

BAB II

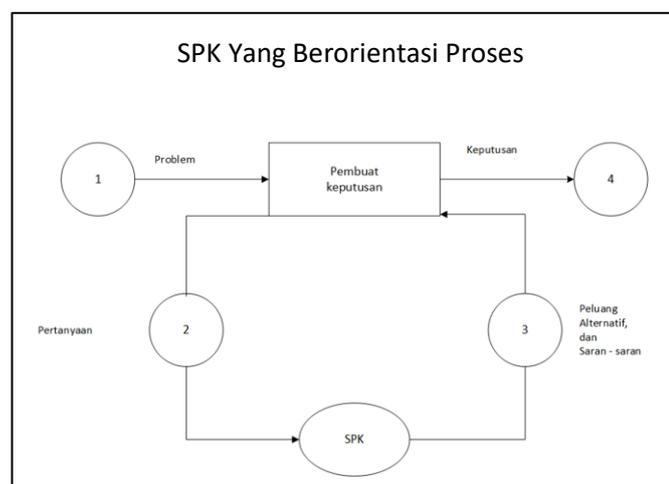
LANDASAN TEORI

2.1 Sistem Pendukung Keputusan (*Decision Support System*)

Menurut Siti Aisyah (2019:2) Sistem Pendukung Keputusan (SPK) atau *Decision Support System* (DSS), merupakan suatu sistem interaktif yang mendukung penentuan keputusan melalui alternatif-alternatif yang diperoleh dari hasil pengolahan data, informasi dan perancangan model. SPK adalah sistem berbasis model yang terdiri dari prosedur-prosedur dalam pemrosesan data beserta pertimbangan-pertimbangannya, guna membantu manajer mengambil keputusan.

Menurut Aditya Dwi Jatmiko, Sri Huning Anwariningsih dan Dahlan Susilo (2016:13) Sistem Pendukung Keputusan (SPK) atau *Decision Support Systems* (DSS) adalah sebuah sistem yang dimaksudkan untuk mendukung para pengambil keputusan manajerial dalam situasi keputusan semi terstruktur. DSS dimaksudkan untuk menjadi alat bantu bagi para pengambil keputusan untuk memperluas kapabilitas mereka, namun tidak untuk menggantikan penilaian mereka.

DSS ditujukan untuk keputusan - keputusan yang memerlukan penilaian atau pada keputusan - keputusan yang sama sekali tidak dapat didukung oleh algoritma (Kusrini, Awaluddin M, 2006).



Gambar 2.1 DSS Proses

Dengan pengertian-pengertian tersebut, dapat disimpulkan bahwa SPK bukan merupakan alat pengambilan keputusan, melainkan suatu sistem yang membantu para pengambil keputusan melengkapi informasi dari data yang telah diolah secara relevan, lalu digunakan untuk membuat keputusan atas suatu masalah secara cepat dan akurat.

2.2.1 Konsep Dasar Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) mulai dikembangkan pada tahun 1960-an, tetapi istilah sistem pendukung keputusan itu sendiri baru muncul pada tahun 1971, yang diciptakan oleh G. Anthony Gorry dan Micheal S.Scott Morton, keduanya adalah profesor di MIT. Hal itu mereka lakukan dengan tujuan untuk menciptakan kerangka kerja guna mengarahkan aplikasi computer kepada pengambilan keputusan manajemen.

Sementara itu, perintis sistem pendukung keputusan yang lain dari MIT, yaitu Peter G.W.Keen yang bekerja sama dengan Scott Morton telah mendefinisikan tiga tujuan yang harus dicapai oleh sistem pendukung keputusan yaitu:

- 1) Sistem harus dapat membantu manajer dalam membuat keputusan guna memecahkan masalah semi terstruktur.
- 2) Sistem harus dapat mendukung manajer, bukan mencoba menggantikannya.
- 3) Sistem harus dapat meningkatkan efektivitas pengambilan keputusan manajer.

Tujuan-tujuan tersebut mengacu pada tiga prinsip dasar sistem pendukung keputusan (Kadarsah, 1998), yaitu :

1) Struktur Masalah

Untuk masalah yang terstruktur, penyelesaian dapat dilakukan dengan menggunakan rumus-rumus yang sesuai, sedangkan untuk masalah terstruktur tidak dapat dikomputerisasi. Sementara itu, sistem pendukung keputusan dikembangkan khusus untuk menyelesaikan masalah yang semi terstruktur.

2) Dukungan Keputusan

Sistem pendukung keputusan tidak dimaksudkan untuk menggantikan manajer, karena komputer berada di bagian terstruktur, sementara manajer berada di bagian tidak terstruktur untuk memberikan penilaian dan melakukan

analisis. Manajer dan komputer bekerja sama sebagai sebuah tim pemecah masalah semi terstruktur.

3) Efektivitas Keputusan

Tujuan utama dari sistem pendukung keputusan bukanlah mempersingkat waktu pengambilan keputusan, tetapi agar keputusan yang dihasilkan dapat lebih baik.

2.2.2 Karakteristik dan Komponen Penyusun SPK

SPK memiliki beberapa karakteristik, yaitu :

- 1) Mendukung proses pengambilan keputusan yang berfokus pada *management by perception*.
- 2) Merupakan *interface* manusia dan mesin, dimana manusia sebagai pemegang kendali proses pengambil keputusan.
- 3) Dapat membantu pengambil keputusan dalam mengenali masalah terstruktur, semi struktur dan tak struktur.
- 4) Memiliki kapasitas dialog untuk memperoleh informasi sesuai kebutuhan.
- 5) Memiliki subsistem-subsistem yang terintegrasi sedemikian rupa sehingga dapat berfungsi sebagai satu kesatuan *item*.
- 6) Membutuhkan struktur data yang dapat melayani kebutuhan informasi bagi seluruh tingkatan manajemen.

SPK memiliki tiga subsistem utama dalam menentukan kapabilitas teknis sistem pendukung keputusan, antara lain :

- 1) Subsistem Manajemen Basis Data.
- 2) Subsistem Manajemen Basis Model.
- 3) Subsistem Dialog.

2.3 Pengertian Beasiswa

Pengertian beasiswa menurut Kamus Besar Bahasa Indonesia yaitu tunjangan uang yang diberikan kepada pelajar atau mahasiswa sebagai bantuan biaya belajar.

Menurut Asnawati, Prahasti dan Liza Yulianti (2021:749) Beasiswa dapat diberikan oleh lembaga pemerintah, perusahaan ataupun yayasan. Pemberian beasiswa dapat dikategorikan pada pemberian cuma-cuma ataupun pemberian

dengan ikatan kerja (biasa disebut ikatan dinas) setelah selesainya pendidikan. Lama ikatan dinas ini berbeda-beda, tergantung pada lembaga yang memberikan beasiswa tersebut. Beasiswa juga banyak diberikan kepada perkelompok (group) misalnya ketika ada event perlombaan yang diadakan oleh lembaga pendidikan, dan salah satu hadiahnya adalah beasiswa.

Beasiswa dapat diartikan sebagai bentuk penghargaan yang diberikan kepada individu agar dapat melanjutkan pendidikan ke jenjang yang lebih tinggi, penghargaan itu dapat berupa akses tertentu pada suatu instansi atau penghargaan berupa bantuan keuangan.

2.4 Multiple Criteria Decision Making (MCDM)

Menurut Diyah Ruswanti (2019:69) *Multi Criteria Decision Making* (MCDM) telah menjadi salah satu area riset operasional yang berkembang paling cepat selama dua dekade terakhir. Teori MCDM sendiri dapat dibagi menjadi *Multi Objective Decision Making* (MODM) dan *Multi Attribute Decision Making* (MADM).

MODM menganalisa subset dari ruang vektor secara berkelanjutan, yang biasanya dibatasi oleh batasan - batasan, dengan mengalokasi semua solusi yang efisien, sebelum menentukan nilai optimum berdasarkan pada pilihan user (pengguna). Oleh karena itu, MODM memungkinkan untuk digunakan dalam perencanaan operasional, semisal pemrograman tujuan.

MADM adalah suatu pendukung pengambilan keputusan dengan masukan beberapa atribut/variabel. Dalam *literature* teori keputusan, metode MADM digolongkan menurut informasi tertentu.

2.5 Pengertian Metode Simple Additive Weighting (SAW)

Metode SAW juga merupakan metode MADM yang paling sederhana dan paling banyak digunakan. Metode ini juga metode yang paling mudah untuk diaplikasikan, karena mempunyai algoritma yang tidak terlalu rumit.

Menurut Sri Eniyati (2011:174) Metode SAW dikenal sebagai istilah 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. Formula untuk melakukan normalisasi tersebut adalah sebagai berikut:

$$r_{ij} = \begin{cases} \frac{X_{ij}}{\text{Max}_{X_{ij}}} & \text{jika } j \text{ adalah atribut keuntungan (benefit)} \\ \frac{\text{Min}_{X_{ij}}}{X_{ij}} & \text{jika } j \text{ adalah atribut biaya (cost)} \end{cases}$$

r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$.

Keterangan :

- r_{ij} = nilai rating kinerja ternormalisasi
- X_{ij} = nilai atribut yang dimiliki dari setiap kriteria
- $\text{Max}_{X_{ij}}$ = nilai terbesar dari setiap kriteria
- $\text{Min}_{X_{ij}}$ = nilai terkecil dari setiap kriteria
- benefit = jika nilai terbesar adalah terbaik kriteria
- cost = jika nilai terkecil adalah terbaik kriteria
- dimana r_{ij} adalah rating kinerja ternormalisasi dari alternatif A_i pada atribut C_j ; $i=1,2,\dots,m$ dan $j=1,2,\dots,n$. Nilai preferensi untuk setiap alternatif (V_i) diberikan sebagai:

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

Keterangan :

- V_i = ranking untuk setiap alternatif
- w_j = nilai bobot dari setiap kriteria
- r_{ij} = nilai rating kinerja ternormalisasi
- Nilai V_i yang lebih besar mengindikasikan bahwa alternatif A_i lebih terpilih.

2.6 Pengertian Bahasa Pemrograman

Instruksi standar untuk memerintah komputer yaitu bahasa pemrograman. Bahasa ini dapat memungkinkan seorang yang ahli program untuk menentukan secara persis data mana yang akan diolah oleh komputer.

2.6.1 PHP

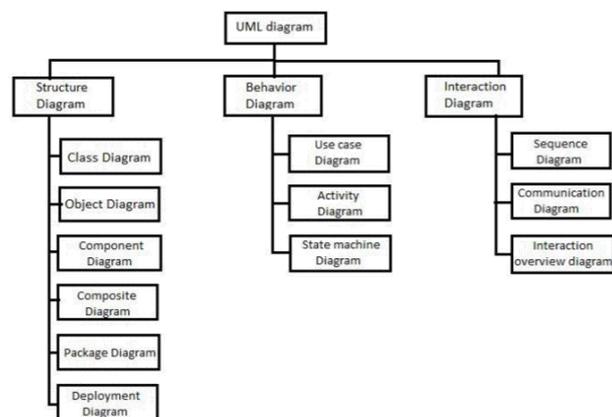
PHP adalah script pemrograman yang terletak dan dieksekusi di server. Salah satunya adalah untuk menerima, mengolah, dan menampilkan data dari dan ke sebuah situs. Data akan diolah ke sebuah database server untuk kemudian hasilnya ditampilkan di browser sebuah situs.

PHP merupakan bahasa berbentuk script yang ditempatkan didalam server baru kemudian diproses. Kemudian hasil pemrosesan dikirim kepada web browser klien. Bahasa pemrograman ini dirancang khusus untuk membentuk web dinamis. Artinya, pemrograman PHP dapat membentuk suatu tampilan berdasarkan permintaan terkini, misalnya halaman menampilkan daftar siswa. Halaman tersebut akan selalu mengalami perubahan mengikuti jumlah data siswa yang telah ditambahkan.

2.7 UML (Unified Modeling Language)

UML merupakan bahasa visul untuk pemodelan dan komunikasi mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. UML hanya berfungsi untuk melakukan pemodelan. Jadi penggunaan UML tidak terbatas pada metodologi tertentu, meskipun pada kenyataannya UML paling banyak digunakan pada metodologi berorientasi objek.

Menurut (Sukamto & Shalahuddin, 2014). UML terdiri dari 13 diagram yang dikelompokkan dalam 3 kategori. Pembagian kategori dan macam- macam diagram tersebut dapat dilihat pada Gambar 2.2.



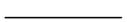
Gambar 2.2 Macam – macam diagram UML

2.7.1 Use Case Diagram

Diagram *use case* merupakan pemodelan untuk kelakuan sistem informasi yang akan dibangun. *Use case* mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibangun. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada pada sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi tersebut.

Berikut ini adalah simbol-simbol diagram use case, seperti yang terlihat pada Tabel 2.1 dibawah ini :

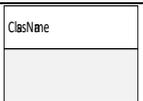
Tabel. 2.1 *Simbol Usecase Diagram*

SIMBOL	KETERANGAN
	<i>Actor</i> Menggambarkan tokoh atau seseorang yang berinteraksi dengan sistem. Dan dapat menerima dan memberi informasi pada sistem.
	<i>Use case</i> Menjelaskan fungsi dari kegunaan sistem yang di rancang.
	<i>Association</i> Menghubungkan antara <i>use case</i> dengan aktor tertentu.
	<i>Include</i> Menunjukkan bahwa <i>use case</i> satu merupakan bagian dari <i>use case</i> lainnya.
	<i>Extend</i> Menunjukkan arah panah secara putus-putus dari <i>use case</i> ke <i>base use case</i> .

2.7.2 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem. Kelas memiliki apa yang disebut atribut dan metode atau operasi. Berikut adalah simbol- simbol class diagram, seperti terlihat pada Tabel 2.2 berikut ini:

Tabel. 2.2 *Simbol Class Diagram*

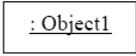
SIMBOL	KETERANGAN
	<i>Class</i> Menggambarkan sebuah kelas pada sistem yang terbagi menjadi 3 bagian. Bagian atas adalah nama kelas. Bagian tengah adalah atribut kelas. Bagian bawah adalah metode dari kelas.

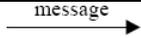
	<i>Association</i> Hubungan statis antar kelas. menggambarkan kelas yang memiliki atribut berupa kelas lain atau kelas yang harus mengetahui eksistensi kelas lain.
	<i>Agregation</i> Hubungan yang menyatakan bahwa suatu kelas menjadi atribut bagi kelas lain.
	<i>Composition</i> Bentuk khusus dari agregation dimana kelas yang menjadi bagian diciptakan setelah kelas whole dibuat.
	<i>Generalization</i> Relasi antar kelas dengan makna generalisasi/spesialisasi (umum-khusus)
	<i>Directed Assocoation</i> Asosiasi dengan makna kelas yang satu digunakan oleh kelas yang lain.

2.7.3 Sequence Diagram

Diagram sekuen menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek. Untuk menggambarkan diagram sekuen maka harus diketahui objek-objek yang terlibat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. Berikut adalah simbol-simbol yang ada pada sequence diagram seperti terlihat pada Tabel 2.3 berikut ini:

Tabel. 2.3 *Simbol Sequence Diagram*

SIMBOL	NAMA	KETERANGAN
	Object	Object merupakan instance dari sebuah class dan dituliskan tersusun secara horizontal. Digambarkan sebagai sebuah class (kotak) dengan nama obyek didalamnya yang diawali dengan sebuah titik koma.
	Actor	Actor juga dapat berkomunikasi dengan object, maka actor juga dapat diurutkan sebagai kolom. Simbol Actor sama dengan simbol pada Actor Use Case Diagram.
	Lifeline	Lifeline mengindikasikan keberadaan sebuah object dalam basis waktu. Notasi untuk Lifeline adalah garis putus-putus vertikal yang ditarik dari sebuah obyek.

	Activation	Activation dinotasikan sebagai sebuah kotak segi empat yang digambar pada sebuah lifeline. Activation mengindikasikan sebuah obyek yang akan melakukan sebuah aksi.
	Message	Message, digambarkan dengan anak panah horizontal antara Activation. Message mengindikasikan komunikasi antara object-object.

2.8 *Blackbox Testing*

Blackbox testing adalah tahap yang digunakan untuk menguji kelancaran program yang telah dibuat. Pengujian ini penting dilakukan agar tidak terjadi kesalahan alur program yang telah dibuat. *Blackbox* testing merupakan pengujian aspek fundamental sistem tanpa memperhatikan struktur logika internal perangkat lunak. Metode ini digunakan untuk mengetahui apakah perangkat lunak berfungsi dengan benar. Pengujian *Black Box* berusaha menemukan kesalahan dalam kategori:

1. Fungsi-fungsi yang tidak benar atau hilang
2. Kesalahan interface
3. Kesalahan dalam struktur data atau akses database eksternal
4. Kesalahan kinerja
5. Inisialisasi dan kesalahan terminasi

Menurut Wahyu Nur Cholifah, Yulianingsih dan Sri Melati Sagita (2018:207) Metode *Blackbox Testing* merupakan salah satu metode yang mudah digunakan karena hanya memerlukan batas bawah dan batas atas dari data yang di harapkan, estimasi banyaknya data uji dapat dihitung melalui banyaknya field data entri yang akan diuji, aturan entri yang harus dipenuhi serta kasus batas atas dan batas bawah yang memenuhi. Dan dengan metode ini dapat diketahui jika fungsionalitas masih dapat menerima masukan data yang tidak diharapkan maka menyebabkan data yang disimpan kurang valid.

Menurut Tri Snadhika Jaya (2018:46) Black-Box Testing merupakan Teknik pengujian perangkat lunak yang berfokus pada spesifikasi fungsional dari perangkat lunak . Blackbox Testing bekerja dengan mengabaikan struktur kontrol sehingga perhatiannya difokuskan pada informasi domain. Blackbox Testing

memungkinkan pengembang software untuk membuat himpunan kondisi input yang akan melatih seluruh syarat-syarat fungsional suatu program.

Keuntungan penggunaan metode Blackbox Testing adalah (1) Penguji tidak perlu memiliki pengetahuan tentang bahasa pemrograman tertentu; (2) Pengujian dilakukan dari sudut pandang pengguna, ini membantu untuk mengungkapkan ambiguitas atau inkonsistensi dalam spesifikasi persyaratan; (3) Programmer dan tester keduanya saling bergantung satu sama lain.

Kekurangan dari metode Blackbox Testing adalah: (1) Uji kasus sulit disain tanpa spesifikasi yang jelas; (2) Kemungkinan memiliki pengulangan tes yang sudah dilakukan oleh programmer; (3) Beberapa bagian back end tidak diuji sama sekali. Saat ini terdapat banyak metoda atau teknik untuk melaksanakan Black Box Testing, antara lain: (1) Equivalence Class Partitioning; (2) Boundary Value Analysis; (3) Error Guessing.