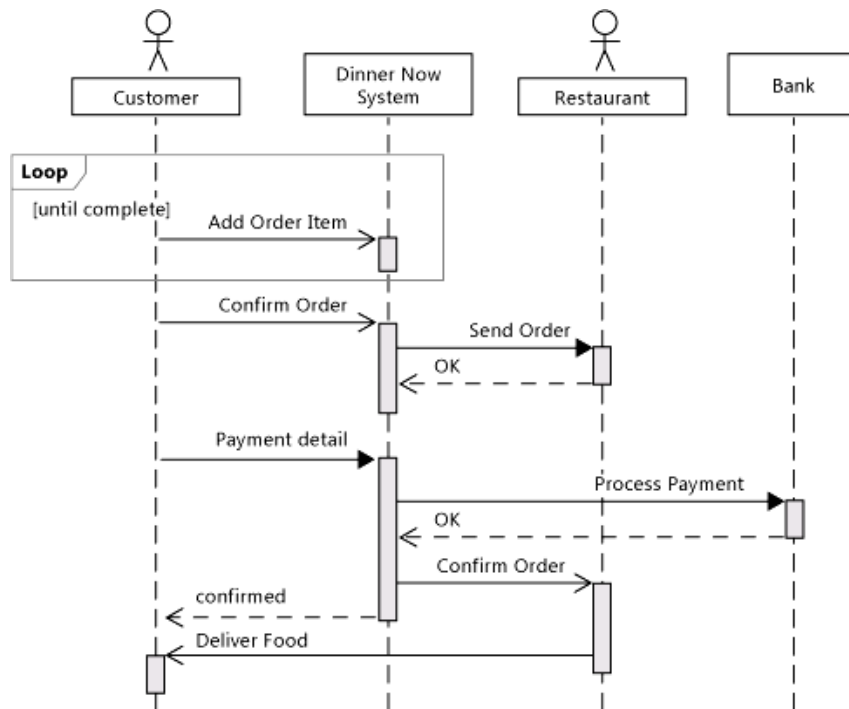
Sequence Diagram adalah diagram yang digunakan untuk menggambarkan perilaku pada sebuah skenario. Diagram ini menunjukkan sejumlah contoh objek dan pesan yang diletakkan diantara objek-objek ini di dalam *Usecase*.

Komponen utama Sequence Diagram terdiri dari objek yang dituliskan dengan kotak berbentuk segiempat dan memiliki nama. Pesan diawali oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan *progress vertical*.

Untuk membentuk *Sequence Diagram*, perlu diketahui notasi-notasi yang termasuk kedalam *Sequence Diagram*. Peraturan dalam membuat *Sequence Diagram* yaitu:

1. Objek atau partisipan diletakkan di dekat bagian atas diagram dengan urutan dari kiri ke kanan.
2. Setiap partisipan terhubung dengan garis titik-titik yang disebut Lifeline. Pada Lifeline terdapat kotak yang disebut Activation. Activation mewakili sebuah eksekusi operasi dari partisipan. Panjang kotak ini berbanding lurus dengan durasi Activation.
3. Sebuah message bergerak dari satu partisipan ke partisipan yang lain dan dari satu Lifeline ke Lifeline yang lain. Sebuah partisipan bisa mengirim sebuah pesan kepada dirinya sendiri.
4. Time adalah diagram yang mewakili waktu pada arah vertikal waktu dimulai dari atas akan dijalankan terlebih dahulu.

Contoh Sequence Diagram dapat dilihat pada gambar II.9 dibawah ini:



Gambar II.9 Sequence Diagram

Sumber: Munawar, 2005:89