

## **BAB II**

### **LANDASAN TEORI**

#### **2.1. Sistem Pendukung Keputusan**

Menurut Monita (2013) dalam jurnal yang berjudul Sistem Pendukung Keputusan Penerimaan Bantuan Langsung Tunai Dengan Menggunakan *Metode Analytical Hierarchy Process* “Konsep sistem pendukung keputusan (SPK) atau *Decision Support Sistem* (DSS) mulai dikembangkan pada tahun 1960-an, tetapi istilah Sistem pendukung keputusan itu sendiri baru muncul pada tahun 1971 yang diciptakan oleh G. Antony Gorry dan Michael S. Scott Morton dengan tujuan untuk menciptakan kerangka kerja guna mengarahkan aplikasi komputer kepada pengambilan keputusan manajemen. sistem tersebut adalah suatu sistem yang berbasis komputer yang ditujukan untuk membantu pengambil keputusan dengan memanfaatkan data dan model tertentu untuk memecahkan berbagai persoalan yang tidak terstruktur. Istilah sistem pendukung keputusan mengacu pada suatu sistem yang memanfaatkan dukungan komputer dalam proses pengambilan keputusan. sistem pendukung keputusan adalah bagian dari sistem informasi berbasis komputer (termasuk sistem berbasis pengetahuan (manajemen pengetahuan) yang dipakai untuk mendukung pengambilan keputusan dalam suatu organisasi atau perusahaan. Dapat juga dikatakan sebagai sistem komputer yang mengolah data menjadi informasi untuk mengambil keputusan dari masalah semi terstruktur yang spesifik.

### 2.1.1 Kriteria Sistem Pendukung Keputusan

Sistem pendukung keputusan dirancang secara khusus untuk mendukung seseorang yang harus mengambil keputusan-keputusan tertentu. Berikut ini beberapa kriteria sistem pendukung keputusan.

a. Interaktif

Sistem pendukung keputusan memiliki *user interface* yang komunikatif sehingga pemakai dapat melakukan akses secara cepat ke data dan memperoleh informasi yang dibutuhkan.

b. Fleksibel

Sistem pendukung keputusan memiliki sebanyak mungkin variable masukan, kemampuan untuk mengolah dan memberikan keluaran yang menyajikan alternatif-alternatif keputusan kepada pemakai.

c. Data Kualitas

Sistem pendukung keputusan memiliki kemampuan untuk menerima data kualitas yang dikuantitaskan yang sifatnya subyektif dari pemakainya, sebagai data masukan untuk pengolahan data. Misalnya terhadap kecantikan yang bersifat kualitas, dapat dikuantitaskan dengan pemberian bobot nilai seperti 75 atau 90.

d. Prosedur Pakar

Sistem pendukung keputusan mengandung suatu prosedur yang dirancang berdasarkan rumusan formal atau juga berupa prosedur kepakaran seseorang atau kelompok dalam menyelesaikan suatu bidang masalah dengan fenomena tertentu.

## 2.2. Internet Service Provider (ISP)

Internet Service Provider (ISP) adalah sebuah perusahaan penyedia layanan pada jasa sambungan internet dengan jasa lainnya yang saling berhubungan. ISP ini memiliki infrastruktur sebagai sarana telekomunikasi yang terkoneksi pada internet. ISP akan membagi kapasitas koneksi pada internet yang dimilikinya untuk para pelanggannya yang membutuhkan jasa internet. Biasanya sistem langganan telah ditetapkan dahulu ketentuannya. Sistem langganan ini pembayarannya tiap bulan. Ada juga provider telekomunikasi lainnya yang menerapkan sistem langganan dengan menggunakan sistem yang berbasis kuota (Sanjaya, 2009).

## 2.3. Metode Simple Additive Weigthing (SAW)

Menurut Eprilianto,et.al (2011) dalam jurnal Sistem Pendukung Keputusan Pemberian Beasiswa Menggunakan Metode *Simple Additive Weigthing* Di Universitas Panca Marga Probolinggo “Metode *Simple Additive Weigthing* (SAW) sering juga dikenal istilah metode penjumlahan terbobot. Konsep dasar metode SAW adalah mencari penjumlahan terbobot dari rating kinerja pada setiap *alternatif* pada semua atribut. Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating *alternatif* yang ada.

Metode ini merupakan metode yang paling terkenal dan paling banyak digunakan dalam menghadapi situasi *Multiple Attribute Decision Making*

(MADM). MADM itu sendiri merupakan suatu metode yang digunakan untuk mencari *alternatif* optimal dari sejumlah *alternatif* dengan kriteria tertentu.

Metode SAW ini mengharuskan pembuat keputusan menentukan bobot bagi setiap atribut. Skor total untuk *alternatif* diperoleh dengan menjumlahkan seluruh hasil perkalian antara rating (yang dapat dibandingkan lintas atribut) dan bobot tiap atribut. Rating tiap atribut haruslah bebas dimensi dalam arti telah melewati proses normalisasi matriks sebelumnya.

Berikut adalah rumus perhitungan Metode SAW :

#### A. Normalisasi Matriks

Metode SAW membutuhkan proses normalisasi matriks keputusan (X) ke suatu skala yang dapat diperbandingkan dengan semua rating alternatif yang ada.

$$r_{ij} = \begin{cases} \frac{x_{ij}}{\text{Max}_i X_{ij}} & \text{Jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_i X_{ij}}{X_{ij}} & \text{Jika } j \text{ adalah atribut biaya (cost)} \end{cases} \quad (1)$$

Dimana :

$r_{ij}$  = rating kinerja ternormalisasi.

$\text{Max}_i$  = nilai maksimum dari setiap baris dan kolom.

$\text{Min}_i$  = nilai minimum dari setiap baris dan kolom.

$X_{ij}$  = baris dan kolom dari matriks.

$(r_{ij})$  adalah rating kinerja ternormalisasi dari alternative pada atribut  $i=1,2,\dots,m$  dan  $j=1,2,\dots,n$ .

#### B. Nilai Preferensi

Nilai preferensi untuk setiap alternatif  $V_i$  diberikan sebagai:

$$V_i = \sum_{j=1}^n W_j r_{ij} \quad (2)$$

Dimana :

$V_i$  = Nilai akhir dari alternative

$W_j$  = Bobot yang telah ditentukan

$r_{ij}$  = Normalisasi matriks

Nilai  $V_i$  yang lebih besar mengindikasikan bahwa alternative  $A_i$  lebih terpilih.

### 2.3.1 Langkah Penyelesaian Simple Additive Weighting (SAW)

Ada beberapa langkah dalam penyelesaian metode Simple Additive Weighting (SAW). Yang diterapkan sebagai berikut :

1. Menentukan kriteria-kriteria yang dijadikan acuan dalam pendukung keputusan yaitu  $C_i$ .
2. Menentukan rating kecocokan setiap *alternative* pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria ( $C_i$ ).
4. Kemudian melakukan normalisasi matriks berdasarkan persamaan yang disesuaikan dengan jenis atribut (atribut keuntungan ataupun atribut biaya) sehingga diperoleh matriks ternormalisasi  $R$ .

5. Hasil akhir diperoleh dari proses perankingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai *alternatif* terbaik ( $A_i$ ) sebagai solusi.

### 2.3.2 Kelebihan Metode Simple Additive Weighting (SAW)

Kelebihan dari model *Simple Additive Weighting* (SAW) dibandingkan dengan model pengambilan keputusan yang lain terletak pada kemampuannya untuk melakukan penilaian secara lebih tepat karena didasarkan pada nilai kriteria dan bobot preferensi yang sudah ditentukan, selain itu SAW juga dapat menyeleksi *alternatif* terbaik dari sejumlah *alternatif* yang ada karena adanya proses perankingan setelah menentukan nilai bobot untuk setiap atribut. Berikut dapat dilihat tabel 2.1 dan tabel 2.2

Tabel 2.2 Alternatif

Tabel 2.1 Kriteria

No.	Kriteria
1.	Kecepatan Transfer Data
2.	Bandwidth
3.	Memiliki GGC(Google Global Cache)
4.	Ip Public /30
5.	Backbone/saluran utama koneksi
6.	Layanan yg diberikan
7.	Keamanan Data/ Firewall
8.	Harga
9.	Hardware
10.	Teknologi yg digunakan
11.	Dukungan teknis 24 jam
12.	Respon cepat trouble
13.	Garansi

No.	Alternatif
1.	Citra.Net
2.	Adi Sangroho.Net
3.	Solusindo Bintang Pratama (SBP).Net
4.	G Media.Net
5.	Nettindo
6.	Smartlink
7.	Solonet
8.	Buminet
9.	Solomedianet
10.	Lintas Arta
11.	Solo Sarana Network (SSN)
12.	Speedy
13.	FastLink
14.	ADSNet
15.	PowerNet
16.	RouteLINK
17.	PC 24 Net

## 2.4. Database

Database merupakan kumpulan data yang saling berhubungan satu dengan yang lainnya yang diorganisasikan berdasarkan sebuah skema atau struktur tertentu. Ada beberapa hal yang harus di perhatikan dalam membangun suatu database, yaitu :

a. Entitas

Merupakan suatu object yang memiliki karakteristik tertentu sebagai suatu attribute, misalnya entitas mahasiswa memiliki attribute, misalnya entitas mahasiswa memiliki attribute nama, alamat dan lain-lain.

b. Field

Mempresentasikan suatu atribut dari record yang menunjukkan suatu item dari data, seperti misalnya nama, alamat, dan lain sebagainya. Kumpulan dari field membentuk suatu record.

c. Record

Kumpulan dari field membentuk suatu record. Record menggambarkan suatu unit data individu tertentu. Kumpulan dari record membentuk suatu file. Misalnya file personalia, tiap-tiap record dapat mewakili data tiap-tiap karyawan.

d. Attribute

Merupakan karakteristik dari sebuah entitas. Nama, alamat merupakan attribute dari entitas mahasiswa.

Database adalah sekumpulan tabel-tabel yang berisi data dan merupakan kumpulan field atau kolom. Struktur file yang menyusun sebuah database adalah Data *Record* dan *Field*. Data adalah suatu satuan informasi yang akan diolah. Sebelum diolah, data dikumpulkan didalam suatu file database. *Record* adalah data yang isinya merupakan satu kesatuan seperti namauser dan password. Setiap keterangan yang mencakup namauser dan password dinamakan satu record. Setiap record diberi nomor urut yang disebut nomor record (Record number). *Field* adalah sub bagian dari record (Anhar ST, 2010).

Menurut Fathansyah (2012:2) basis data terdiri dari 2 kata, yaitu basis dan data. Basis kurang lebih dapat diartikan sebagai markas atau gudang, tempat bersarang atau berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, dan sebagainya.

Basis data (*database*) dapat didefinisikan dalam sejumlah sudut pandang seperti:

- a. Himpunan kelompok data yang saling berhubungan yang diorganisasi sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
- b. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redundancy*) yang tidak perlu, untuk memenuhi kebutuhan.



- c. Kumpulan file/table/arsip yang saling berhubungan yang disimpan kedalam media penyimpanan elektronik.

## 2.5. MySQL

Menurut Anhar (2010:45) “MySQL (*My Structure Query Language*) adalah salah satu *databases management sistem* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, *Postagre SQL*, dan lainnya. MySQL berfungsi untuk mengolah database menggunakan bahasa SQL. MySQL bersifat *open source* sehingga kita bias menggunakannya secara gratis. Pemrograman PHP juga sangat mendukung/ support dengan database MySQL.

Kelebihan MySQL :

- a. Dapat bekerja di beberapa *platform* yang berbeda, seperti LINUX, Windows, MacOS dll.
- b. Dapat dijadikan aplikasi database yang *portable* dan memiliki ukuran database yang cukup kecil.
- c. MySQL dapat diperoleh dengan mudah dan gratis.
- d. Sintaksnya lebih mudah dipahami dan tidak rumit serta pengaksesan database dapat dilakukan dengan mudah.
- e. Memiliki lebih banyak *type* data seperti : *signed/unsigned integer* yang memiliki panjang data sebesar 1,2,3,4 dan 8 byte, FLOAT, DOUBLE, CHAR, VARCHAR, TEXT, BLOB, DATE, TIME, DATETIME, TIMETAMP, YEAR, SET, dan tipe ENUM.

- f. Mendukung penuh terhadap perintah / *query* SQL GROUP BY dan ORDER BY. Mendukung fungsi (COUNT(), COUNT(DISTINCT), AVG(), STD(), SUM(), MAX() AND MIN() ).
- g. Mendukung terhadap LEFT OUTER JOIN dengan ANSI SQL dan sintak ODBC.
- h. Dapat dengan mudah di *backup* dan *restore database* dari satu sistem ke sistem yang lain.
- i. Mendukung ODBC (*Open Database Connectivity*) dalam lingkungan Windows. Sebagai contoh kita dapat menggunakan *Access* untuk *connect* ke MySQL server.
- j. MySQL merupakan program yang *multithreaded*, sehingga dapat dipasang pada server yang memiliki multi CPU.
- k. Dapat dikoneksikan dengan banyak bahasa pemrograman seperti C, C++, Java, Perl, PHP dan Python.
- l. *Privilege* (hak) dan password sangat fleksibel dan aman serta mengizinkan '*Host-Based*' Verifikasi.

## 2.6. PHP (Hypertext Preprocessor)

Menurut Octavian (2010) dalam bukunya yang berjudul Menjadi Programmer Jempola Menggunakan PHP, "PHP (PHP *Hypertext Prosesor*) adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemrograman berbasis kode-kode (*script*) yang di gunakan untuk mengolah suatu data dan mengirimkannya kembali ke *web browser* menjadi kode HTML. Kode PHP mempunyai ciri-ciri khusus, yaitu:

- a. Hanya dapat dijalankan menggunakan web server misalnya: *Apache*.
- b. Kode PHP dapat diletakkan dan dijalankan di web *server*.
- c. Kode PHP dapat digunakan untuk mengakses data bases, seperti: MySQL, PostgreSQL, Oracle dan lain-lain.
- d. Merupakan software yang bersifat *open source*.
- e. Gratis untuk didownload dan digunakan.
- f. Memiliki sistem *multiplatform*, artinya dapat dijalankan menggunakan sistem operasi apapun, seperti Linux, Unix, Windows dan lain-lain.

Ketika *e-commerce* semakin berkembang, situs-situs yang statis pun semakin ditinggalkan karena dianggap sudah tidak memenuhi keinginan pasar karena situs tersebut harus tetap dinamis selama setiap hari. Pada saat ini bahasa PERL dan GGI sudah jauh ketinggalan jaman sehingga sebagian besar *designer* web banyak beralih ke bahasa *server-side scripting* yang lebih dinamis seperti PHP. Seluruh aplikasi berbasis web dapat dibuat dengan PHP. Namun kekuatan yang paling utama PHP adalah pada konektivitasnya dengan *sistem database* di dalam web.

Keunggulan lainnya dari PHP adalah PHP juga mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP# bahkan HTTP. PHP dapat diinstal sebagai bagian atau modul dari *apache web server* atau sebagai CGI *script* yang mandiri. Banyak keuntungan yang dapat diperoleh jika menggunakan PHP sebagai modul dari *apache* diantaranya adalah :

- a. Tingkat keamanan yang cukup tinggi

- b. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemrograman web lainnya yang berorientasi pada *server-side-scripting*.
- c. Akses ke sistem database yang lebih fleksible, seperti MySQL.  
Adapun kelebihan-kelebihan dari PHP yaitu:
  - a. Mudah dibuat dan berkecepatan tinggi
  - b. PHP dapat berjalan lintas *platform*, yaitu dapat berjalan dalam sistem operasi dan *web server* apapun.
  - c. Dapat digunakan secara gratis.
  - d. Termasuk bahasa yang *embedded*, yakni dapat diletakkan dalam *tag* HTML.
  - e. Termasuk *server side programming*, sehingga kode asli/*source code* PHP tidak dapat dilihat di *browser* pengguna, yang terlihat hanya kode dalam format HTML.
  - f. Dapat memanfaatkan sumber-sumber aplikasi yang dimiliki oleh server, seperti misalnya untuk keperluan database connection, PHP dapat melakukan koneksi dengan berbagai database seperti MySQL, Oracle, *Sybase*, *mSQL*, *Solid*, *Generic ODBC*, *Postgres SQL*, *dBase*, *Direct MS-SQL*, *Velocis*, *IBM DB2*, *Interbase*, *frontbase*, *empress*, dan semua database yang mempunyai provider ODBC seperti misalnya MS Access dan lain-lain.

- g. PHP dapat melakukan semua aplikasi program CGI, seperti mengambil nilai form, menghasilkan halaman web yang dinamis, mengirimkan dan menerima *cookies*.
- h. PHP juga mendukung komunikasi dengan layanan lain melalui protocol IMAP, SNMP, NNTP, POP3 dan HTTP dan lainnya.

## **2.7. Adobe Dreamweaver**

Adobe Dreamweaver merupakan program penyunting halaman web dari Adobe Sistem yang dulu dikenal sebagai Macromedia Dreamweaver dari Macromedia. Program ini banyak digunakan oleh pengembang web karena fitur-fiturnya yang lengkap serta kemudahan dalam penggunaannya. Versi terakhir Macromedia Dreamweaver sebelum Macromedia dibeli oleh Adobe Systems yaitu versi 8. Kemudian setelah dibeli oleh Adobe Systems berkembang ke versi selanjutnya yakni versi 9 (CS3) versi 10 yang ada dalam Adobe Creative Suite 4 (CS4) dan versi terbarunya CS5. Dengan menggunakan program ini, seorang programmer web dapat dengan mudah membuat dan mendesain webnya karena bersifat WYSIWYG (What You See Is What You Get) yang artinya apa yang kita lihat pada halaman desain, maka semua itu akan kita peroleh pada browser. Dengan kelebihan ini seorang programmer atau web designer dapat langsung melihat hasil buaatannya tanpa harus membukanya pada browser (Kristanto Andri, 2003).

Sebagai editor web yang handal, Adobe Dreamweaver tentunya dilengkapi dengan kemampuan manajemen situs, yang memudahkan kita mengelola keseluruhan elemen yang ada dalam sebuah situs. Kita juga dapat

melakukan evaluasi situs dengan melakukan pengecekan broken link, kompatibilitas browser, termasuk validasi tag-tag HTML dan CSS yang tidak sesuai dengan pedoman secara otomatis serta perkiraan waktu download pada sebuah halaman web.

Adobe Dreamweaver CS4 merupakan salah satu program aplikasi yang digunakan untuk membangun sebuah website, baik secara grafis maupun dengan menulis kode sumber secara langsung. Adobe Dreamweaver CS4 memudahkan pengembang website untuk mengelola halaman-halaman website dan asset-aset, baik gambar (image), animasi flash, video, suara dan lain sebagainya (Komputer, 2010).

## **2.8. CSS**

CSS merupakan singkatan dari Cascading Style Sheet. Kegunaannya adalah untuk mengatur tampilan dokumen HTML, contohnya seperti pengaturan jarak antar baris, teks, warna dan format border bahkan penampilan file gambar. CSS dikembangkan oleh W3C, organisasi yang mengembangkan teknologi internet. Tujuannya tak lain untuk mempermudah proses penataan web. Perlu diingat, CSS hanyalah berupa kumpulan script yang tujuannya bukan untuk menggantikan HTML, melainkan pelengkap agar dokumen HTML bisa tampil lebih cantik dan dinamis.

Sejak ditemukannya CSS pada awal decade 90an, CSS terus dikembangkan dan diserap oleh web developer. Hingga sekarang telah mencapai versi ke-3. Kode CSS bersifat lintas platform, yang berarti script

ini dapat dibaca oleh berbagai macam sistem operasi dan browser. Hanya saja browser seperti Internet Explorer, seringkali salah mengartikan script CSS yang menyebabkan ketidaksempurnaannya tampilan dokumen HTML. Script CSS perlu dioptimalkan agar tampil maksimal pada browser internet explorer (jayan,2010).

## **2.9. Rekayasa Perangkat Lunak**

Menurut Shalahuddin (2011), perangkat lunak adalah program computer yang terasosiasi dengan dokumentasi perangkat lunak seperti dokumen kebutuhan, model desain, dan cara pengguna (user manual). Sebuah program computer tanpa terasosiasi dengan dokumentasinya maka belum dapat disebut perangkat lunak.




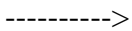
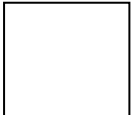


Perangkat lunak adalah suatu instruksi (program computer) yang dibuat oleh sebuah perusahaan computer untuk mengerjakan tugas tertentu. Memberikan fungsi dan untuk kerja seperti yang diinginkan atau mengatur struktur data memungkinkan program memanipulasi informasi secara proporsional, dan mengatur dokumen yang menggambarkan operasi dan kegunaan program.

## **2.10. UML (Unified Modeling Language)**

UML (*Unified Modeling Language*) adalah sebuah bahasa permodelan yang berorientasi objek dan menjadi standar dalam visualisasi, merancang, dan mendokumentasi sitem perangkat lunak untuk penyederhanaan permasalahan-permasalahan yang kompleks.

UML (*Unified Modelling Language*) adalah bahasa permodelan untuk sistem atau perangkat lunak yang berpradigma berorientasi objek. Permodelan (*modeling*) sesungguhnya digunakan untuk penyederhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami (Nugroho, 2010).

Tabel 2.3 Keterangan Atribut UML

Simbol	Keterangan
	<b>Actor:</b> Sebuah peran yang dimainkan oleh seseorang, sistem, atau perangkat yang memiliki saham dalam keberhasilan operasi dari sistem.
	<b>Use Case :</b> Untuk mengungkapkan tujuan bahwa sistem harus dicapai.
	<b>Generalization :</b> Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
	<b>Dependency :</b> Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memperngaruhi elemen yang bergantung pasanya elemen yang tidak mandiri ( <i>independent</i> ).
	<b>Sistem :</b> Menspesifikasikan paket yang menampilkan sistem secara terbatas.
	<b>Note :</b> Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.
	<b>Collaboration :</b> Interaksi aturan-aturan dan elemen lain



Simbol	Keterangan
	yang bekerja sama untuk menyediakan perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> )
_____	Association : Apa yang menghubungkan antara objek satu dengan objek lainnya.

### 2.10.1 Diagram Dasar dalam UML

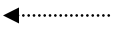
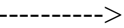

Berikut ini adalah penjelasan mengenai berbagai diagram UML serta tujuannya :

#### 1. *Use case diagram*

*Use case diagram* merupakan permodelan untuk menggambarkan kelakuan (behavior) sistem secara keseluruhan yang akan dibuat diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat.

Tabel 2.4 Keterangan Atribut *Use case diagram*

Simbol	Keterangan
_____	<b>Generalization</b> : Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
◇	<b>Nary Association</b> : Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
▭	<b>Class</b> : Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
○	<b>Collaboration</b> : Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan

Simbol	Keterangan
	perilaku yang lebih besar dari jumlah dan elemen-elemennya ( <i>sinergi</i> )
	<b>Realization</b> : Operasi yang benar-benar dilakukan oleh suatu objek.
	<b>Dependency</b> : Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan mempengaruhi elemen yang bergantung pasanya elemen yang tidak mandiri ( <i>independent</i> ).
	<b>Association</b> : Apa yang menghubungkan antara objek satu dengan objek lainnya.

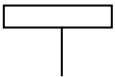
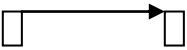
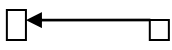
## 2. Diagram Struktur Statis

UML menawarkan dua diagram untuk memodelkan struktur statis sistem informasi, yaitu:

### 1. *Class Diagram*

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

Tabel.2.5.Keterangan Atribut Class Diagram

Simbol	Keterangan
	LifeLine : Objek <i>entity</i> , antarmuka yang saling berinteraksi.
	Message : Spesifikasi dari komunikasi antar object yang memuat informasi-informasi tentang aktifitas yang terjadi.
	Spesifikasi dari komunikasi antar objek yang

Simbol	Keterangan
	memuat informasi-informasi tentang aktifitas yang terjadi.

## 2. *Object Diagram*

Serupa dengan class diagram, tetapi object diagram memodelkan *instance object actual* dengan menunjukkan nilai-nilai saat ini dari atribut *instance*. Object Diagram menyajikan “*snapshot/potret*” tentang objek sistem pada point waktu tertentu. Diagram ini tidak digunakan sesering Class Diagram, tetapi saat digunakan dapat membantu seorang developer memahami struktur sistem secara lebih baik.

## 3. *Diagram Interaksi*







Diagram interaksi memodelkan sebuah interaksi, terdiri dari satu set objek, hubungan-hubungannya, dan pesan yang terkirim di antara objek. Model diagram ini memodelkan *behavior* (kelakuan) sistem yang dinamis dan UML memiliki dua diagram untuk tujuan ini, yaitu :

### a. *Sequence Diagram*

Diagram sekuen menggambarkan kelakuan/perilaku objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Oleh karena itu untuk menggambar diagram sekuen maka harus diketahui objek-objek yang

terlihat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu.

Tabel 2.6. Keterangan Atribut Sequence Diagram

Simbol	Keterangan
	<b>Initial Pseudo State</b> : Bagaimana objek dibentuk atau diawali
	<b>Final State</b> : Bagaimana objek dibentuk dan dihancurkan
	<b>State</b> : Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
	<b>Transition</b> : Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya.
	<b>Association</b> : Apa yang menghubungkan antara objek satu dengan objek lainnya.
	<b>Note</b> : Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Collaboration Diagram*

Serupa dengan diagram rangkaian/sekuensi, tetapi tidak focus pada timing atau sekuensi pesan. Diagram ini justru menggambarkan interaksi (atau kolaborasi) antara objek dalam sebuah format jaringan. Diagram rangkaian maupun diagram kolaborasi merupakan *isomorphic* artinya kita dapat mengubah dari satu diagram ke diagram lain.





#### 4. *State Diagram*

UML memiliki sebuah diagram untuk memodelkan behavior objek khusus yang kompleks (*state chart*) dan sebuah diagram untuk memodelkan behavior dari sebuah use case atau sebuah metode, yaitu:

##### a. *Diagram statechart*

digunakan untuk memodelkan behavior objek khusus yang dinamis. Diagram ini mengilustrasikan siklus hidup objek berbagai keadaan yang dapat diasumsikan oleh objek dan event-event (kejadian) yang menyebabkan objek beralih dari satu state ke state lain.

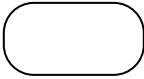




Tabel 2.7. Keterangan Atribut *Diagram Statechart*

Simbol	Keterangan
	<b>Action</b> : State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<b>Point</b> : Digunakan untuk menggambarkan apakah akan masuk (entry point) ke dalam state atau akan keluar (exit point)
	<b>Initial Node</b> : Bagaimana objek dibentuk atau diawali
	<b>Activity Final Node</b> : Bagaimana objek dibentuk dan dihancurkan
[guard]	<b>Guard</b> : Syarat terjadinya transisi yang bersangkutan
event	<b>Event</b> : Digunakan untuk mendeskripsikan kondisi yang menyebabkan sesuatu pada state.

b. *Diagram activity*

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

Tabel.2.8. Keterangan Atribut Diagram Activity

Simbol	Keterangan
	<b>Action</b> : State dari sistem yang mencerminkan eksekusi dari suatu aksi.
	<b>Activity</b> : Menunjukkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain.
	<b>Initial Node</b> : Bagaimana objek dibentuk atau diawali
	<b>Activity Final Node</b> : Bagaimana objek dibentuk dan dihancurkan
	<b>Fork Node</b> : Satu aliran yang pada tahap tertentu berubah menjadi beberapa aliran

5. Diagram Implementasi

Diagram implementasi juga memodelkan struktur sistem informasi, yaitu:

a. **Componen Diagram**

Digunakan untuk menggambarkan organisasi dan ketergantungan komponen-komponen software sistem. Komponen diagram dapat

digunakan untuk menunjukkan bagaimana kode pemrograman dibagi menjadi modul-modul (atau komponen).

b. Deployment

Digunakan untuk mendiskripsikan arsitektur fisik dalam istilah “node” untuk *hardware* dan *software* dalam sistem.