

BAB II

LANDASAN TEORI

2.1 Pemilihan Mobil Bekas

Mobil bekas merupakan kendaraan roda empat atau lebih yang pernah dimiliki orang lain, baik secara pribadi atau atas nama perusahaan atau instansi. Mobil bekas yang dijual berasal dari berbagai merk dan tahun pembuatan serta harga yang lebih murah dari mobil baru. Penjualan mobil bekas biasanya dilakukan secara langsung antara pembeli dan penjual melalui showroom mobil bekas. Jenis mobil bekas yang dijual seperti jenis mobil *multipurpose vehicle* (MPV), *sport utility vehicle* (SUV), sedan.

Dalam pemilihan mobil bekas, pengguna melihat mobil yang tersedia dan spesifikasi mobil. Informasi inilah yang digunakan untuk mengevaluasi semua alternatif yang ada dalam menentukan sistem pendukung keputusan pemilihan mobil bekas. Setidaknya ada dua kriteria yang diinginkan dalam pemilihan produk yaitu manfaat yang diperoleh dan kepuasan yang diharapkan.

2.2 Sistem Pendukung Keputusan (SPK)

Sistem Pendukung Keputusan atau *Decision Support System* (DSS) merupakan sistem informasi berbasis komputer yang menghasilkan berbagai alternatif keputusan untuk membantu dalam berbagai permasalahan yang terstruktur ataupun tidak terstruktur dengan menggunakan data atau model [2]. Sistem pendukung keputusan hanya sarana dalam membantu menghasilkan keputusan yang sudah diperhitungkan dari berbagai data yang sudah ada, yang mana keputusan akhir tetap ditangan pengguna sistem. Dapat disimpulkan sistem pendukung keputusan bukan merupakan alat pengambil keputusan melainkan sistem untuk membantu dalam mengambil keputusan yang didasari informasi data.

Sistem pendukung keputusan bertujuan untuk menyediakan informasi, membimbing, memberikan prediksi serta mengarahkan kepada pengguna informasi

agar dapat melakukan pengambilan keputusan dengan lebih baik [3]. Sistem Pendukung Keputusan merupakan implementasi dalam pengambilan keputusan yang telah diperkenalkan oleh ilmu – ilmu seperti *Operation Research* dan *Management Science*.

Sistem Pendukung Pengambilan Keputusan menurut Man dan Watson [4] adalah suatu sistem interaktif, yang membantu pengambil keputusan melalui penggunaan data dan model-model keputusan untuk memecahkan masalah-masalah yang sifatnya semi struktur dan tidak terstruktur”.

Karakteristik sistem pendukung keputusan adalah:

1. SPK dirancang untuk membantu pengambil keputusan dalam memecahkan masalah yang sifatnya semiterstruktur dan tidak terstruktur.
2. Dalam pengolahannya SPK mengkombinasikan penggunaan model-model/teknik analisis dengan teknik pemasukan data konvensional serta fungsi-fungsi pencari/integrasi informasi.
3. SPK dirancang sedemikian rupa, sehingga dapat digunakan dengan mudah oleh orang-orang yang tidak memiliki dasar kemampuan pengoperasian computer tinggi.
4. SPK dirancang dengan menekankan pada aspek fleksibilitas serta kemampuan adaptasi yang tinggi. Sehingga mudah disesuaikan dengan berbagai perubahan lingkungan yang terjadi dan kebutuhan pemakai. [4]

2.3 Metode *Analytical Hierarchy Process* (AHP)

Analytical Hierarchy Process (AHP) merupakan model pendukung keputusan yang dikembangkan oleh Thomas L. Saaty. Dengan hirarki suatu masalah yang kompleks dan tidak terstruktur dipecahkan ke dalam beberapa kelompok kriteria, kemudian disusun dalam bentuk hirarki. Model pendukung keputusan ini akan menguraikan masalah multi faktor atau multi kriteria yang kompleks [5]. Metode ini digunakan untuk mendefinisikan permasalahan yang dihadapi kemudian disusun dalam bentuk hirarki yang mana lebih terstruktur dan sistematis. Hirarki fungsional merupakan kebutuhan metode AHP yang inputnya

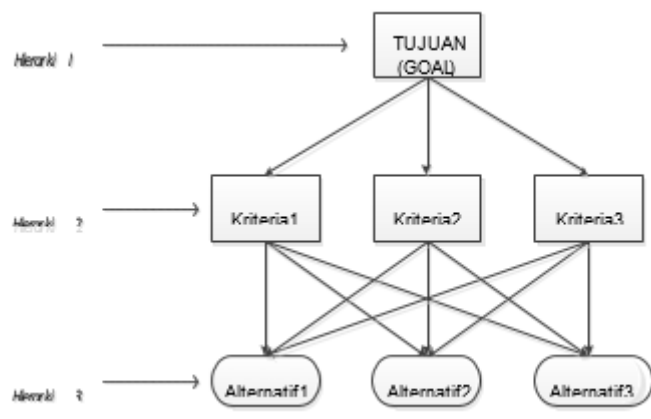
dari persepsi manusia. AHP sering digunakan sebagai metode pemecahan masalah dibanding dengan metode lainnya karena alasan sebagai berikut :

1. Struktur yang berhirarki, sebagai konsekuensi dari kriteria yang dipilih, sampai pada subkriteria yang paling kompleks.
2. Memperhitungkan validasi sampai batas toleransi inkonsistensi berbagai kriteria dan alternatif yang dipilih oleh keputusan.
3. Memperhitungkan daya tahan output analisis sensitivitas pengambilan keputusan. [6].

Dalam menyelesaikan permasalahan metode AHP ada beberapa prinsip yang harus dipahami diantaranya adalah sebagai berikut:

1. Membuat Hierarki

Hirarki adalah gambaran dari permasalahan yang kompleks dalam struktur dengan tingkatan dimana tingkat paling atas adalah tujuan dan diikuti tingkat kriteria, subkriteria dan seterusnya sampai pada tingkat tingkat paling bawah yaitu tingkat alternatif, Saaty (1994) [7]. Struktur hirarki AHP ditunjukkan seperti pada Gambar II.1



Gambar II.1 Struktur Hirarki Metode AHP

Hirarki utama adalah tujuan yang akan dicapai atau penyelesaian persoalan yang akan dikaji. Hirarki kedua adalah kriteria, kriteria yang harus dipenuhi oleh semua alternatif agar menjadi pilihan yang paling ideal. Hirarki ketiga adalah alternatif persoalan atau penyelesaian masalah.

2. Penilaian kriteria dan alternatif

Kriteria dan alternatif dilakukan dengan perbandingan berpasangan [8], untuk berbagai persoalan, skala 1 sampai 9 adalah skala terbaik untuk mengekspresikan pendapat. Nilai dan definisi pendapat dapat diukur menggunakan tabel analisis seperti tabel II.1.

Table II-1 Skala Perbandingan Kepentingan Kriteria

Intensitas Kepentingan	Definisi
1	Kedua elemen sama pentingnya
3	Elemen yang satu sedikit lebih penting daripada elemen yang lainnya
5	Elemen yang satu lebih penting daripada yang lainnya
7	Satu elemen jelas lebih mutlak penting daripada elemen lainnya
9	Satu elemen mutlak penting daripada elemen lainnya
2,4,6,8	Nilai-nilai antara dua nilai pertimbangan-pertimbangan yang berdekatan
Kebalikan	Jika aktivitas i mendapat satu angka dibandingkan dengan aktivitas j, maka j memiliki nilai kebalikannya dibandingkan dengan i

3. Penentuan Prioritas

Setiap kriteria dan alternatif dilakukan perbandingan berpasangan (*pairwise comparisons*). Nilai-nilai perbandingan disesuaikan dengan judgement yang telah ditentukan untuk menghasilkan nilai bobot. Bobot dan prioritas dihitung dengan memanipulasi matriks berpasangan.

4. Konsistensi Logis

Konsistensi memiliki dua makna, yang pertama Objek-objek yang serupa bisa dikelompokkan sesuai dengan keseragaman dan relevansi. Kedua menyangkut tingkat hubungan antar objek yang

didasarkan pada kriteria tertentu. Perhitungan konsistensi logis dilakukan dengan mengikuti langkah-langkah sebagai berikut:

- a. Mengkalikan matriks dengan prioritas

$$(A)(W^T) \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots \\ a_{n1} & \dots & \dots & a_{nn} \end{bmatrix} \begin{bmatrix} W_1 \\ W_2 \\ \dots \\ W_n \end{bmatrix}$$

- b. Menjumlahkan hasil perkalian per baris.
c. Hasil penjumlahan tiap baris dibagi prioritas bersangkutan dan hasilnya dibagi dengan jumlah elemen yang didapatkan

$$t = \frac{1}{n} \sum_{i=1}^n \left(\frac{\text{elemen ke-}i \text{ pada } (A)(W^T)}{\text{elemen ke-}i \text{ pada } W^T} \right)$$

- d. Indeks Konsistensi (CI)

$$CI = \frac{(\lambda \text{ Maks} - n)}{n}$$

- e. Rasio Konsistensi

Table II-2 Indeks Ratio Konsistensi

N	1	2	3	4	5	6	7	8	9
R	0.0	0.0	0.5	0.9	1.1	1.2	1.3	1.4	1.4
1	0	0	8	0	2	4	2	1	5

Menghitung Consistency Ratio (CR) dengan rumus:

$$CR = \frac{CI}{IR}$$

Dimana CR = Consistency Ratio

CI = Consistency Index

IR = Indeks Random Consistency.

Jika nilainya lebih dari 10%, maka penilaian dari data judgment harus diperbaiki. Namun jika rasio konsistensi (CI/IR) kurang atau sama dengan 0,1, maka hasil perhitungan bisa dinyatakan benar. [9]

2.4 Smartphone

Smartphone adalah telepon selular pintar yang meliputi *mikroprosesor*, *memori*, layar dan modem. Telepon genggam ini mempunyai kemampuan *mobile computing*, dengan *mobile computing smartphone* memiliki kemampuan yang tak bisa dibandingkan dengan ponsel biasa. *Smartphone* mempunyai fungsi sebagai *media portable* yang menggabungkan fungsionalitas PC dengan akses data WiFi dan internet *broadband*, di mana terdapat pesan teks, kamera, pemutar musik, video, game, akses email, tv digital, *search engine*, pengelola informasi pribadi, *fitur GPS*.

2.5 Android

Android adalah sebuah sistem operasi pada handphone yang bersifat terbuka dan berbasis pada sistem operasi *Linux*. Android menyediakan platform terbuka bagi para pengembang untuk menciptakan aplikasi mereka sendiri yang akan digunakan untuk bermacam peranti bergerak [10]. Awalnya, *Google Inc* membeli Android Inc, pendatang baru yang membuat peranti lunak untuk ponsel. Kemudian untuk mengembangkan Android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *TMobile*, dan *Nvidia*. Pada saat perilisan perdana Android, 5 November 2007.

Sistem operasi Android dibagi menjadi lima bagian di dalam empat lapisan utama, yaitu:

1. *Linux Kernel*

Merupakan *kernel* yang menjadi dasar Android, di mana terdapat seluruh perangkat *driver* tingkat rendah untuk berbagai komponen perangkat keras pada perangkat Android.

2. *Libraries*

Bagian ini berisi seluruh kode yang menyediakan fitur utama pada sistem operasi Android. Contohnya yaitu *SQLite library* menyediakan dukungan database sehingga sebuah aplikasi dapat menggunakannya sebagai penyimpanan data. *WebKit library* menyediakan fungsi untuk *web browsing*.

3. Android Runtime

Android *Runtime* menyediakan sekumpulan *core libraries* yang memungkinkan developer untuk membuat aplikasi android menggunakan bahasa pemrograman *Java*. Android *Runtime* juga mencakup *Dalvik virtual machine* yang memungkinkan setiap aplikasi android untuk berjalan dalam prosesnya sendiri. Dalvik merupakan mesin *virtual* yang dirancang khusus untuk Android dan dioptimalkan untuk *battery-powered* perangkat mobile dengan memori yang terbatas dan CPU.

4. Application Framework

Bagian ini memperlihatkan berbagai kemampuan sistem operasi android kepada pengembang aplikasi sehingga dapat menggunakannya di dalam aplikasi yang dikembangkan.

5. Application

Pada lapisan teratas ini akan ditemukan aplikasi yang disertakan bersama perangkat Android, seperti telepon, kontak, *browser*, dan lainlain, serta aplikasi yang di-*download* dan *install* dari *Android Market*.

2.6 Java

Bahasa *java* merupakan bahasa tingkat tinggi yang berorientasi *objek* atau OOP dan program java juga terdiri dari bagian-bagian yang disebut dengan *Class*. Kumpulan kelas di pustaka *Java* disebut dengan *Java Application Programming Interface* (API). Teknologi *Java* memiliki tiga komponen penting, yaitu: *Programming-language specification*, *Application-programming interface*, *Virtual-machine specification Java Development Kit* (JDK), [11]. Java juga dirancang agar dapat dijalankan di semua platform dengan kata lain bisa dijalankan di banyak *Operating System* (OS) seperti *Linux*, *Windows*, *Machintos*. Java juga dirancang untuk menghasilkan aplikasi – aplikasi dengan performansi yang terbaik, seperti aplikasi database *Oracle 8i/9i* yang core-nya dibangun menggunakan bahasa pemrograman *Java*. Sedangkan *Java* bersifat *neutral architecture*, karena *Java Compiler* yang digunakan untuk mengkompilasi kode program *Java* dirancang untuk menghasilkan kode yang netral terhadap semua arsitektur perangkat keras yang disebut sebagai *Java Bytecode*.

2.7 Android Studio

Android Studio merupakan sebuah *Integrated Development Environment* (IDE) khusus untuk membangun aplikasi yang berjalan pada platform android. Android studio ini berbasis pada *IntelliJ IDEA*, sebuah IDE untuk bahasa pemrograman Java. Bahasa pemrograman yang didukung di Android Studio adalah *Java*, *C++*, *Go*, dan *Kotlin*, sedangkan untuk membuat tampilan atau *layout*, digunakan bahasa *XML*. Android studio juga terintegrasi dengan *Android Software Development Kit* (SDK) untuk *deploy* ke perangkat android. Android Studio juga merupakan pengembangan dari *eclipse*, dikembangkan menjadi lebih kompleks dan professional yang telah tersedia didalamnya Android Studio IDE, Android SDK tools.

2.8 Basis Data

Data adalah Representasi fakta dunia nyata yang mewakili suatu objek seperti manusia, barang, hewan, peristiwa, konsep, keadaan, dan sebagainya yang direkam dalam bentuk angka, huruf, *symboleks*, gambar, bunyi atau kombinasinya. Basis Data adalah Sekumpulan data yang terintegrasi yang diorganisasikan untuk memenuhi kebutuhan para pemakai di dalam suatu organisasi. Satu basis data menunjukkan kumpulan data yang dipakai dalam satu lingkup informasi. Dalam satu file terdapat *record-record* yang sejenis, sama besar, sama bentuk, merupakan satu kumpulan entity yang seragam. Satu record terdiri dari *field-field* yang saling berhubungan untuk menunjukkan bahwa *field* tersebut dalam satu pengertian yang lengkap dan direkam dalam satu record. Suatu sistem manajemen basis data berisi satu koleksi data yang saling berelasi dan satu set program untuk mengakses data tersebut. Jadi sistem manajemen basis data dan set program pengelola untuk menambah data, menghapus data, mengambil data dan membaca data Jauh sebelum ditemukan dan diaplikasikan basis data dalam pengembangan sistem dan aplikasi, industri teknologi informasi menggunakan pendekatan aplikasi berbasis file atau *file-base approach*.

2.9 *Firebase*

Firebase adalah BaaS (*Backend as a Service*) penyedia layanan *realtime database* dan *backend*. Layanan yang dimiliki *firebase* mempermudah pekerjaan *Mobile Apps Developer* yang dapat menyimpan dan melakukan sinkronisasi data ke banyak *user* dan sebagai penyedia layanan pengembangan aplikasi. *Firebase* memiliki banyak library yang memungkinkan untuk pengembang aplikasi untuk *Android*, *Ios*, *Javascript*, *Java*, *Objective-C* dan *Node JS*. Fitur yang dimiliki *Firebase* adalah sebagai berikut:

1. *Firebase Analytics*
2. *Firebase Cloud Messaging dan Notification*
3. *Firebase Authentication*
4. *Firebase Remoter Config*
5. *Firebase Real Time Database*
6. *Firebase Crash Reporting*


2.10 UML






UML (*Unified Modeling Language*) merupakan bahasa visual yang dipergunakan untuk mendokumentasikan dan pemodelan mengenai sebuah sistem dengan menggunakan diagram dan teks-teks pendukung. Alat bantu yang digunakan dalam menggambarkan pemodelan sebuah sistem yang dirancang meliputi:

7. *Diagram Use Case*

Diagram Use Case menggambarkan apa saja aktifitas yang dilakukan oleh suatu sistem dari sudut pandang pengamatan luar. Diagram ini digunakan untuk mengetahui kejadian seseorang berinteraksi apa saja yang ada dalam sistem.

Table II-3 *Diagram Use Case*




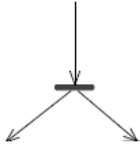

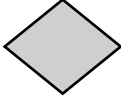

Gambar	Keterangan
	<i>Use Case</i> menggambarkan fungsionalitas yang disediakan sistem sebagai unit-unit yang bertukar

Gambar	Keterangan
	pesan antar unit dengan aktor, yang dinyatakan dengan menggunakan kata kerja
	<i>Actor</i> atau Aktor adalah <i>Abstraction</i> dari orang atau sistem yang lain yang mengaktifkan fungsi dari target sistem. Untuk mengidentifikasi aktor, harus ditentukan pembagian tenaga kerja dan tugas-tugas yang berkaitan dengan peran pada konteks target sistem. Orang atau sistem bisa muncul dalam beberapa peran. Perlu dicatat bahwa aktor berinteraksi dengan <i>Use Case</i> , tetapi tidak memiliki kontrol terhadap <i>use case</i>
	Asosiasi antara aktor dan <i>use case</i> , digambarkan dengan garis tanpa panah yang mengindikasikan siapa atau apa yang meminta interaksi secara langsung dan bukannya mengindikasikan data.
	Asosiasi antara aktor dan <i>use case</i> yang menggunakan panah terbuka untuk mengindikasikan bila aktor berinteraksi secara pasif dengan sistem
	<i>Include</i> , merupakan di dalam <i>use case</i> lain (<i>required</i>) atau pemanggilan <i>use case</i> oleh <i>use case</i> lain, contohnya adalah pemanggilan sebuah fungsi program
	<i>Extend</i> , merupakan perluasan dari <i>use case</i> lain jika kondisi atau syarat terpenuhi

1. *Diagram Activity*

Activity diagram merupakan state diagram khusus, di mana sebagian besar state adalah sebuah aliran kerja sistem sebelumnya (*internal processing*). Diagram ini lebih menggambarkan proses-proses dan jalur-jalur aktivitas dari level atas secara umum.


Table II-4 Activity diagram




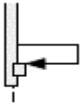


Gambar	Keterangan
	<i>Start Point</i> , diletakkan pada pojok kiri atas dan merupakan awal aktivitas
	<i>End Point</i> , akhir aktivitas
	<i>Activities</i> , menggambar kan suatu proses/kegiatan bisnis
	<i>Fork/percabangan</i> , digunakan untuk menunjukkan kegiatan yang dilakukan secara paralel atau untuk menggabungkan dua kegiatan paralel menjadi satu
	<i>Join (penggabungan) atau rake</i> , digunakan untuk menunjukkan adanya dekomposisi
	<i>Decision Points</i> , menggambar kan pilihan untuk pengambilan keputusan, <i>true</i> atau <i>false</i>
	<i>Swimlane</i> , pembagian <i>activity diagram</i> untuk menunjukkan siapa melakukan apa

3. Diagram Sequence

Diagram Sequence merupakan salah satu diagram Interaction yang menjelaskan bagaimana suatu operasi itu dilakukan; message (pesan) apa yang dikirim dan kapan pelaksanaannya. Simbol yang digunakan dalam *Diagram Sequence* seperti table II.5

Table II-5 Simbol Diagram Sequence

Gambar	Keterangan
	<i>Entity Class</i> , merupakan bagian dari sistem yang berisi kumpulan kelas berupa entitas-entitas yang membentuk

Gambar	Keterangan
	gambaran awal sistem dan menjadi landasan untuk menyusun basis data
	<i>Boundary Class</i> , berisi kumpulan kelas yang menjadi <i>interfaces</i> atau interaksi antara satu atau lebih aktor dengan sistem, seperti tampilan form entry dan form cetak
	<i>Control class</i> , suatu objek yang berisi logika aplikasi yang tidak memiliki tanggung jawab kepada entitas, contohnya adalah kalkulasi dan aturan bisnis yang melibatkan berbagai objek
	<i>Message</i> , simbol mengirim pesan antar <i>class</i>
	<i>Recursive</i> , menggambarkan pengiriman pesan yang dikirim untuk dirinya sendiri
	<i>Activation</i> mewakili sebuah eksekusi operasi dari objek, panjang kotak ini berbanding lurus dengan durasi aktivasi sebuah operasi
	<i>Lifeline</i> merupakan garis titik yang terhubung dengan objek, sepanjang <i>lifeline</i> terdapat <i>activation</i>

2. Class Diagram

Class Diagram menggambarkan keadaan secara luas dari tiap-tiap kelas suatu sistem dan menghubungkan kelas tersebut. Class diagram menggambarkan struktur dan deskripsi atribut sebuah kelas yang dikoneksikan.

Class Diagram secara khas meliputi, Kelas (*Class*), Relasi *Assosiations*, *Generalitation* dan *Aggregation*, atribut (*Attributes*), operasi (*operation/method*) dan *visibility*. Tingkat akses objek eksternal kepada suatu operasi atau attribute dan Hubungan antar kelas mempunyai keterangan yang disebut dengan *Multiplicity* atau *Cardinality*.

BAB III