

BAB II

LANDASAN TEORI

2.1 Tinjauan Pustaka Terkait Teknik atau Metode

2.1.1 Sistem Pendukung Keputusan

Sistem Pendukung Keputusan (SPK) atau Decision Support System (DSS) merupakan bagian dari sistem informasi berbasis komputer yang pada dasarnya digunakan untuk membantu pengambilan keputusan dalam keadaan yang semi terstruktur dan keadaan yang tidak terstruktur (Erwandi, Mulyani Sri dan Senjaya, 2018).

Sistem Pendukung Keputusan (SPK) merupakan sistem yang dapat membantu proses penilaian, perubahan kriteria, dan perubahan nilai bobot yang berguna untuk memudahkan pengambilan keputusan yang terkait untuk dapat memilih yang layak diterima atau tidak (Sasongko, Astuti dan Septya, 2017).

Sistem Pendukung Keputusan (SPK) bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi agar dapat melakukan pengambilan keputusan dengan lebih baik dengan melakukan tahapan-tahapan sebagai berikut :

1. Tahapan Pemahaman (*Intelligence Phace*)

Tahapan ini merupakan proses penelusuran dan pendeteksian dari lingkup problematika serta proses pengenalan masalah. Data masukan diperoleh, diproses, dan diuji dalam rangka mengidentifikasi masalah.

2. Tahap Perancangan (*Design Phace*)

Tahap ini merupakan proses pengembangan dan pencarian alternatif tindakan / solusi yang dapat diambil. Tersebut merupakan representasi kejadian nyata yang disederhanakan, sehingga diperlukan proses validasi dan verifikasi untuk mengetahui keakuratan model dalam meneliti masalah yang ada.

3. Tahap Pemilihan (*Choice Phace*)

Tahap ini dilakukan pemilihan terhadap diantaraberbagai alternatif solusi yang dimunculkan pada tahap perencanaan agar ditentukan / dengan memperhatikan kriteria – kriteria berdasarkan tujuan yang akan dicapai.

4. Tahap Impelementasi (*Implementation Phace*)

Tahap ini dilakukan penerapan terhadap rancangan sistem yang telah dibuat pada tahap perancangan serta pelaksanaan alternatif tindakan yang telah dipilih pada tahap pemilihan.

2.1.2 Metode AHP

Analitycal Hierarchy Process (AHP) Adalah metode untuk memecahkan suatu situasi yang kompleks tidak terstruktur kedalam beberapa komponen dalam susunan yang hirarki, dengan memberi nilai subjektif tentang pentingnya setiap variabel secara relatif, dan menetapkan variabel mana yang memiliki prioritas paling tinggi guna mempengaruhi hasil pada situasi tersebut.

Proses pengambilan keputusan pada dasarnya adalah memilih suatu alternatif yang terbaik. Seperti melakukan penstrukturan persoalan, penentuan alternatif-alternatif, penetapan nilai kemungkinan untuk variabel aleatori, penetap nilai, persyaratan preferensi terhadap waktu, dan spesifikasi atas resiko. Betapapun melebarnya alternatif yang dapat ditetapkan maupun terperinci penjajagan nilai kemungkinan, keterbatasan yang tetap melingkupi adalah dasar perbandingan berbentuk suatu kriteria yang tunggal.

Peralatan utama *Analitycal Hierarchy Process* (AHP) adalah memiliki sebuah hirarki fungsional dengan input utamanya persepsi manusia. Dengan hirarki, suatu masalah kompleks dan tidak terstruktur dipecahkan ke dalam kelomok-kelompoknya dan diatur menjadi suatu bentuk hirarki.

Kelebihan AHP dibandingkan dengan lainnya adalah :

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling dalam
2. Memperhitungkan validitas sampai dengan batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh para pengambil keputusan
3. Memperhitungkan daya tahan atau ketahanan output analisis sensitivitas pengambilan keputusan.

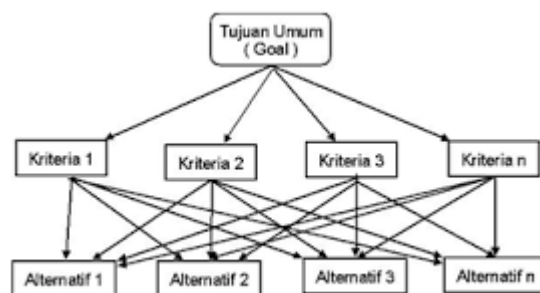
Selain itu, AHP mempunyai kemampuan untuk memecahkan masalah yang multi obyektif dan multi-kriteria yang berdasarkan pada perbandingan preferensi dari setiap elemen dalam hirarki. Jadi, model ini merupakan suatu model pengambilan keputusan yang komprehensif.

2.1.3 Prinsip Dasar Pemikiran AHP

Secara teknis dan pada dasarnya dalam metode AHP terdiri dari prinsip-prinsip dasar dalam memahami AHP. Adapun prinsip dasar tersebut menurut (Basuki & Andharini, 2016) yaitu:

1. Menyusun Hierarki

Persoalan yang akan diselesaikan, diuraikan menjadi unsur-unsurnya, yaitu kriteria dan alternatif, kemudian disusun menjadi struktur hirarki.



Gambar.2.1 Hierarki

2. Penilaian Kriteria dan Alternatif

Kriteria dan alternatif dinilai melalui perbandingan berpasangan. Untuk persoalan yang ada skala 1 sampai skala 9 adalah skala terbaik dalam mengekspresikan pendapat. Perbandingan dilakukan berdasarkan kebijakan pembuat keputusan dengan cara menilai tingkat kepentingan antara elemen yang satu dengan elemen yang lainnya

Tabel 2.1 Skala penilaian perbandingan berpasangan

Intensitas Kepentingan	Keterangan	Penjelasan
1	Kedua elemen sama Pentingnya	Dua elemen mempunyai pengaruh yang sama besar terhadap tujuan
3	Elemen yang satu sedikit lebih penting dari pada elemen yang lainnya	Pengalaman dan penilaian sedikit menyokong satu elemen dibandingkan elemen lainnya
5	Elemen yang satu lebih penting dari pada elemen yang lainnya	Pengalaman dan penilaian sangat kuat menyokong satu elemen dibandingkan elemen lainnya
7	Elemen yang satu jelas lebih mutlak penting daripada yang lainnya	Satu elemen yang kuat disokong dan dominan terlihat dalam praktek
9	Satu elemen mutlak penting daripada elemen lainnya	Bukti yang mendukung elemen yang satu terhadap elemen lain memiliki tingkat penegasan tertinggi yang mungkin menguatkan
2,4,6,8	Nilai-nilai antara dua nilai	Nilai ini diberikan bila

	pertimbangan- pertimbangan yang berdekatan	ada dua kompromi di antara dua pilihan
Kebalikan	Jika aktifitas i mendapat satu angka dibandingkan dengan aktifitas j , maka j memiliki kebalikannya dibandingkan dengan i	

3. Penentuan Prioritas

Untuk setiap kriteria dan alternatif, perlu dilakukan perbandingan berpasangan. Nilai-nilai perbandingan relatif kemudian diolah untuk menentukan peringkat alternatif dari seluruh alternatif. Baik kriteria kualitatif maupun kuantitatif dapat dibandingkan sesuai dengan penilaian yang telah ditentukan untuk menghasilkan suatu bobot dan prioritas. Bobot atau prioritas ini dihitung dengan manipulasi matriks atau dengan penyelesaian matematik.

4. Konsistensi Logis

Konsistensi logis menurut (Kusrini, 2016) memiliki dua makna. Pertama, objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua, menyangkut hubungan antar objek yang didasarkan pada kriteria tertentu. (Handayani, 2015) “Nilai rasio konsistensi harus 10% atau kurang. Pada referensi yang lain menyebutkan bahwa hasil perhitungan nilai inkonsistensi antara 0 hingga 1. Jika lebih dari 10%, pertimbangan yang telah dibuat mungkin agak acak dan mungkin perlu untuk diperbaiki”. (Malik & Haryanti, 2018) “Inkonsistensi ini dapat disebabkan oleh kesalahan memasukkan penilaian, kurangnya informasi, kurangnya konsentrasi, dunia nyata yang tidak selalu konsisten, atau model struktur hirarki yang kurang sesuai”.

2.1.4 Klasterisasi Perguruan Tinggi

Suatu Proses pengelompokan perguruan tinggi berdasarkan kinerja perguruan tinggi yaitu dengan melihat kinerja masukan dengan bobot 40% yang meliputi kinerja input (15%) dan proses (25%), serta kinerja luaran dengan bobot 60% yang meliputi Kinerja Output (25%) dan Outcome (35%) dari indikator tersebut perguruan tinggi dapat diklasterisasikan ke 5 klaster yaitu sangat Sehat, Sehat, Cukup Sehat, Tidak Sehat dan sangat Tidak Sehat.

- Kinerja Input terdiri dari Jumlah Dosen S3, Jumlah Guru Besar, Rasio Dosen, Mahasiswa Asing dan Dosen Asing.
- Kinerja Proses terdiri dari APT, APS, E-Learning, Kerjasama, Ketaatan PDDIKTI, Laporan Keuangan.
- Kinerja Output terdiri dari Jumlah Artikel, Kinerja Riset, Kinerja Mahasiswa dan Akreditasi Internal.
- Kinerja Outcome terdiri dari Inovasi, daya Serap Mahasiswa, Sitasi, Hak Paten dan pengabdian Masyarakat.

2.2 Basis Data (*Database*)

Basis data adalah kumpulan data yang saling berhubungan secara logis dan didesain untuk mendapatkan data yang dibutuhkan oleh suatu organisasi. (Indrajani, 2015). Sistem basis data adalah sistem terkomputerisasi yang tujuan utamanya adalah memelihara data yang sudah diolah atau informasi dan membuat informasi tersedia saat dibutuhkan. (Rosa dan Shalahuddin, 2015).

2.3 Bahasa Pemrograman dan Perangkat Lunak Pendukung

Bahasa pemrograman dan perangkat lunak pendukung yang digunakan dalam penulisan ini adalah antara lain *Website, XAMPP, PHP, HTML, CSS, dan JavaScript*.

2.3.1 *Website*

Website atau disingkat web, dapat diartikan sekumpulan halaman yang terdiri dari beberapa laman yang berisi informasi dalam bentuk data digital baik berupa text, gambar, video, audio, dan animasi lainnya yang disediakan melalui jalur koneksi internet. (Abdulloh, 2015).

2.3.2 XAMPP

Xampp adalah sebuah paket kumpulan software yang terdiri dari *Apache*, *MySQL*, *PhpMyAdmin*, *PHP*, *Perl*, *Filezilla*, dan lainnya.” Fungsi dari *Xampp* adalah untuk memudahkan instalasi lingkungan PHP, di mana biasanya lingkungan pengembangan web memerlukan *PHP*, *Apache*, *MySQL*, dan *PhpMyAdmin*. (MADCOMS, 2016).

Xampp dikembangkan oleh perusahaan *apache friends* yang memiliki kelebihan bisa berperan sebagai *Server web Apache* untuk simulasi pengembangan website. *Tool* pengembangan web ini mendukung teknologi web populer seperti PHP, MySQL, dan Perl. Dengan menggunakan perangkat lunak XAMPP pengembang web dapat mengembangkan web berbasis *database* secara mudah.

2.3.3 PHP

PHP singkatan dari *Hypertext Preprocessor* yang merupakan *server-side programming*, yaitu bahasa pemrograman yang diproses di sisi server.” Fungsi utama PHP dalam membangun website adalah untuk melakukan pengolahan data pada database. Data website akan dimasukkan ke database, diedit, dihapus, dan ditampilkan pada website yang diatur oleh PHP. (Rohi Abdulloh, 2015).

2.3.4 HTML

HTML singkatan dari *Hypertext Markup Language*, yaitu script yang berupa tag-tag untuk membuat dan mengatur struktur website. (Rohi Abdulloh, 2015).

Beberapa tugas HTML dalam membangun website diantaranya, sebagai berikut :

- a. Menentukan layout website.
- b. Memformat text dasar seperti pengaturan paragraph, dan format font.
- c. Membuat list dan formulir.
- d. Membuat tabel, gambar, video, audio, dan link

2.3.5 CSS

Cascading Style Sheets yaitu skrip yang digunakan untuk mengatur desain website.” Walaupun HTML mempunyai kemampuan untuk mengatur tampilan website, namun kemampuannya sangat terbatas. Fungsi CSS adalah memberikan pengaturan yang lebih lengkap agar struktur website yang dibuat dengan HTML terlihat lebih indah. (Rohi Abdulloh, 2015)

2.3.6 *Java Script*

JavaScript ialah suatu bahasa *scripting* yang digunakan sebagai fungsionalitas dalam membuat suatu *web*. (Hidayatullah dan Kawistara, 2015).

2.3.7 MySQL

MySQL (*My Structure Query Language*) adalah salah satu jenis database server yang sangat terkenal dan banyak digunakan untuk membangun aplikasi web yang menggunakan database sebagai sumber dan pengelolaan datanya. (Arief, 2011).

2.4 UML (Unified Modeling Language)

UML (Unified Model Language) merupakan alat industri standar yang memungkinkan kita untuk mengkomunikasikan dengan jelas tentang kebutuhan, arsitektur dan desain. UML merupakan salah satu alat yang paling berguna dan paling populer di dalam dunia pengembangan sistem. Hal ini disebabkan karena UML adalah sebuah metode pemodelan visual yang memungkinkan kepada pengembang untuk membuat sebuah *blueprints* yang dapat di mengerti, sehingga hasil blueprint ini dapat dimengerti antara satu dengan yang lainnya. (Muhammad Taufik, 2015).

Widodo dan Herlawati (2011), “UML diaplikasikan untuk maksud tertentu”, biasanya antara lain untuk:

1. Merancang perangkat lunak.
2. Sarana komunikasi antara perangkat lunak dengan proses bisnis.
3. Menjabarkan sistem secara rinci untuk analisa dan mencari apa yang diperlukan sistem.

4. Mendokumentasi sistem yang ada, proses-proses dan organisasinya.

Tabel 2.1 Tipe Diagram UML (*Unified Modelling Language*)

No.	Diagram	Tujuan
1	<i>Class</i>	Memperlihatkan himpunan kelas-kelas, antarmukaantarmuka, kolaborasi-kolaborasi, serta relasi-relasi
2	<i>Package</i>	Memperlihatkan kumpulan kelas-kelas, merupakan dari diagram komponen
3	<i>Use Case</i>	Diagram ini memperlihatkan himpunan use case dan aktor-aktor (suatu jenis khusus dari kelas)
4	<i>Sequence</i>	Diagram interaksi yang menekankan pada pengiriman pesan dalam suatu waktu tertentu
5	<i>Communication</i>	Sebagai pengganti diagram kolaborasi UML 1.4 yang menekankan organisasi struktural dari obyek-obyek yang menerima serta mengirim pesan
6	<i>Statechart</i>	Diagram status memperlihatkan keadaan-keadaan pada sistem, memuat status (state), transisi, kejadian serta aktivitas
7	<i>Activity</i>	Tipe khusus dari diagram status yang memperlihatkan aliran dari suatu aktivitas ke aktivitas lainnya dalam suatu sistem
8	<i>Component</i>	Memperlihatkan organisasi serta kebergantungan sistem atau perangkat lunak pada komponenkomponen yang telah ada sebelumnya
9	<i>Deployment</i>	Memperlihatkan konfigurasi saat aplikasi dijalankan (run-time)

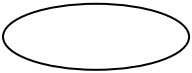
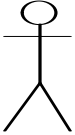

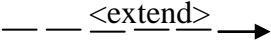
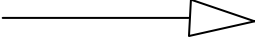
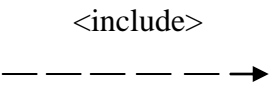
2.4.1 Jenis-jenis Diagram UML (*Unified Modelling Language*)

2.4.1.1 Use Case Diagram

Use case diagram merupakan pemodelan untuk kelakuan (behavior) sistem informasi yang akan dibuat. Use case mendeskripsikan sebuah interaksi antara satu atau lebih aktor dengan sistem informasi yang akan dibuat. (Sukamto dan Shalahuddin, 2013).

Berikut adalah simbol-simbol yang ada pada diagram *use case* :

Table 2.2 Simbol *Use Case Diagram*



Simbol	Deskripsi
Use Case 	Fungsional yang disediakan sistem sebagai unit-unit yang paling saling bertukar pesan antar unit atau aktor: biasanya dinyatakan dengan menggunakan kata kerja di awal frase nama use case
Aktor / actor 	Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat diluar sistem informasi yang akan itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang tapi aktor belum merupakan orang, biasanya dinyatakan menggunakan kata benda di awal frase nama aktor
Asosiasi / association 	Komunikasi antar aktor dan use case yang berpartisipasi pada use case atau use case memiliki interaksi dengan aktor
Ekstensi / extend 	Case tambahan memiliki nama depan yang sama dengan use case yang ditambahkan
Generalisasi/generalization 	Hubungan generalisasi dan spesialisasi (umum-khusus) antara dua buah use case dimana fungsi satu adalah fungsi yang lebih umum dari lainnya
Menggunakan/include/uses 	Fungsinya atau sebagai syarat dijalankan use case ini.

2.4.1.2 Activity Diagram

Activity diagram menggambarkan workflow (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak”. Yang perlu diperhatikan disini adalah bahwa diagram aktivitas menggambarkan aktivitas sistem bukan apa yang dilakukan aktor, jadi aktivitas yang dapat dilakukan oleh sistem. (Sukamto dan Shalahuddin,2013).

Adapun simbol-simbol yang digunakan dalam *activity diagram* adalah sebagai berikut:



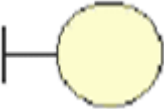


Table 2.3 Simbol *Activity Diagram*


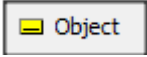


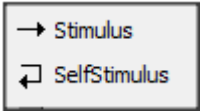
Keterangan	Simbol	Deskripsi
Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
Percabangan		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
Penggabungan		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.
Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.

2.4.1.3 Sequence Diagram

Diagram sequence menggambarkan kelakuan objek pada use case dengan mendeskripsikan waktu hidup objek dan message yang dikirimkan dan diterima antar objek”. Oleh karena itu untuk menggambarkan diagram sequence maka harus diketahui objek-objek yang terlihat dalam sebuah use case beserta metode-metode yang dimiliki kelas yang diinstansiasi menjadi objek itu. *Sequence diagram* menunjukkan urutan event kejadian dalam suatu waktu. Komponen sequence diagram terdiri atas obyek yang dituliskan dengan kotak segiempat bernama. Message diwakili oleh garis dengan tanda panah dan waktu yang ditunjukkan dengan progress vertikal. (Sukamto dan Shalahuddin, 2013). Berikut simbol-simbol yang digunakan dalam sequence diagram adalah:

Table 2.4 Simbol *Sequence Diagram*

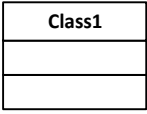
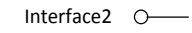
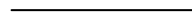
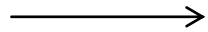
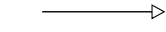
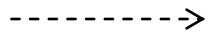
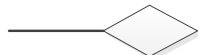
No.	Symbol	Keterangan
1.		Menggambarkan orang yang berinteraksi dengan sistem
2.		Menggambarkan hubungan kegiatan yang akan dilakukan
3.		Menggambarkan sebuah penggambaran dari form
4.		Menggambarkan penghubung antara boundary dengan tabel
5.		Menggambarkan tempat mulai dan berakhirnya sebuah pesan

6.	Line message 	Menggambarkan pengiriman pesan
7.	Object 	Menyatakan objek yang berinteraksi pesan
8.	Garis hidup 	Menyatakan kehidupan suatu objek
9.	Waktu aktif 	Menyatakan objek dalam keadaan aktif dan berinteraksi, semua yang terhubung dengan waktu aktif ini adalah sebuah tahapan yang dilakukan di dalamnya actor tidak memiliki waktu aktif
10.	Stimulus 	Menyatakan suatu objek mengirimkan pesan untuk menjalankan operasi yang ada pada objek lain

2.4.1.4 Class Diagram

Class diagram menggambarkan struktur sistem dari segi pendefinisian kelas-kelas yang akan dibuat untuk membangun sistem”. Diagram kelas dibuat agar pembuat program atau programmer membuat kelas-kelas sesuai rancangan didalam diagram kelas agar antara dokumentasi perancangan dan perangkat lunak sinkron.(Sukamto dan Shalahuddin, 2013). Berikut simbol-simbol yang digunakan dalam class diagram adalah sebagai berikut:

Tabel 2.5 Simbol *Class Diagram*

Simbol	Deskripsi
<p>Kelas</p> 	Kelas pada struktur sistem.
<p>Natarmuka/<i>interface</i></p> 	Sama dengan konsep <i>interface</i> dalam pemograman berorientasi objek.
<p>Asosiasi</p> 	Relasi antar kelas dalam makna umum, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Asosiasi berarah</p> 	Relasi antar kelas dengan makna kelas yang satu digunakan oleh kelas yang lain, asosiasi biasanya juga disertai dengan <i>multiplicity</i> .
<p>Generalisasi</p> 	Relasi antar kelas dengan makna generalisasi-spesialisasi (umum-khusus).
<p>Kebergantungan</p> 	Relasi antar kelas dengan makna kebergantungan antar kelas.
<p>Agregasi</p> 	Relasi antar kelas dengan makna semua bagian (<i>whole-part</i>).

2.5 RUP (*Rational Unified Process*)

RUP (*Rational Unified Process*) adalah pendekatan pengembangan perangkat lunak yang dilakukan berulang-ulang (*iterative*), fokus pada arsitektur (*architecture-centric*), lebih diarahkan berdasarkan penggunaan kasus (*use case driven*). RUP merupakan proses rekayasa perangkat lunak dengan pendefinisian yang baik (*well defined*) dan penstrukturan yang baik (*well structured*). RUP menyediakan pendefinisian struktur yang baik untuk alur hidup proyek perangkat lunak. (Sukamto dan Shalahuddin,2013).

Relational Unified Process (RUP) memiliki 4 tahap atau fase yang dapat dilakukan pula secara iterative. Berikut adalah penjelasan untuk setiap fase RUP :

1. *Inception* (Permulaan)

Tahap ini lebih pada pemodelan proses bisnis yang dibutuhkan (*business modeling*) dan mendefinisikan kebutuhan akan sistem yang akan dibuat (*requirements*).

2. *Elaboration* (Perluasan/Perencanaan)

Tahap ini lebih difokuskan pada perencanaan arsitektur sistem. Tahap ini juga dapat mendeteksi apakah arsitektur sistem yang diinginkan dapat dibuat atau tidak. Mendeteksi risiko yang mungkin terjadi dari arsitektur yang dibuat. Tahap ini lebih analisis dan desain sistem serta implementasi sistem yang fokus pada sistem (*prototype*).

3. *Construction* (Konstruksi)

Tahap ini fokus pada pengembangan komponen dan fitur – fitur sistem. Tahap ini lebih pada implementasi dan pengujian sistem yang fokus pada implementasi perangkat lunak pada kode program.

4. *Transition* (Transisi)

Tahap ini lebih pada *deployment* atau instalasi sistem agar dapat dimengerti oleh *user*. Aktivitas pada tahap ini termasuk pada pelatihan *user*, pemeliharaan dan pengujian sistem apakah sudah memenuhi harapan *user*.