BAB II

LANDASAN TEORI

2.1. Sistem Pendukung Keputusan

Menurut Monita (2013) dalam jurnal yang berjudul Sistem Pendukung Keputusan Penerima Bantuan Langsung Tunai Dengan Menggunakan *Metode Analytical Hierarcy Process* "Konsep sistem pendukung keputusan (SPK) atau *Decision Support System* (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 variabel masukan, kemampuan untuk mengolah dan memberikan keluaran yang menyajikan alternatif-alternatif keputusan kepada pemakai.

c. Data Kualitas

   Sistem pendukung keputusan memiliki kemampuan utuk menerima data kualitas yang dikuantitaskan yang sifatnya subyektif dari pemakai nya, 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. *Simple Additive Weighting Method*

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

$$r_{ij} = \begin{cases} \dfrac{x_{ij}}{\underset{i}{Max}\, x_{ij}} & \textit{jika j atribut keuntungan (benefit)} \\ \dfrac{\underset{i}{Min}\, x_{ij}}{x_{ij}} & \text{jika j atribut biaya (cost)} \end{cases}$$

dimana rij adalah rating kinerja ternormalisasi dari alternatif Ai pada atribut Cj; i=1,2,...,m dan j=1,2,...,n. Nilai preferensi untuk setiap alternative (Vi) diberikan rumus sebagai berikut:

$$V_i = \sum_{j=1}^{n} w_j r_{ij}$$

Nilai Vi yang lebih besar mengindikasikan bahwa alternatif Ai lebih terpilih.

2.2.1. 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 Ci.
2. Menentukan rating kecocokan setiap alternatif pada setiap kriteria.
3. Membuat matriks keputusan berdasarkan kriteria (Ci).
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 perangkingan yaitu penjumlahan dari perkalian matriks ternormalisasi R dengan vector bobot sehingga diperoleh nilai terbesar yang dipilih sebagai alternatif terbaik (Ai) sebagai solusi.

2.2.2. 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.

2.3. Database

Menurut Anhar (2010:45), "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*".

Menurut Fathansyah (2012:2) basis data terdiri atas 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 didefenisikan dalam sejumlah sudut pandang seperti:

a. Himpunan kelompok data (arsip) 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 (*redudancy*) yang tidak perlu, untuk memenuhi kebutuhan.
c. Kumpulan file/tabel/arsip yang saling berhubungan yang disimpan kedalam media penyimpanan elektronik.

2.4. MySQL

Menurut Anhar (2010:45) "MySQL (*My Structure Query Language*) adalah salah satu *databases management system* (DBMS) dari sekian banyak DBMS seperti Oracle, MS SQL, *Postagre* SQL, dan lainya". MySQL berfungsi untuk mengolah database menggunakan bahasa SQL. MySQL

bersifat *open source* sehingga kita bisa menggunakannya secara gratis. Pemograman 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, TIMESTAMP, 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 OUTHER 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 bahasa C, C++, Java, Perl, PHP dan Python.
l. *Privilege* (hak) dan password sangat fleksibel dan aman serta mengijinkan '*Host-Based'* Verifikasi.

2.4. PHP

Menurut Octavian (2010) dalam bukunya yang berjudul Menjadi Programmer Jempola Menggunakan PHP "PHP (PHP *Hypertext Prosesor*) adalah akronim dari *Hypertext Preprocessor*, yaitu suatu bahasa pemograman berbasiskan 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 diletakan dan dijalankan di *web server*.

c. Kode PHP dapat digunakan untuk mengakses data bases, seperti: MY SQL, 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 statispun semakin ditinggalkan karena dianggap sudah tidak memenuhi keinginan pasar karena situs tersebut harus tetap dinamis selama setiap hari. Pada saat ini bahasa PERL dan CGI 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 *system database* di dalam web.

Keunggulan lainnya dari PHP adalah PHP juga mendukung komunikasi dengan layanan seperti protocol IMAP, SNMP, NNTP, POP3 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* di antaranya adalah :

a. Tingkat keamanan yang cukup tinggi

b. Waktu eksekusi yang lebih cepat dibandingkan dengan bahasa pemograman web lainnya yang berorientasi pada *server-side scripting*.

c. Akses ke sistem database yang lebih fleksibel. 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 dlihat 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 profider 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 protokol IMAP, SNMP, NNTP, POP3 dan HTTP dan lainnya.

2.5. *Macromedia Dreamweaaver*

Adobe Dreamweaver merupakan program penyunting halaman web dari Adobe Systems 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 yaitu CS5.

Adobe Dreamweaver CS4 merupakan salah satu program apliksi 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).

2.6. *Webserver & Apache*

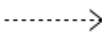
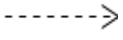
Web server merupakan perangkat lunak yang mengelola (mengatur) permintaan user dari browser dan hasilnya dikembalikan kembali ke browser. Contoh web server adalah IIS (*Internet Information Services*) produk Microsoft Corp (Supardi, 2010).

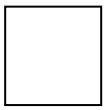
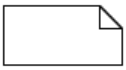
2.7. UML

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

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

Tabel 2. 1: Keterangan Atribut UML

No	Gambar	Nama	Keterangan
1		*Actor*	Menspesifikasikan himpuan peran yang pengguna mainkan ketika berinteraksi dengan *use case*.
2		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempengaruhi elemen yang bergantung padanya elemen yang tidak mandiri (*independent*).
3		*Generalization*	Hubungan dimana objek anak (*descendent*) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
4		*Include*	Menspesifikasikan bahwa *use case* sumber secara *eksplisit*.

No	Gambar	Nama	Keterangan
5		*Extend*	Menspesifikasikan bahwa *use case* target memperluas perilaku dari *use case* sumber pada suatu titik yang diberikan.
6		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
7		*System*	Menspesifikasikan paket yang menampilkan sistem secara terbatas.
8		*Use Case*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
9		*Collaboration*	Interaksi aturan-aturan dan elemen lain yang bekerja sama untuk menyediakan prilaku yang lebih besar dari jumlah dan elemen-elemennya (sinergi).
10		*Note*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi

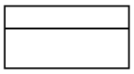
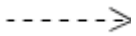
2.7.1. Diagram Dasar dalam UML

Berikut ini adalah penjelasan mengenai berbagai diagram UML serta tujuannya:

1. *Use case diagram*

   Use case diagram merupakan pemodelan untuk menggambarkan kelakuan (behavior) sistem secara keseluran yang akan dibuat. Diagram use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem yang akan dibuat.

   Tabel 2. 2: Keterangan Atribut *Use case diagram*

No	Gambar	Nama	Keterangan
1		*Generalization*	Hubungan dimana objek anak *(descendent)* berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk (*ancestor*).
2		*Nary Association*	Upaya untuk menghindari asosiasi dengan lebih dari 2 objek.
3		*Class*	Himpunan dari objek-objek yang berbagi atribut serta operasi yang sama.
4		*Collaboration*	Deskripsi dari urutan aksi-aksi yang ditampilkan sistem yang menghasilkan suatu hasil yang terukur bagi suatu aktor
5		*Realization*	Operasi yang benar-benar dilakukan oleh suatu objek.
6		*Dependency*	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri *(independent)* akan mempegaruhi elemen yang bergantung padanya elemen yang tidak mandiri
7		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya

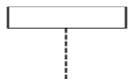
2. Diagram Struktur Statis

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

a. Class diagram

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

Tabel 2. 3: Keterangan Atribut Class Diagram

No	Gambar	Nama	Keterangan
1		*LifeLine*	Objek *entity*, antarmuka yang saling berinteraksi.
2		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi
3		*Message*	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi

b. *Object Diagram*

Serupa dengan class diagram, tetapi object diagram memodelkan *isntance object actual* dengan menunjukan 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.

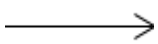
3. Diagram Interaksi

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

Tabel 2. 4: Keterangan Atribut Sequence Diagram

No	Gambar	Nama	Keterangan
1		*State*	Nilai atribut dan nilai link pada suatu waktu tertentu, yang dimiliki oleh suatu objek.
2		*Initial Pseudo State*	Bagaimana objek dibentuk atau diawali
3		*Final State*	Bagaimana objek dibentuk dan dihancurkan
4		*Transition*	Sebuah kejadian yang memicu sebuah state objek dengan cara memperbaharui satu atau lebih nilai atributnya

No	Gambar	Nama	Keterangan
5		*Association*	Apa yang menghubungkan antara objek satu dengan objek lainnya.
6		*Node*	Elemen fisik yang eksis saat aplikasi dijalankan dan mencerminkan suatu sumber daya komputasi.

b. *Collaboration Diagram*

Serupa dengan diagram rangkaian/sekuensi, tetapi tidak fokus 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.

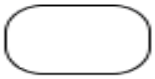
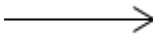
4. State Diagram

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. 5: Keterangan Atribut *Diagram Statechart*

No	Gambar	Nama	Keterangan
1		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
2		*Point*	Digunakan untuk menggambarkan apakah akan masuk (entry point) ke dalam state atau akan keluar (exit point)
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5	[guard]	*Guard*	Syarat terjadinya transisi yang bersangkutan
6	event	*Event*	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 aktor, jadi aktivitas yang dapat dilakukan oleh sistem.

Tabel 2. 6: Keterangan Atribut Diagram Activity

No	Gambar	Nama	Keterangan
1		*Activity*	Memperlihatkan bagaimana masing-masing kelas antarmuka saling berinteraksi satu sama lain
2		*Action*	State dari sistem yang mencerminkan eksekusi dari suatu aksi
3		*Initial Node*	Bagaimana objek dibentuk atau diawali.
4		*Activity Final Node*	Bagaimana objek dibentuk dan dihancurkan
5		*Fork Node*	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 menunjukan 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.