

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Sistem Penunjang Keputusan

##### 2.1.1 Konsep Sistem Penunjang Keputusan

Sistem merupakan kumpulan sub-sub sistem (elemen) yang saling berkorelasi satu dengan yang lainnya untuk mencapai tujuan tertentu. Sistem penunjang keputusan dapat diartikan sebagai suatu sistem yang dirancang yang digunakan untuk mendukung manajemen didalam pengambilan keputusan.

Pada awal 1970-an, Scott Morton pertama kali mengungkapkan konsep utama Sistem Penunjang Keputusan (SPK). Ia mendefinisikan SPK sebagai Sistem berbasis komputer interaktif, yang membantu pembuat keputusan dalam menggunakan data dan model untuk memecahkan masalah yang tidak terstruktur. Pada tahun 1978 Keen dan Scott Morton kembali mendefinisikan SPK adalah Sistem Penunjang Keputusan menyediakan sumber daya intelektual individu dengan kemampuan berbasis komputer untuk meningkatkan kualitas keputusan dan membantu manajemen dalam menangani masalah semi terstruktur [2]. Menurut Herbert A. Simon, model pengambilan keputusan dimulai dari tahap pemahaman (*intelligence phase*), tahap perancangan (*design phase*), tahap pemilihan (*choice phase*) dan tahap implementasi (*implementation phase*) [2].

Dari definisi di atas dapat disimpulkan bahwa Sistem Penunjang Keputusan adalah suatu sistem informasi spesifik yang ditujukan untuk membantu manajemen dalam mengambil keputusan yang berkaitan dengan persoalan atau masalah yang bersifat semi terstruktur dan tidak terstruktur. Sistem ini memiliki fasilitas untuk menghasilkan berbagai alternatif yang secara interaktif digunakan oleh pemakai [3].

Manfaat dari sistem penunjang keputusan adalah menyediakan mekanisme untuk membuat keputusan yang lebih tepat, resolusi yang tepat waktu dan efisiensi yang lebih besar dalam menangani masalah yang seimbang dengan pendekatan organisasi dan sosial. Dengan adanya penanganan, pengumpulan dan analisis data

yang mudah, memungkinkan pengguna akhir untuk membuat keputusan yang lebih tepat dengan lebih cepat.

### 2.1.2 Komponen Sistem Penunjang Keputusan

Sistem pendukung keputusan dibangun oleh empat komponen yaitu subsistem manajemen data, subsistem manajemen model, subsistem antar muka pengguna, dan subsistem manajemen berbasis pengetahuan.

Subsistem yang membangun sistem pendukung keputusan dijelaskan sebagai berikut [2]:

1. Subsistem Manajemen Data (*Data Management Subsystem*)

Termasuk *database*, yang mengandung data yang relevan untuk pelbagai situasi dan diatur oleh software yang disebut *Database Management Systems* (DBMS).

2. Subsistem Manajemen Model (*Model Management Subsystem*)

Melibatkan model finansial, statistikal, *management science*, atau berbagai model kuantitatif lainnya, sehingga dapat memberikan ke sistem suatu kemampuan analitis, dan manajemen *software* yang diperlukan.

3. Subsistem Antarmuka Pengguna (*User Interface Subsystem*)

*User* dapat berkomunikasi dan memberi perintah pada DSS melalui subsistem ini. Ini berarti menyediakan antarmuka.

4. Subsistem Manajemen Pengetahuan (*Knowledge-based Management Subsystem*)

Subsistem optional ini dapat mendukung subsistem lain atau bertindak sebagai komponen yang berdiri sendiri

### 2.1.3 Karakteristik Sistem Penunjang Keputusan

Menurut Turban beberapa karakteristik SPK yang membedakan dengan sistem informasi lainnya adalah [2]:

1. Mendukung pengambil keputusan, terutama pada situasi semi terstruktur dan tidak terstruktur dengan memadukan pertimbangan manusia dan informasi terkomputerisasi.
2. Mendukung semua level manajerial, dari eksekutif puncak sampai manajer lapangan.
3. Mendukung individu dan kelompok. Masalah yang kurang terstruktur sering memerlukan keterlibatan individu dari departemen dan tingkat organisasional yang berbeda atau bahkan dari organisasi lain.
4. Mendukung semua keputusan independen dan sekuensial. Keputusan bisa dibuat satu kali, beberapa kali, atau berulang (dalam interval yang sama).
5. Mendukung di semua fase proses pengambilan keputusan: intelegensi, desain, pilihan, dan implementasi.
6. Mendukung pada berbagai proses dan gaya pengambilan keputusan.
7. Adaptasi sepanjang waktu. Pengambilan keputusan harus reaktif, dapat menghadapi perubahan kondisi secara tepat dan dapat mengadaptasikan SPK untuk memenuhi perubahan tersebut.
8. Pengguna merasa nyaman dengan sistem. *User-friendly*, dukungan grafis yang baik dan bahasa antar muka yang sesuai sehingga dapat meningkatkan efektivitas SPK.
9. Meningkatkan efektivitas pengambilan keputusan (akurasi, *timelines*, kualitas) ketimbang pada efisiensi (biaya pengambilan keputusan).
10. Pengambil keputusan memiliki kontrol penuh terhadap semua langkah proses pengambilan keputusan dalam memecahkan masalah.
11. Pengguna akhir dapat mengembangkan dan memodifikasi sistem sendiri.
12. Menggunakan model-model dalam penganalisisan situasi pengambilan keputusan.

13. Menyediakan akses untuk berbagai sumber data, format dan tipe, mulai dari sistem informasi geografis (GIS) sampai sistem berorientasi objek.
14. Dapat dilakukan sebagai alat *standalone* yang digunakan oleh seorang pengambil keputusan pada satu lokasi atau di distribusikan di satu organisasi keseluruhan dan di beberapa organisasi sepanjang rantai persediaan

## 2.2 Metode Penunjang Keputusan

*Multi Criteria Decision Making* adalah salah satu metode yang paling baik membantu proses pengambilan keputusan yang memiliki banyak kriteria. MCDM dibagi menjadi *Multi-Objective Decision Making* (MODM) dan *Multi-Attribute Decision Making* (MADM). Namun, sering kali istilah MADM dan MCDM digunakan untuk mengartikan kelas model yang sama. MCDM / MADM berkonsentrasi pada masalah dengan ruang keputusan yang berbeda [4]. Dalam masalah ini sekumpulan alternatif keputusan telah ditentukan sebelumnya. Dalam menyelesaikan masalah MCDM ada beberapa metode yang digunakan, yaitu *Simple Additive Weighting Method* (SAW), *Weighted Product* (WP), *ELECTRE*, *Technique for Order Preference by Similarity to Ideal Solution* (TOPSIS), *Analytic hierarchy Process* (AHP).

### 2.2.1 Pengertian Metode *Analytic Hierarchy Process* (AHP)

Dalam pengambilan keputusan pada tahap pemilihan (*choice phase*) diperlukan metode untuk memilih alternatif terbaik dari beberapa kriteria. Metode yang dibutuhkan dapat menyelesaikan masalah multi kriteria. Salah satu metode untuk menyelesaikan masalah multi kriteria adalah metode *Analytic hierarchy Process* (AHP).

*Analytic Hierarchy Process* (AHP) dikembangkan oleh T. L. Saaty berdasarkan pengalamannya pada 1970an saat melakukan proyek penelitian pada Badan Pengendalian dan Senjata Amerika Serikat. Dari pengembangan Saaty ditemukan bahwa dibutuhkan metodologi yang mudah dipahami dan diterapkan untuk pengambilan keputusan yang kompleks. Metode AHP merupakan metode yang digunakan dalam pengambilan keputusan untuk masalah yang kompleks, tidak terstruktur dan multi kriteria yang akan diuraikan ke dalam suatu hierarki [5]

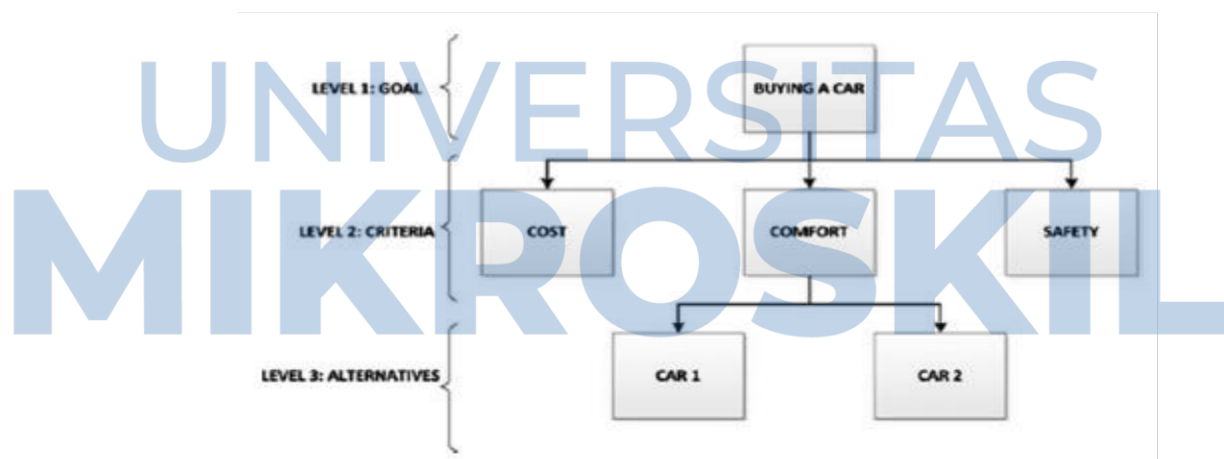
Hirarki didefinisikan sebagai suatu representasi dari sebuah permasalahan yang kompleks dalam suatu struktur multilevel dimana level tertinggi adalah tujuan, yang diikuti level subtujuan, kriteria, sub kriteria, dan seterusnya ke bawah hingga level terendah berupa alternatif. Dengan hirarki, suatu masalah yang kompleks dapat diuraikan ke dalam kelompok-kelompoknya yang kemudian diatur menjadi suatu bentuk hirarki sehingga permasalahan akan tampak lebih terstruktur dan sistematis [5]

### 2.2.2 Langkah-Langkah Metode AHP

Adapun langkah penyelesaian suatu masalah menggunakan metode *Analytic hierarchy Process* (AHP) yaitu [5]:

1. Langkah pertama dalam analisis AHP adalah membangun hierarki untuk menganalisis keputusan. Proses hierarki AHP adalah menyusun masalah sebagai hierarki.

Berikut adalah contoh struktur hierarki dalam proses membeli sebuah mobil metode AHP:



Gambar 2. 1 Hierarki AHP

Pada level 1 adalah akar dari hierarki yang merupakan tujuan atau sasaran dari masalah yang dianalisis, pada gambar diatas tujuannya adalah membeli mobil. Pada level 2 merupakan kriteria yang akan kita gunakan untuk memutuskan pembelian. Dalam contoh diatas, terdapat tiga kriteria yaitu: biaya, kenyamanan, dan keamanan.

Pada level 3 adalah alternatif yang jumlahnya bisa lebih dari satu (sebanyak  $n$ ), dalam contoh terdapat 2 alternatif, yaitu mobil 1 dan mobil 2.

Keuntungan dari dekomposisi hierarki ini jelas, yaitu dengan menyusun masalah dengan cara ini dimungkinkan untuk lebih memahami keputusan yang akan dicapai, kriteria yang akan digunakan, dan alternatif yang akan dievaluasi. Langkah ini sangat penting dan untuk masalah yang lebih kompleks, dimungkinkan meminta partisipasi ahli untuk memastikan bahwa semua kriteria dan alternatif yang mungkin telah dipertimbangkan. Perhatikan juga bahwa dalam masalah yang kompleks mungkin perlu menyertakan level tambahan dalam hierarki seperti sub-kriteria.

- Langkah kedua dalam proses AHP adalah mendapatkan prioritas relatif (bobot) untuk kriteria. Itu disebut relatif karena prioritas kriteria yang diperoleh diukur satu sama lain dan tidak semua kriteria sama pentingnya dalam waktu tertentu. Untuk mendapatkan prioritas untuk kriteria dilakukan perbandingan berpasangan dengan menyusun kriteria-kriteria tersebut dalam bentuk matriks perbandingan berpasangan, perbandingan dilakukan berdasarkan pilihan dari pembuat keputusan dengan menilai tingkat kepentingan suatu elemen dibanding elemen lainnya menggunakan skala 1 sampai 9 yang telah ditetapkan oleh Saaty seperti tabel di bawah ini:

Tabel 2. 1 Intensitas Kepentingan

Intensitas kepentingan	Keterangan
1	Jika dua kriterian sama penting
3	Jika "i" secara lemah lebih penting dari pada "j"
5	Jika "i" secara kuat lebih penting dari pada "j"
7	Jika "i" secara sangat kuat lebih penting dari pada "j"

9	Jika "i" secara absolut lebih penting dari pada "j"
2, 4, 6, 8	Nilai-nilai antara dua nilai pertimbangan yang berdekatan
Kebalikan	Jika aktivitas "i" mendapat suatu angka dibandingkan dengan aktivitas "j", maka "j" memiliki nilai kebalikannya dibandingkan dengan "i"

Berikut adalah contoh matriks perbandingan berpasangan antar kriteria dalam membeli mobil:

Buying a car	Cost	Comfort	Safety
Cost	1	7	3
Comfort	1/7	1	1/3
Safety	1/3	3	1

Gambar 2. 2 Contoh Matriks Perbandingan Berpasangan

Berdasarkan contoh diatas, misalnya pengambil keputusan menganggap bahwa biaya sangat lebih penting daripada faktor kenyamanan, sel perbandingan biaya-kenyamanan (yaitu, perpotongan baris "biaya" dan kolom "kenyamanan") akan berisi nilai 7. Secara matematis ini berarti bahwa rasio kepentingan biaya versus kepentingan kenyamanan adalah tujuh ( $\text{biaya/kenyamanan} = 7$ ). Karena itu, perbandingan sebaliknya, pentingnya kenyamanan relatif terhadap pentingnya biaya, akan menghasilkan kebalikan dari nilai ini ( $\text{kenyamanan/biaya} = 1/7$ ) seperti yang ditunjukkan pada sel biaya kenyamanan dalam gambar matriks perbandingan diatas. Demikian pula, jika pengambil keputusan menganggap bahwa biaya lebih penting daripada keselamatan ( $\text{biaya/keselamatan} = 3$ ), kita akan memasukkan 3 dalam sel biaya-keselamatan dan sel biaya-keamanan akan berisi  $1/3$  ( $\text{keamanan/biaya} = 1/3$ ). Terakhir, jika pengambil keputusan merasa bahwa keselamatan lebih penting daripada kenyamanan, sel keamanan-kenyamanan akan berisi nilai 3 dan sel kenyamanan-keamanan akan memiliki kebalikan  $1/3$ . Perhatikan dalam matriks perbandingan diatas, bahwa ketika pentingnya suatu kriteria dibandingkan dengan dirinya sendiri; misalnya, biaya versus biaya, kenyamanan versus kenyamanan, atau keamanan versus

keselamatan; nilai input adalah 1 yang sesuai dengan intensita sama pentingnya dalam skala Saaty.

Untuk mendapatkan nilai prioritas dari setiap kriteria, dibutuhkan normalisasi matriks perbandingan, yaitu dengan menjumlahkan nilai dari setiap kolom seperti dibawah ini:

Buying a car	Cost	Comfort	Safety
Cost	1.000	7.000	3.000
Comfort	0.143	1.000	0.333
Safety	0.333	3.000	1.000
Sum	1.476	11.000	4.333

Gambar 2. 3 Contoh Total Kolom Matriks Perbandingan Berpasangan

Selanjutnya, membagi setiap nilai dari sel dengan total kolom yang bersangkutan untuk memperoleh normalisasi matriks, berikut adalah hasil normalisasi matriks:

Buying a car	Cost	Comfort	Safety
Cost	0.677	0.636	0.692
Comfort	0.097	0.091	0.077
Safety	0.226	0.273	0.231

Gambar 2. 4 Contoh Normalisasi Matriks

Dari normalisasi matriks diatas, diperoleh prioritas kelesuruhan kriteria dengan menghitung nilai rata-rata setiap baris (misalnya untuk baris biaya:  $0,677 + 0,636 + 0,692)/3 = 0,669$ ), seperti dibawah ini:

Buying a car	Cost	Comfort	Safety	Priority
Cost	0.677	0.636	0.692	<b>0.669</b>
Comfort	0.097	0.091	0.077	<b>0.088</b>
Safety	0.226	0.273	0.231	<b>0.243</b>

Gambar 2. 5 Contoh Prioritas Keseluruhan Kriteria

3. Mengukur konsistensi setiap matriks perbandingan. Menghitung *Consistency Index* (CI) dengan rumus:  $CI = (\lambda_{maks} - n)/(n-1)$

Dimana:  $\lambda_{maks}$  adalah nilai konsistensi vektor; n adalah banyak elemen.

4. Menghitung *Consistency Ratio* (CR)

Dengan rumus  $CR = CI / RI$



Dimana  $CI = Consistency Index$ ;  $RI = Index Random Consistency$  (nilai yang berasal dari tabel random)

5. Memeriksa konsistensi hierarki dengan tabel dibawah ini

Tabel 2. 2 Daftar Index Random Konsistensi

Ukuran matriks	Index Random (RI)
1, 2	0,000
3	0,5245
4	0,8815
5	1,1086
6	1,2479
7	1,3417
8	1,4056
9	1,4499
10	1,4854

Jika  $CR < 0,1$  maka nilai perbandingan berpasangan pada matriks kriteria yang diberikan konsisten. Jika  $CR > 0,1$  maka nilai perbandingan berpasangan pada matriks kriteria yang diberikan tidak konsisten. Sehingga jika tidak konsisten, maka pengisian nilai-nilai pada matriks berpasangan pada unsur kriteria maupun alternatif harus diulang.

6. Hasil akhirnya berupa prioritas global sebagai nilai yang digunakan oleh pengambil keputusan berdasarkan skor tertinggi.

## 2.3 Metode Pengembangan Sistem

### 2.3.1 *Rapid Application Development (RAD)*

*Rapid Application Development (RAD)* merupakan model yang dikembangkan oleh James Martin pada tahun 1991. Menurut Kendall, RAD adalah pendekatan berorientasi objek terhadap pengembangan sistem serta perangkat-perangkat lunak. RAD memfokuskan pada kecepatan dalam pengembangan sistem untuk memenuhi kebutuhan pengguna dengan membangun *prototype* yang disempurnakan secara bersama oleh pengguna dan pengembang dari waktu ke waktu. RAD bertujuan mempersingkat waktu yang biasanya diperlukan dalam siklus hidup pengembangan sistem tradisional antara perancangan dan penerapan suatu sistem informasi [1].

Menurut Kendall, ada tiga fase dalam RAD yaitu [1]:

1. *Requirement Planning*, dalam tahap ini pengguna dan penganalisis bertemu untuk mengidentifikasi tujuan-tujuan aplikasi atau sistem serta untuk mengidentifikasi syarat-syarat informasi yang ditimbulkan dari tujuan-tujuan tersebut. Orientasi dalam fase ini adalah menyelesaikan masalah-masalah perusahaan. Meskipun teknologi informasi dan sistem bisa mengarahkan sebagian dari sistem yang diajukan, fokusnya akan selalu tetap pada upaya pencapaian tujuan-tujuan perusahaan.
2. *Design Workshop*, Fase ini adalah fase untuk merancang dan memperbaiki yang bisa digambarkan sebagai *workshop*. Penganalisis dan pemrogram dapat bekerja membangun dan menunjukkan representasi visual desain dan pola kerja kepada pengguna. *Workshop* desain ini dapat dilakukan selama beberapa hari tergantung dari ukuran aplikasi yang akan dikembangkan. Selama *workshop* desain RAD, pengguna merespon prototipe yang ada dan penganalisis memperbaiki modul-modul yang dirancang berdasarkan respon pengguna. Apabila seorang pengembangnya merupakan pengembang atau pengguna yang berpengalaman, Kendall menilai bahwa usaha kreatif ini dapat mendorong pengembangan sampai pada tingkat terakselerasi.
3. *Implementation*, Pada fase implementasi ini, penganalisis bekerja dengan para pengguna secara intens selama *workshop* dan merancang aspek-aspek bisnis dan

non-teknis perusahaan. Segera setelah aspek-aspek ini disetujui dan sistem-sistem dibangun dan disaring, sistem-sistem baru atau bagian dari sistem diujicoba dan kemudian diperkenalkan kepada organisasi.







## 2.4 Metode Pemodelan Sistem

### 2.4.1 Unified Modelling Language (UML)

*Unified Modelling Language* (UML) adalah sebuah “bahasa” pemodelan yang memiliki pembendaharaan kata dan cara untuk mempresentasikan secara fokus pada konseptual dan fisik dari suatu sistem atau perangkat lunak [6]. Pemodelan (*modelling*) sesungguhnya digunakan untuk penyerhanaan permasalahan-permasalahan yang kompleks sedemikian rupa sehingga lebih mudah dipelajari dan dipahami.

#### 2.4.1.1 Use Case Diagram







*Use case* atau diagram *use case* 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. *Use case* digunakan untuk mengetahui fungsi apa saja yang ada di dalam sebuah sistem informasi dan siapa saja yang berhak menggunakan fungsi-fungsi itu. Berikut adalah simbol-simbol yang ada pada *Use Case Diagram* [7].

No	Nama Komponen	Simbol	Deskripsi
1	<i>Use Case</i>		Fungsionalitas yang disediakan sistem sebagai unit-unit yang saling bertukar pesan antar unit atau aktor.
2	Aktor/Aktor		Orang, proses, atau sistem lain yang berinteraksi dengan sistem informasi yang akan dibuat di luar sistem informasi yang akan dibuat itu sendiri, jadi walaupun simbol dari aktor adalah gambar orang, tapi aktor belum tentu merupakan orang, biasanya digunakan menggunakan kata benda di awal / <i>use</i> nama aktor.
3	<i>Asosiasi/ Association</i>		Komunikasi antara aktor dan <i>use case</i> yang berpartisipasi pada <i>use case</i> atau <i>use case</i> memiliki interaksi dengan aktor.
4	<i>Include</i>		Relasi <i>use case</i> tambahan ke seluruh <i>use case</i> dimana <i>use case</i> yang ditambahkan memerlukan <i>use case</i> ini untuk menjalankan fungsi.
5	<i>Generalisasi/ Generalization</i>		Hubungan generalisasi antara dua buah <i>use case</i> dimana fungsi yang satu adalah fungsi yang lebih umum dari lainnya.
6	Ekstensi / <i>Extend</i>		Relasi <i>use case</i> tambahan ke seluruh <i>use case</i> dimana <i>use case</i> yang ditambahkan dapat berdiri sendiri walau tanpa <i>use case</i> tambahan, biasanya <i>use case</i> tambahan memiliki nama depan yang sama dengan <i>use case</i> yang ditambahkan, misal arah panah menguraiah pada <i>use case</i> yang ditambahkan.

Gambar 2. 6 Simbol Use Case Diagram

### 2.4.1.2 Activity Diagram

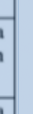





Diagram aktivitas atau *activity diagram* menggambarkan *workflow* (aliran kerja) atau aktivitas dari sebuah sistem atau proses bisnis atau menu yang ada pada perangkat lunak. Berikut adalah simbol-simbol yang ada pada *Activity Diagram* [7]:

No	Nama Komponen	Simbol	Deskripsi
1	Status awal		Status awal aktivitas sistem, sebuah diagram aktivitas memiliki sebuah status awal.
2	Aktivitas		Aktivitas yang dilakukan sistem, aktivitas biasanya diawali dengan kata kerja.
3	Percabangan/ <i>Decision</i>		Asosiasi percabangan dimana jika ada pilihan aktivitas lebih dari satu.
4	Penggabungan/ <i>Join</i>		Asosiasi penggabungan dimana lebih dari satu aktivitas digabungkan menjadi satu.
5	Status akhir		Status akhir yang dilakukan sistem, sebuah diagram aktivitas memiliki sebuah status akhir.
6	<i>Swimlane</i>		Memisahkan organisasi bisnis yang bertanggung jawab terhadap aktivitas yang terjadi.

Gambar 2. 7 Simbol Activity Diagram

### 2.4.1.3 Sequence Diagram




*Sequence diagram* merupakan diagram yang digunakan untuk memperlihatkan interaksi antar objek dalam perintah berurutan. *Sequence diagram* bertujuan untuk mendefinisikan urutan kejadian yang dapat menghasilkan *output* yang diinginkan. *Sequence diagram* merupakan diagram yang paling sesuai untuk mengembangkan model deskripsi use case menjadi spesifikasi desain. Berikut adalah simbol-simbol yang ada pada *Sequence Diagram* [6]:

NO	GAMBAR	NAMA	KETERANGAN
1		<i>LifeLine</i>	Objek entity, antarmuka yang saling berinteraksi.
		<i>Actor</i>	Digunakan untuk menggambarkan user / pengguna.
2		<i>Message</i>	Spesifikasi dari komunikasi antar objek yang memuat informasi-informasi tentang aktifitas yang terjadi.
3		<i>Boundary</i>	Digunakan untuk menggambarkan sebuah form.
4		<i>Control Class</i>	Digunakan untuk menghubungkan boundary dengan tabel.
5		<i>Entity Clas</i>	Digunakan untuk menggambarkan hubungan kegiatan yang akan dilakukan.

Gambar 2. 8 Simbol Sequence Diagram

#### 2.4.1.4 Class Diagram

*Class diagram* merupakan sebuah spesifikasi yang jika diinstansiasi akan menghasilkan sebuah objek dan merupakan inti dari pengembangan dan desain berorientasi objek. *Class diagram* menunjukkan hubungan antar class dalam sistem yang sedang dibangun dan bagaimana mereka saling berkolaborasi untuk mencapai suatu tujuan. Berikut simbol-simbol menggambar *Class Diagram* [6]:

NO	GAMBAR	NAMA	KETERANGAN
1		Generalization	Hubungan dimana objek anak ( <i>descendent</i> ) berbagi perilaku dan struktur data dari objek yang ada di atasnya objek induk ( <i>ancestor</i> ).
2		N-ary 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 actor.
5		Realization	Operasi yang benar-benar dilakukan oleh suatu objek.
6		Dependency	Hubungan dimana perubahan yang terjadi pada suatu elemen mandiri ( <i>independent</i> ) akan memengaruhi elemen yang bergantung padanya elemen yang tidak mandiri.
7		Association	Apa yang menghubungkan antara objek satu dengan objek lainnya.

Gambar 2. 9 Simbol *Class Diagram*

## 2.5 Konsep Basis Data

Basis data adalah sekumpulan data yang terintegrasi yang diorganisasi untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi. Terintegrasi maksudnya adalah setiap data akan memiliki hubungan dengan data yang lainnya. Dalam pengembangan sistem informasi termasuk sistem penunjang keputusan diperlukan basis data sebagai media penyimpanan data [9]. Basis data dapat menghasilkan informasi yang cepat dan tepat sehingga penggunaan basis data dapat membantu pengambil keputusan untuk segera memutuskan suatu masalah berdasarkan informasi yang ada.

Tujuan dalam merancang basis data adalah [9]:

### 1. Kecepatan dan kemudahan (*Speed*)

Pemanfaatan basis data memungkinkan untuk dapat menyimpan data atau melakukan perubahan/manipulasi terhadap data atau menampilkan kembali data tersebut dengan cepat dan mudah.

### 2. Efisiensi ruang penyimpanan (*Space*)

Penggunaan ruang penyimpanan di dalam basis data dilakukan untuk mengurangi jumlah *redudancy* (pengulangan) data, baik dengan melakukan penerapan sejumlah pengodean atau dengan membuat relasi-relasi (dalam bentuk *file*) antar kelompok data yang saling berhubungan.

### 3. Keakuratan (*Accuracy*)

Pemanfaatan pengodean atau pembentukan relasi antar data bersama dengan penerapan aturan/batasan tipe data, domain data, keunikan data, dan sebagainya, yang diterapkan dalam basis data, sangat berguna untuk menentukan ketidakakuratan pemasukan atau penyimpanan data.

### 4. Ketersediaan (*Availability*)

Dengan basis data dapat mem-*backup* data, memilah-milah data mana yang masih diperlukan dan data mana yang perlu disimpan ke tempat lain. Hal ini

mengingat pertumbuhan transaksi suatu organisasi dari waktu ke waktu membutuhkan media penyimpanan yang semakin besar.

#### 5. Kelengkapan (*Completeness*)

Lengkap atau tidaknya data yang dikelola bersifat relatif, baik terhadap kebutuhan pemakai maupun terhadap waktu. Dalam sebuah basis data, struktur dari basis data harus disimpan. Untuk mengakomodasi kebutuhan kelengkapan data yang semakin berkembang, tidak cukup hanya menambah *record- record* data, tetapi juga harus melakukan penambahan struktur dalam basis data.

#### 6. Keamanan (*Security*)

Sistem keamanan digunakan untuk dapat menentukan siapa saja yang boleh menggunakan basis data dan menentukan jenis operasi apa saja yang boleh dilakukan.

#### 7. Tampilan pengguna (*User view*)

Pemakai basis data sering kali tidak terbatas hanya pada satu pemakaian saja atau oleh satu sistem aplikasi saja. Basis data yang dikelola oleh sistem yang mendukung lingkungan multi *user*, akan dapat memenuhi kebutuhan ini, tetapi dengan menjaga/menghindari terhadap munculnya persoalan baru seperti inkonsistensi data (karena data yang sama diubah oleh banyak pemakai pada saat bersamaan).

### 2.6 Software Pendukung

Dalam pengembangan sistem pendukung keputusan guru berprestasi, software pendukung yang digunakan diantaranya, yaitu *Microsoft visio*, *MySQL*, *Microsoft Visual Studio*, dan *Crystal Report*.

#### 2.6.1 Microsoft Visio

*Microsoft Visio* merupakan *software* komputer yang biasanya digunakan untuk membuat diagram alir, *brainstorm*, dan skema jaringan. *Microsoft visio* menggunakan grafik vektor untuk membuat diagramnya, sehingga dapat membantu pengguna dalam meningkatkan kinerja, mulai dari mempersiapkan penggambaran diagram seperti

DFD, ERD, UML, Jaringan, rancangan *user interface* dan sejenisnya. Terlebih adanya sejumlah template yang disediakan, Dapat memungkinkan pengguna untuk membuat diagram dengan mudah, serta professional [13].

### 2.6.2 MySQL

*MySQL (My Structure Query Language)* adalah sebuah perangkat lunak sistem manajemen basis data SQL (*Database Management System*) atau DBMS yang *multithread, multi-user*. *MySQL* adalah DBMS yang *open source* dengan dua bentuk lisensi, yaitu *Free Software* (perangkat lunak bebas) dan *Shareware* (perangkat lunak berpemilik yang penggunaannya terbatas). *MySQL* merupakan *database engine* atau *server database* yang mendukung bahasa *database SQL* sebagai bahasa interaktif dalam mengelola data [14].

### 2.6.3 Microsoft Visual Studio

*Visual studio* adalah sebuah lingkungan pengembangan terpadu IDE (*Integrated Development Environment*) dari *Microsoft*. Hal ini digunakan untuk mengembangkan program komputer untuk sistem operasi *Microsoft Windows* super famili, serta situs web, aplikasi web dan layanan web. *Visual Studio* menggunakan *Microsoft platfoem* pengembangan perangkat lunak seperti *API Windows, Windows Forms Windows Presentation Foundation, Windows Stor* dan *Microsoft Silverlight*. *Visual Studio* dapat menghasilkan baik kode asli dan kode yang dikelola [15].

### 2.6.4 Crystal Report

*Crystal Report* merupakan aplikasi untuk membuat laporan baik berupa sumber data dari *database* buatan *Microsoft* maupun *database* buatan lainnya. Menggunakan *Crystal Report* memungkinkan *user* menghasilkan laporan yang mencakup visualisasi dengan penerapan bisnis baru ke dalam laporan tergantung kebutuhan perusahaan maupun keinginan *user* [15].

## 2.7 Pengujian Sistem

Pengujian sistem merupakan proses mengeksekusi sistem perangkat lunak untuk menentukan apakah sistem tersebut cocok dengan spesifikasi sistem dan berjalan



sesuai dengan lingkungan yang diinginkan. Pengujian sistem sering diasosiasikan dengan pencarian *bug*, ketidaksempurnaan program, kesalahan pada baris program yang menyebabkan kegagalan pada eksekusi sistem perangkat lunak. Kesalahan sistem perangkat lunak sangat bervariasi, mulai dari *output* yang salah, sistem yang *crash*, sampai pada sistem yang menggunakan memori terlalu banyak atau sistem yang berjalan terlalu lambat. *Bug-bug* seperti itu terjadi karena kode program yang dieksekusi belum diuji, urutan program yang dieksekusi belum diuji, urutan program yang dieksekusi belum diuji, kombinasi dari nilai *input* yang belum diuji, atau lingkungan pemrograman belum diuji [16].

### 2.7.1 Pengujian Black Box

*Black Box testing* adalah cara pengujian hanya dilakukan dengan menjalankan atau mengeksekusi unit atau modul, kemudian diamati apakah hasil dari mengeksekusi itu sesuai dengan proses bisnis yang diinginkan. *Black box testing* berfokus untuk menemukan kesalahan dari fungsi-fungsi yang salah atau hilang, kesalahan *interface*, kesalahan dalam struktur data atau akses database eksternal, kesalahan performa dan kesalahan inisialisasi dan terminasi [16].

### 2.8 Penelitian Terdahulu

Penelitian terdahulu digunakan sebagai acuan dan perbandingan untuk mengkaji penelitian dari penelitian-penelitian yang telah dilakukan sebelumnya. Adapun hasil penelitian-penelitian sebelumnya dapat ditunjukkan pada tabel sebagai berikut:

Tabel 2. 3 Penelitian Terdahulu

Penulis/Tahun	Judul Penelitian	Tujuan Penelitian	Metode	Kesimpulan Penelitian
Nugroho Bagus Wibowo, Dian Anubhaki (2020)	Sistem Informasi Penunjang Keputusan Penentuan Guru	Menghasilkan suatu sistem penunjang keputusan untuk	Metode <i>Analytic hierarchy Process</i> (AHP)	Dengan adanya sistem penunjang keputusan ini, dapat membantu pihak Tata Usaha dan Kepala

	Terbaik Pada Sekolah Smp Islam Al Hikmah dengan Metode Analytical Hierarchy Process (AHP)	menentukan guru terbaik dengan mengunakan metode (AHP).		Sekolah pada SMP Islam Al-Hikmah dalam menentukan guru terbaik dengan metode laporan cetak hasil perankingan
Eko Desta Budi Santoso, Nasrul Rofiah Hidayati, Fatim Nugrahanti (2018)	Rancang Bangun Sistem Pendukung Keputusan Penilaian Kinerja Karyawan dengan Menggunakan Metode AHP Berbasis Desktop Pada PDAM Kabupaten Madiun	Merancang dan membangun sistem pendukung keputusan penilaian kinerja karyawan dengan menggunakan metode AHP yang dapat menghasilkan perhitungan yang akurat dikarenakan dalam perhitungannya menggunakan metode AHP	Metode AHP ( <i>analytical hierarchy process</i> ).	Dengan adanya sistem penilaian kinerja karyawan dapat mempermudah pegawai dalam melakukan penilaian kinerja karyawan dan sistem yang dibangun dapat membantu dalam pengambilan keputusan pimpinan dengan menggunakan metode AHP sehingga perhitungan menjadi lebih akurat.

<p>Yodi Agung Dharmawan, Grace Gata (2019)</p>	<p>Sistem Pendukung Keputusan Penilaian Kinerja Karyawan dengan Metode Analytical Hierarchy Process (AHP) (Studi Kasus pada CV. Dwi Agung Mandiri)</p>	<p>Penelitian ini bertujuan memudahkan manager untuk proses penilaian kinerja karyawan, dapat memberikan hasil penilaian kinerja karyawan yang dapat menghasilkan perhitungan nilai bobot kriteria dan bobot alternatif dan perankingan untuk menentukan karyawan berkinerja terbaik, dapat memudahkan manager dalam proses pembuatan laporan.</p>	<p>Metode AHP (<i>analytical hierarchy process</i>).</p>	<p>Sistem penunjang keputusan penilaian kinerja karyawan dirancang untuk mempermudah proses penilaian kinerja dan Mempermudah proses pengolahan data serta akan mengurangi terjadinya kesalahan dalam perhitungan penilaian kinerja karyawan, sistem ini dirancang untuk memudahkan bagi manager dalam proses pembuatan laporan, dengan diterapkannya sistem ini akan</p>
------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------	---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

				<p>menghasilkan nilai</p> <p>bobot kriteria dan bobot alternatif, dibuatnya fitur</p> <p>menu perangkingan manager dapat mengetahui hasil nilai ranking kinerja karyawan yang terbaik.</p>
Ahmad Kamal (2017)	Sistem Penunjang Keputusan Penentuan Penerima Beasiswa Dengan Metode Multiple AHP	Untuk mengembangkan suatu sistem penunjang keputusan yang terkomputerisasi guna memberikan informasi secara kuantitatif sebagai bahan pertimbangan dalam pengambilan keputusan mengenai penentuan	Metode AHP ( <i>analytical hierarchy process</i> ).	Sistem hanya menghitung nilai AHP dan menampilkan penerima beasiswa yang berhak mendapatkan beasiswa berdasarkan nilai tertinggi dimana jumlah penerima beasiswa sesuai dengan kebijakan yang diinput pada sistem.

		penerima beasiswa.		
Renny Oktapiani, Ramlan Subakti, M. Azhar Lihan Sandy, Domeniq Gladys Tsafara Kartika, Davi Firdaus (2020)	Penerapan Metode AHP ( <i>Analytical Hierarchy Process</i> ) untuk pemilihan jurusan di SMK Doa Bangsa Palabuhanratu	Penerapan sistem pendukung keputusan menggunakan metode Analytic Hierarchy Process (AHP) untuk mempermudah dalam pemilihan jurusan di SMK Doa Bangsa.	Metode AHP ( <i>Analytical Hierarchy Process</i> ).	Dengan adanya penerapan metode AHP dalam menentukan pemilihan jurusan ini diharapkan dapat membantu calon siswa dan sekolah dalam menentukan pemilihan jurusan dan untuk penelitian selanjutnya dapat dibuat aplikasi yang dapat digunakan sebagai antarmuka pemilihan jurusan di SMK Doa Bangsa Palabuhanratu
Amirul Khoiri (2016)	Sistem Penunjang Keputusan Siswa	Untuk menerapkan metode AHP dalam	Metode AHP ( <i>Analytical</i>	Metode AHP sudah dapat diterapkan untuk melakukan

	<p>Berprestasi Di Smp Ma`Arif 10 Bangunrejo Lampung Tengah Menggunakan Metode AHP</p>	<p>penyeleksian siswa-siswi berprestasi di SMP Ma`arif 8 Bangunrejo Lampung Tengah sehingga dapat mempermudah dewan guru menentukan siapa yang menjadi siswa-siswi berprestasi.</p>	<p><i>Hierarchy Process</i>).</p>	<p>penyeleksian siswa berprestasi, dan dapat dibuat suatu rancangan sistem pengambilan keputusan seleksi siswa berprestasi menggunakan metode AHP.</p>
--	---------------------------------------------------------------------------------------	-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	-----------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------

# UNIVERSITAS MIKROSKIL