



### **Hak cipta dan penggunaan kembali:**

Lisensi ini mengizinkan setiap orang untuk menggubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

### **Copyright and reuse:**

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

## BAB II

### LANDASAN TEORI

#### 2.1. Landasan *E-commerce*

Keuntungan dari *E-commerce*. *E-commerce*, di mata netizen memiliki keuntungan di mana *user* selaku *Customer* sudah mengakui *Marketplace* ini sebagai sebuah revolusi dibanding pengganti dari toko konvensional (Yeng Sin, 2016).

*Customer* yang sudah merasakan kemudahan dan kecepatan dari menggunakan internet akan berinisiatif memanfaatkan media tersebut untuk kebutuhan yang lebih jauh. Pergantian ini bermula dari *Website* komunikasi seperti media sosial. Populernya media sosial membuat *user* ingin lebih dari sekedar bertukar informasi dalam internet. Dari situlah banyak dari netizen mulai memanfaatkan internet sebagai alat berbisnis. *User* yang menyadari praktisnya berbelanja *Online* pun membuat keputusan untuk memanfaatkan media ini lebih jauh, baik menjadi pelanggan tetap *E-commerce*, atau membangun yang baru.

Saat ini, istilah *E-commerce* merujuk kepada *Website* yang menjual barang dan jasa. Istilah tersebut awalnya ditunjukkan kepada penggunaan *Website* untuk mengiklankan produk ketika pertama kali muncul di tahun 1994. Barulah pada tahun 2006, istilah *E-commerce* digunakan untuk mendefinisikan sebuah tempat di mana terjadi kegiatan jual beli produk secara *Online* (Yunita Dewi, 2014). Penamaan ini terjadi oleh karena peningkatan drastis dari jumlah toko *Online* yang didirikan sejak tahun tersebut. Indonesia sendiri mengalami peningkatan dari

jumlah toko *Online* pada tahun 2008. Jumlah tersebut terus meningkat seiring berjalannya waktu sampai sekarang.

Kata *E-commerce* itu sendiri diartikan sebagai pemasaran *Online*. Namun, pemasaran yang dimaksud tidak hanya sekedar produk, melainkan seluruh informasi *Website* termasuk konsepnya. Pemasaran *Online* tersebut meliputi layanan yang diberikan, cara pembayaran, peraturan berbelanja bagi *user*, dan juga promosi seperti testimonial maupun periklanan interaktif lewat email. Pemasaran ini bertujuan untuk meyakinkan *user* mengenai keamanan dari toko *Online* tersebut. Kemudian juga untuk merealisasikan *Website* sebagai media self-service di mata *user*.

Konsep dasar dari lancarnya sebuah bisnis *E-commerce* adalah sebuah sistem yang diciptakan oleh pihak ketiga untuk menciptakan kondisi yang menjamin keamanan dari kedua belah pihak dalam bertransaksi. Efektivitas dari *E-commerce* dapat didefinisikan sebagai persepsi dari *Customer* apabila ruang lingkup *website* ini memiliki sistem pertahanan yang melindungi transaksi dari ancaman-ancaman dunia maya (Fang 2014). Hal ini digambarkan dengan sesi pembayaran di mana sistem transaksi *Online* hanya akan mengijinkan transaksi setelah *Customer* setuju untuk membayar. Tindakan ini memberikan rasa aman kepada kedua belah pihak.

Namun hingga akhirnya, konsep bukanlah sesuatu yang bisa dicerna oleh semua orang. Ini dikarenakan, biarpun perkembangan *E-commerce* yang sangat pesat dalam beberapa tahun terakhir ini juga disertai dengan meningkatnya standar perlindungan yang diberikan. Namun tidak sedikit juga *Customer* dan netizen yang

masih memiliki pandangan yang sebaliknya terhadap keamanan yang ditawarkan oleh *website* ini.

## **2.2.Pembuatan dan Pengembangan *E-commerce***

Suksesnya keberadaan *e-commerce* tidak luput dari *Customer* yang sangat puas dengan kehadiran sistem informasi ini. Era digital telah mendorong banyak aktifitas untuk bisa dilakukan lewat di depan komputer (Leong, 2016). Perubahan ini telah terbukti lebih praktis di mata generasi milenial yang sudah menjadikan IT dan internet sebagai kebutuhan sehari-hari. Oleh karena itu, ketika *Customer* dihadapkan dengan *Website* yang berfungsi sebagai tempat jual beli. *E-commerce* memberikan keputusan atas banyaknya produk yang bisa dicari. Dan juga banyaknya bentuk mode web itu sendiri

Ada beberapa teori yang menyelidiki faktor penentu sebuah usaha, terutama yang menengah ke bawah (SMEs) untuk beralih menggunakan *E-commerce*. Salah satu dari Teori tersebut adalah *Technological, Organizational, and Environmental Framework* atau juga disebut *framework TOE* (Rahayu, 2015). Teori ini dikembangkan untuk menjelaskan pengaruh faktor penghubung dalam berinovasi lewat *e-commerce*. *Framework TOE* diakui sebagai model terdepan dalam mempelajari penggunaan *Online Marketplace*. Ini dikarenakan dalam prakteknya, *TOE* tidak hanya mempertimbangkan faktor teknologi saja, namun juga faktor eksternal seperti lingkungan. Metode tersebut akan menghasilkan deskripsi yang lebih rinci dan umum.

Faktor-faktor yang dipelajari oleh *framework* TOE:

a. Faktor Teknologi

Faktor yang menjadi konteks utama ini mengacu kepada beberapa efek yang terjadi ketika mengadopsi teknologi dalam bentuk *Online Marketplace*. Kelebihan atau keuntungan memakai teknologi adalah yang pertama. Lalu dilanjutkan dengan biaya dari pemakaian teknologi. Dan yang terakhir adalah kecocokan atau kompatibilitas dari organisasi ketika ingin menggunakan teknologi. Faktor ini akan mempengaruhi manajemen dari organisasi dalam mengalokasikan sumber daya dan tenaga untuk membangun *website* yang diinginkan.

b. Faktor organisasi

Faktor ini mencakup kesiapan dari organisasi atau perusahaan untuk menggunakan sistem informasi yang baru. Proses seperti perpindahan menuju penggunaan IT akan menjadi bagian dari kesiapan perusahaan. Selain itu, faktor ini juga mempelajari ukuran dari perusahaan atau organisasi yang akan mengadopsi teknologi baru. Besar kecilnya sebuah usaha akan menentukan seberapa kompleks dari *E-commerce* yang akan dikembangkan.

c. Faktor lingkungan

Faktor lingkungan atau *environment* mengacu kepada hubungan antara perusahaan, baik dari sisi pemilik maupun keseluruhan organisasi, dengan pihak luar. Faktor ini mempelajari tekanan yang diberikan pihak-pihak tersebut kepada perusahaan. Lalu menjelaskan bagaimana tekanan tersebut mendorong perusahaan untuk mengadopsi sistem informasi yakni *E-commerce*. Tekanan ini juga akan terus

berlanjut sebagai kritik untuk mengembangkan *Online* marketing menjadi lebih baik. Pihak-pihak tersebut terdiri dari tekanan dari *Customer*, kompetitor, mitra bisnis (perusahaan IT maupun *Software House*), dan juga dukungan dari pemerintah.

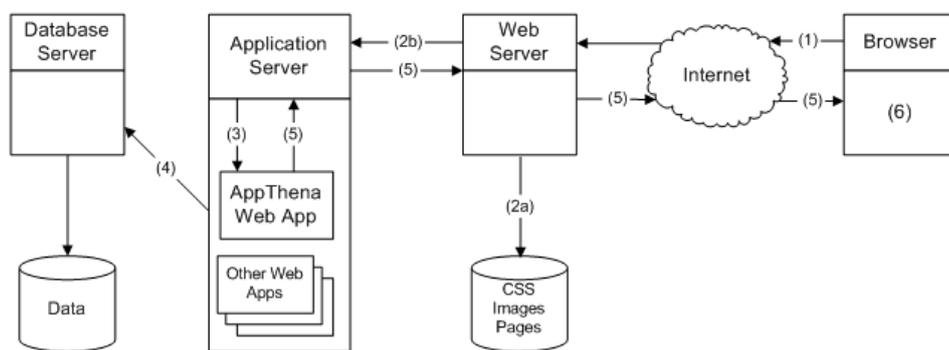
#### d. Faktor individu

Faktor yang terakhir mempelajari penggunaan *e-commerce* dari sudut pandang pemilik usaha atau owner. Aspek penting seperti inovasi dan pengambilan keputusan berada di tangan pemilik usaha. Keputusan dari sang pemilik dapat menentukan sukses tidaknya sebuah perusahaan ketika mengadopsi *e-commerce*. Keberhasilan tersebut dapat ditentukan dari kemampuan pemilik usaha dalam bidang IT. Kemampuan si pemilik akan terlihat dari cara ia memanfaatkan teknologi baru di perusahaannya. Selain itu, faktor individu juga mempelajari pengalaman si pemilik dalam dunia IT sebelum mengadopsi *e-commerce*.

### **2.3. Struktur Pemrograman Aplikasi Web**

*E-commerce* merupakan suatu bentuk dari *Website* yang bernama aplikasi web. Variasi *Website* ini dibedakan dengan adanya interaksi antar *user*, fungsi untuk menyimpan *file* data, dan juga adanya akses untuk konten tertentu di dalam *Website*. Sistem yang melibatkan hubungan antara *client-side*, *server-side*, dan *Database* *web* ini sudah diterapkan oleh banyak *Website* besar di internet. *Website* sendiri adalah aplikasi yang disimpan, diubah, dan dijalankan di platform web *server* (Raharjo, 2016). *Website* bekerja dengan merespon setiap permintaan yang dibuat

*user* melalui *web browser*. Lalu mengembalikan isi respon tersebut kepada *user*. Tujuan dari aplikasi ini adalah untuk membuat halaman *Website* yang tampil di *browser* bersifat dinamis. Selain itu, *website* juga dapat digunakan untuk mengakses *file* data baik yang dinamis maupun statis.



**Gambar 2.1. Contoh arsitektur model kerja aplikasi web (Andrews, 2013)**

Masing-masing sisi dari aplikasi web seperti pada gambar 2.1. dibangun di atas *framework* dengan bahasa pemrograman sebagai alat pengembangannya. Tampilan dari halaman *Website* didesain dengan menggunakan bahasa *html* dan *Java*, lalu dimodifikasi dengan *css*. Sedangkan untuk *server-side*, fungsi *website* dibangun dengan bahasa *php*.

### 2.3.1. Dasar *HTML*

*HTML* atau Hyper Text Markup Language adalah kode yang ditulis atau diketik di dalam *file* teks dengan aturan tertentu (Raharjo, 2016). *File* ini kemudian menjadi halaman bagi *user* apabila ditampilkan melalui aplikasi web *browser*. Satu dokumen atau *file html* dapat diartikan sebagai sebuah halaman web. Alasannya untuk memberi pengertian apabila semua halaman yang tampil di web

menggunakan kode *html*. Pembuatan dokumen *html* bersifat fleksibel oleh karena dapat dibuat lewat aplikasi *Text Editor* apapun. Contoh dari aplikasi tersebut adalah Notepad atau Sublime Text.

Selain menggunakan *Text Editor*, *html* juga dapat dibuat di aplikasi yang dikhususkan untuk perancangan web. Aplikasi atau *Software* perancangan web ini menyediakan fitur untuk pembuatan kode-kode *html*. Aplikasi ini sangat mempermudah pembuatan dokumen *html*. Contoh dari aplikasi perancangan web adalah seperti Adobe Dreamweaver atau Microsoft Visual Studio.

Kode *html* dari sebuah halaman *Website* dapat dicari dengan menggunakan aplikasi web *browser*. Proses ini dikenal di Google Chrome dengan nama Inspect Element. Sedangkan di Firefox, proses ini dikenal dengan nama View-Page Source. Tujuannya adalah untuk membantu seseorang mencari referensi dan mempelajari *Website* lain.

### **2.3.2. Dasar CSS**

CSS merupakan kode bahasa untuk mendesain web yang digunakan dengan mengatur tampilan sebuah halaman web. Fungsi utama dari bahasa ini adalah untuk merancang, mengubah dan membentuk halaman *Website* (Raharjo, 2016). CSS bekerja dengan membantu desain dari halaman yang sudah memiliki *HTML* atau *XHTML* di dalamnya. oleh karena itu, CSS tidak dapat digunakan apabila halaman yang ingin didesain tidak memiliki kode *HTML* terlebih dahulu.

CSS bertujuan untuk memisahkan isi dari halaman web dengan tampilan estetikanya. Hal ini akan membuat proses pemrograman lebih teratur. Alasannya

karena dokumen kerangka web yang berisikan konten sudah terpisah dari dokumen untuk kostumisasi halaman web. Pemisahan ini akan membantu proses pengulangan pada tampilan halaman web. Dengan demikian, perancang dapat membuat banyak halaman web yang berbeda dengan hanya membutuhkan 1 dokumen *CSS* untuk kostumisasi saja. Pengaturan ini akan menjadikan pengembangan web lebih simpel dan hemat waktu.

Konten yang dapat diatur oleh *CSS* antara lain adalah ukuran objek, bentuk, warna, penempatan dan format dari sebuah halaman. Lalu, *CSS* dapat mengatur konten tulisan dan paragraf. Kemudian, bahasa ini akan mengatur style yang digunakan pada objek.

### **2.3.3. Dasar *PHP***

*PHP* merupakan salah satu bahasa pemrograman skrip yang dirancang untuk membangun *website* (Raharjo, 2016). Kode yang ditulis menggunakan *PHP* ditujukan untuk diterjemahkan ke dalam dokumen *HTML*. Proses ini terjadi ketika *user* menjalankan fungsi untuk memanggil dari web *browser*. Kode yang dibuat dengan *PHP* akan di parsing di dalam web *browser* oleh interpreter milik *PHP* itu sendiri. Baru setelah itu, hasil terjemahan ditampilkan kepada *user* di halaman web *browser*. *PHP* dikatakan sebagai bahasa sisi *server* (*server-side*) oleh karena prosesnya yang terjadi di lingkungan web *server*. Kode *PHP* tidak bisa terlihat apabila *user* menggunakan perintah Inspect Element milik aplikasi web *browser* yang ia gunakan. Selain menggunakan *PHP*, terdapat alternatif pembuatan aplikasi

web seperti Java (JSP – JavaServer Pages dan Servlet), Perl, Python, Ruby, dan juga ASP.

#### **2.4. Metode Unified Modelling Language**

Unified Modelling Language merupakan sebuah metode yang menjadi bahasa utama dalam perancangan, visualisasi dan dokumentasi dari sistem informasi (Dharwiyanti, 2013). Metode ini menjadi standar untuk mengembangkan model sistem seperti *Software* dan aplikasi. Pengembangan UML berorientasi pada object dengan menggunakan class dan operation dalam konsep dasarnya. Oleh karena itu, metode ini cocok digunakan bersamaan dengan bahasa pemrograman yang berorientasi objek seperti C++, C#, Java, dan VB. UML sendiri terdiri dari alat bantu penampil visual dan teknik untuk mendefinisikan rancangan *website*. Hasil notasi UML berbentuk *Diagram-Diagram* dari *website* yang ingin dikembangkan. Kemudian *Diagram-Diagram* tersebut akan dihubungkan satu sama lain lewat syntax milik UML.

Penggunaan UML memiliki 2 tahap. Pertama adalah pembuatan *Diagram* di mana analyst harus dapat menguasai teknik perancangannya (Satzinger, 2012). Kedua, adalah analisis dari *Diagram* yang dibuat beserta dengan mempelajari pengembangannya sesuai dengan standar UML. Penguasaan dari 2 tahap ini memiliki konsep dasar, yaitu structural classification, dynamic behavior, dan model management. Berikut adalah *Diagram-Diagram* yang dimiliki UML:

### 2.4.1. Use case Diagram

*Diagram* yang menjadi desain terdasar ketika merancang menggunakan UML. *Use case* memvisualisasikan sisi fungsionalitas dari *website* yang mau dikembangkan. *Diagram* ini menampilkan beberapa interaksi antara aktor dengan *website* untuk menjalankan aktivitas.

Masing-masing interaksi merupakan pekerjaan tertentu yang diurutkan dari proses awal dari kinerja sistem sampai selesai. Interaksi ini juga harus memfokuskan pada konteks “apa” yang dikerjakan oleh sistem, bukan “bagaimana”.

*Use case* bertujuan untuk membuat deskripsi rincian dari requirement sebuah sistem. Tampilan dari *use case* akan membantu komunikasi dengan klien mengenai perancangan sistem beserta fiturnya.

Suatu rancangan sistem dapat memiliki beberapa *use case*. Setiap dari *Diagram* ini terhubung secara general di mana satu *use case* merupakan spesialisasi dari yang lainnya. Lalu, satu dari *use case* dapat merupakan perpanjangan atau pecahan dari *use case* yang lain. *Diagram* yang lebih kecil ini akan mengikutsertakan fungsionalitas dari *use case* lainnya.

Elemen di dalam *use case Diagram* dapat terlihat pada tabel di gambar 2.2. terdiri dari:

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

- Batas sistem

Batas sistem adalah kotak yang mewakili sistem di dalam *Diagram*. Kotak ini menggambarkan elemen apa saja yang internal dan eksternal terhadap *Diagram*. Nama dari sistem diletakan di atas kotak.

- Aktor

Aktor adalah peran yang dipakai oleh *user* ketika berinteraksi di dalam sistem. Aktor bersifat eksternal terhadap sistem. Aktor bekerja dengan memulai sebuah use case di dalam sistem. Elemen ini memiliki simbol figur manusia.

- Use case

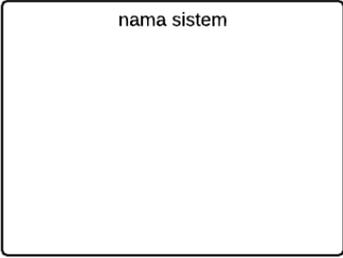
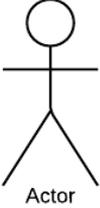
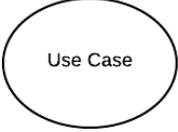
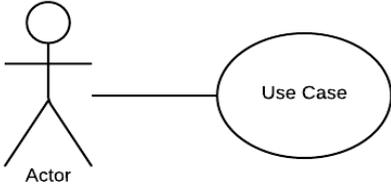
Use case itu sendiri adalah proses utama yang dijalankan oleh sistem. Proses ini menguntungkan aktor dengan cara tertentu. Use case dinamakan memakai kata kerja deskriptif. Elemen ini memiliki simbol berbentuk oval. Aktor adalah peran yang dipakai oleh *user* ketika berinteraksi di dalam sistem.

- Hubungan asosiasi

Elemen ini menghubungkan aktor dengan use case. Tujuannya untuk menunjukan use case mana saja yang berinteraksi dengan aktor. Hubungan digambarkan dengan sebuah garis. Elemen ini lalu menjelaskan komunikasi dua arah antara use case dengan aktor.

U  
N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

**Tabel 2. 1. Elemen dalam use case Diagram (Dennis, 2012).**

Istilah elemen	Simbol
<p>Batas Sistem:</p> <ul style="list-style-type: none"> <li>• Kotak yang mewakili ruang lingkup sistem.</li> <li>• Nama dari sistem diletakan di atas kotak.</li> </ul>	
<p>Aktor:</p> <ul style="list-style-type: none"> <li>• Seseorang atau sistem lain yang berinteraksi dan memperoleh nilai dari sistem.</li> </ul>	
<p>Use case:</p> <ul style="list-style-type: none"> <li>• Mewakili proses penting dari fungsi sistem.</li> </ul>	
<p>Hubungan asosiasi:</p> <ul style="list-style-type: none"> <li>• Garis yang mewakili interaksi aktor dengan use case.</li> </ul>	

### 2.4.2. *Class Diagram*

*Class Diagram* merupakan desain struktur dari atribut suatu sistem beserta dengan fungsi atau metode untuk memanipulasi atribut-atribut tersebut. *Diagram* ini merupakan dasar dari pengembangan dan desain berorientasi objek. Struktur yang ditampilkan terdiri dari hubungan class, package dan objek. Masing-masing dari class ini memiliki deskripsi yang menjadi poin penghubung, baik lewat pemanggilan, pewarisan atau asosiasi. Area dalam deskripsi class terdiri dari nama, metoda dan atribut yang dimaksud itu sendiri. Selain itu, class dalam *Diagram* dapat merupakan implementasi dari sebuah interface.

Istilah class itu sendiri menjadi konten utama yang menyimpan dan mengatur informasi di dalam sistem. Istilah ini mengacu kepada seseorang, tempat, kejadian dan sesuatu di mana sistem akan mendapatkan informasi. Class juga dapat mengacu kepada implementasi dari artefak tertentu seperti objek dan bentuk untuk membangun sistem. Elemen ini digambarkan dengan 3 persegi panjang. Nama class terdapat pada persegi atas. Atribut class terdapat pada persegi tengah. Sedangkan metode class terdapat pada persegi bawah. Gambar 2. 3. Menjelaskan masing-masing elemen tersebut.

Atribut dalam *class Diagram* juga memiliki sifat yang terdiri dari:

- *Public*: class dapat dipanggil oleh class manapun.
- *Protected*: class hanya dapat dipanggil oleh class-class di bawah yang mewarisinya.

- *Private*: class dengan sifat atribut ini hanya dapat dipanggil oleh class tertentu yang bersangkutan.

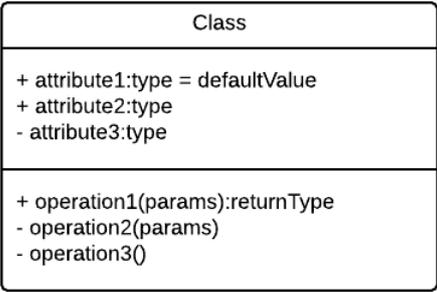
Hubungan antar Class:

1. Pewarisan: hubungan di mana satu class baru merupakan turunan dari class lain. Class yang menjadi anak dalam *Diagram* ini mewarisi atribut dan metode dari class pendahulunya. Lalu anak class juga memiliki fungsionalitas baru. Sifat dari hubungan adalah hierarki.
2. Asosiasi: hubungan di mana satu class memiliki atribut dari class lain yang terhubung dengannya. Tujuannya agar masing-masing class saling mengetahui eksistensi satu sama lain. Sifat dari hubungan adalah statis.
3. Agregasi: hubungan di mana satu class merupakan bagian dari class lain.
4. Dinamis: hubungan di mana satu class menjadi bagian rangkaian pesan yang dikirim oleh class lain menuju class tujuan.

UMMN

UNIVERSITAS  
MULTIMEDIA  
NUSANTARA

**Tabel 2. 2. Elemen dalam class Diagram (Dennis, 2012).**

Istilah elemen	Simbol
<p><b>Class:</b></p> <ul style="list-style-type: none"> <li>• Mewakili seseorang, tempat, atau sesuatu yang sistem harus simpan informasinya.</li> <li>• Tidak sepenuhnya mendeskripsikan operasi yang ada pada class lain.</li> </ul>	
<p><b>Atribut:</b></p> <ul style="list-style-type: none"> <li>• Mewakili properti yang menjelaskan keadaan dari suatu objek.</li> </ul>	<p>+ Nama atribut / nama atribut yang diturunkan</p>
<p><b>Metode:</b></p> <ul style="list-style-type: none"> <li>• Mewakili tindakan atau fungsi yang bisa dijalankan oleh sebuah class.</li> </ul>	<p>+ nama operasi () -nama operasi ()</p>
<p><b>Hubungan:</b></p>	<p><u>1..*</u> kata kerja <u>0..1</u></p>

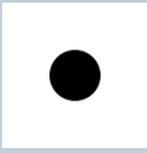
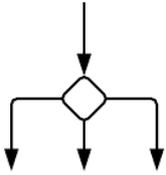
<ul style="list-style-type: none"> <li>• Mewakili hubungan antara beberapa class atau antara class dengan dirinya.</li> </ul>	
---	--

### 2.4.3. Activity Diagram

*Activity Diagram* adalah desain alur dari satu aktivitas dalam sistem yang sedang dirancang. *Diagram* akan menampilkan aktivitas dari awal proses sampai selesai. Lalu disertai dengan keputusan atau pilihan yang terjadi di sepanjang proses. Isi dari tampilan *activity Diagram* berupa kumpulan proses dan jalur yang dapat diambil oleh sistem ketika *activity* terjadi secara umum. Jalur ini diwakili dengan beberapa elemen seperti pada tabel 2.4. Oleh karena itu, *activity Diagram* tidak berorientasi dari sudut pandang tertentu. Setiap dari *activity* ini adalah action di mana setiap transisi dipicu oleh *activity* sebelumnya. *Activity Diagram* mendeskripsikan proses yang dijalankan oleh aktor kepada sistem yang sudah dijelaskan dalam *use case*. Di mana satu *activity Diagram* menjadi kelanjutan dari satu interaksi *use case* atau lebih.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

**Tabel 2. 3. Elemen dalam Activity Diagram (Dennis, Wixom, 2012).**

Istilah elemen	Simbol
<p>Simpul mulai:</p> <ul style="list-style-type: none"> <li>• Mewakili awal dari aktivitas kerja.</li> </ul>	
<p>Simpul selesai:</p> <ul style="list-style-type: none"> <li>• Mewakili akhir dari aktivitas kerja.</li> </ul>	
<p>Aktivitas:</p> <ul style="list-style-type: none"> <li>• Mewakili situasi di mana <i>user</i> atau <i>Software</i> melaksanakan tugas yang menandakan sebuah proses.</li> </ul>	
<p>Simbol keputusan:</p> <ul style="list-style-type: none"> <li>• Sebuah cabang yang ber kondisi di dalam alur <i>Diagram</i>.</li> <li>• Mewakili keputusan dan selalu memiliki dua output atau lebih.</li> </ul>	

## **2.5. Algoritma AHP dan Interaksi Website**

AHP adalah model pembuatan keputusan yang mengelompokkan faktor, aspek atau kriteria yang kompleks menjadi suatu hirarki fungsional berdasarkan persepsi manusia untuk menemukan solusinya (Budiarto, 2012). Model ini memproses suatu masalah kompleks dengan membaginya ke dalam beberapa kelompok. Lalu kelompok-kelompok ini akan diatur dalam struktur tingkatan atau hirarki. Tingkatan dalam hirarki kelompok ini diurutkan dari level tujuan, kemudian dilanjutkan dengan faktor, kriteria, sub-kriteria, dan seterusnya ke bawah di mana alternatif menjadi level paling dasar. Hasilnya, permasalahan akan menjadi sistematis dan nyaman untuk di analisa.

Model ini dikembangkan oleh Thomas L. Saaty (1993). AHP menjadi salah satu model pembuatan keputusan yang paling dikenal. Struktur yang berhirarki menjadi kekuatan utama dari penyebab terkenalnya model ini. Hirarki dari AHP mencari konsekuensi dari kriteria sampai ke bagian terdalam. Lalu, AHP akan mengukur validitas dari kriteria dan alternatif tersebut sampai ke batas toleransi yang paling tidak konsisten. Kemudian, AHP juga memiliki kekuatan dalam membuat perhitungan dari sensitivitas dalam pengambilan keputusan.

## **2.6. Kelebihan dan Kekurangan Algoritma AHP**

Kelebihan AHP (Syaifullah, 2008) dalam membuat keputusan terdapat pada analisis yang diberikan.

1. Konsistensi: AHP menentukan prioritas pilihan dengan menilai dan mengukur konsistensi dari masalah secara logis.

2. Kompleksitas: AHP dapat menyelesaikan permasalahan yang kompleks dengan memanfaatkan pendekatan *website* dan intergrasi secara deduktif.
3. Saling ketergantungan: AHP tidak memerlukan hubungan yang liner. Model ini dapat digunakan pada banyak aspek yang atau secara bebas di dalam *website*.
4. Kesatuan: AHP menyatukan masalah yang luas, kompleks dan berantakan menjadi suatu struktur yang mudah dimengerti.
5. Pengukuran: AHP memiliki metode tersendiri untuk mendapatkan prioritas keputusan beserta skala pengukurannya.
6. Pengulangan proses: Model ini memiliki proses pengulangan untuk meningkatkan pemahaman, penilaian serta seleksi konten dari masalah yang dianalisa.

Kekurangan AHP dalam membuat keputusan (Syaifullah, 2008) terdiri dari:

1. Masalah dalam struktur hirarki: memiliki struktur hirarki berarti membuat AHP harus memiliki banya level di mana level-level di bawah berketergantungan pada level di atasnya. Level atas ini secara langsung menjadi sentral dari model. Kesalahan dalam menentukan tujuan atau faktor yang merupakan level tertinggi akan mempengaruhi level bawah struktur dan seterusnya.
2. Ketergantungan pada *input* utama: seperti halnya dengan masalah pada hirarki, AHP juga dapat dikacaukan lewat *input* utama yang merupakan informasi terkedibel. Namun *input* utama ini sendiri merupakan persepsi

yang bersifat subjektif. Apabila *input* utama ini salah, maka seluruh proses dalam AHP juga tidak akan berlaku lagi.

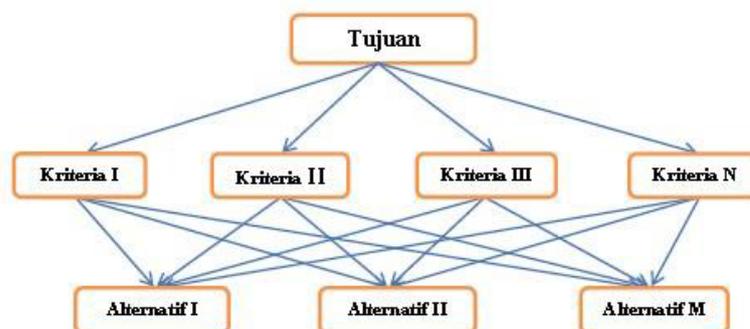
3. Model matematis: Metode yang dijalankan lewat AHP tidak memiliki tahap pengujian dan hasil statistik. Model ini tidak bisa membuktikan kebenaran dari pengambilan keputusan secara konkrit.

## 2.7. Prinsip-prinsip Algoritma AHP

AHP memiliki beberapa prinsip-prinsip yang harus dipelajari dalam menyelesaikan masalah (Budiarto, 2012). Beberapa yang terpenting adalah:

### 2.7.1. Penguraian dengan struktur hirarki

Struktur hirarki dengan contoh seperti gambar 2.5. akan disusun dengan tujuan utama dari pemecahan masalah sebagai level paling atas. Setelah itu dilanjutkan dengan beberapa kriteria untuk memecahkan masalah sebagai tingkatan di bawahnya. Tiap kriteria memiliki intensitas yang berbeda-beda. Mulai dari yang paling cocok sebagai teratas dan sebaliknya. Alternatif dari pemecahan masalah akan dijabarkan di tingkatan level bawah berikutnya.



Gambar 2.2. Struktur Hirarki (Budiarto, 2012)

### 2.7.2. Perbandingan dari penilaian kriteria dan alternatif

Penilaian dilakukan dengan membuat matriks yang berpasangan. Matriks digunakan untuk membuat perbandingan antara aspek-aspek dalam masalah terhadap kriteria dari hirarki beserta tingkatan di atasnya. Tingkatan kepentingan dalam perbandingan ini ditentukan dari skala 1 sampai 9. Skala ini berguna dalam mengekspresikan pendapat. Nilai dan kriteria dapat dijelaskan dalam tabel seperti berikut.

**Tabel 2. 4. Tabel tingkat kepentingan (Budiarto, 2012).**

<b>Tingkat Kepentingan</b>	<b>Definisi</b>	<b>Keterangan</b>
1	Sama pentingnya	Kedua elemen mempunyai pengaruh yang sama.
3	Agak lebih penting yang satu atas yang lainnya	Pengalaman dan penilaian sangat memihak satu elemen dibandingkan dengan pasangannya
4	Cukup penting	Pengalaman dan keputusan menunjukkan kesukaan atas satu aktifitas lebih dari yang lain.
7	Sangat penting	Pengalaman dan keputusan menunjukkan kesukaan yang kuat atas satu aktifitas lebih dari yang lain.
9	Mutlak lebih penting	Satu elemen mutlak lebih disukai dibandingkan dengan pasangannya, pada tingkat keyakinan tertinggi.
2,4,6,8	Nilai tengah diantara dua nilai berdekatan	Bila kompromi dibutuhkan.

### **2.7.3. Membuat matriks perbandingan. – Menentukan Prioritas**

Prioritas dari elemen-elemen kriteria dalam hirarki ditentukan berdasarkan pandangan dari pihak-pihak yang terlibat terhadap proses pengambilan keputusan baik secara langsung maupun tidak langsung. Proses ini juga dapat melibatkan pakar untuk membuat sebuah diskusi. Tahap penentuan prioritas menjadi penentu kontribusi suatu elemen untuk pengambilan keputusan akhir.

### **2.7.4. Melakukan definisi perbandingan pasangan di dalam matriks. – Konsistensi Logis**

Konsistensi dalam struktur AHP harus dapat menghasilkan keputusan yang mendekati valid. Keputusan tersebut dapat dicari dengan mengelompokkan objek dari hirarki agar sesuai dengan keseragaman dan relevansi. Lalu juga dapat dicari melalui tingkat hubungan antar objek yang didasarkan pada kriteria tertentu. Barulah konsistensi dapat dilihat dari rasio indeks hirarki. Rasio yang lebih kecil akan mencerminkan kesempurnaan dari struktur AHP.

Aplikasi AHP:

- 1) Membuat set alternatif
- 2) Perencanaan
- 3) Menentukan prioritas
- 4) Memastikan stabilitas sistem

Tahap penyusunan hierarki ini sendiri bekerja dengan menggambarkan dan menguraikan (Migunani, 2010). Hal itu terjadi dengan memecahkan persoalan yang ada menjadi unsur-unsur terpisah. Pikiran dan ilmu kita akan merincikan unsur-unsur ini sehingga benar-benar terbagi secara hierarkis.

Metode ini akan diadaptasikan ke dalam bentuk pemrograman web untuk dikembangkan di dalam aplikasi. Bahasa pemrograman akan dimanfaatkan untuk menghasilkan dan menyusun hierarki di dalam algoritma. Metode ini sesuai untuk dijadikan pedoman untuk merancang fitur perakitan di aplikasi web untuk Klien.

## **2.8. Pemanfaatan Algoritma AHP**

Metode yang selanjutnya ini akan menjadi dasar dari mekanisme algoritma di dalam salah satu fitur utama di *prototype website* yang mau dikembangkan. Berikut adalah penjelasan dari fitur kinerja yang dimiliki oleh *Website* perakitan PC. Fitur yang menjadi fokus utama proyek ini bekerja dengan memberikan *User* kebebasan dalam merakit sendiri PC yang akan ia beli. Kegiatan perakitan itu akan dilakukan di halaman *Website* secara *Online*. Di salah satu Segmen yang disebut sebagai *Build* ini, *User* akan dihadapi dengan tampilan yang berisikan *template-template* kosong. Jumlah dari *template* akan disesuaikan dengan komponen-komponen yang diperlukan untuk membangun sebuah *desktop* PC. Di sini *user* dapat mengisi *template* kosong itu dengan komponen PC yang sudah disediakan oleh toko atau sedang dijual. Komponen-komponen tersebut akan ditampilkan di *website* dalam bentuk gambar. Lalu gambar-gambar ini akan disediakan di sisi lain halaman

*Website*. Bagian ini diperuntukan agar *User* dapat melihat nama, spesifikasi, ukuran, dan harga dari komponen tersebut. Hasil akhir dari fitur ini adalah gambaran PC jadi yang di inginkan berikut dengan total harga yang diperlukan oleh *User*.

Untuk merealisasikan fitur tersebut tidaklah mudah. Halangan yang terbesar adalah kompatibilitas dari masing-masing komponen kepada komponen yang lainnya. Hal seperti ukuran komponen terutama untuk *Casing desktop* menjadi bagian yang paling dipertimbangkan. Terdapat beberapa *Motherboard* dan PSU yang berukuran besar sehingga hanya bisa ditaruh di *Casing* jenis tertentu. Belum lagi ditambah dengan jumlah RAM dan VGA. Apabila terjadi ketidakcocokan, maka PC tidak dapat dirakit. Hal seperti ini harus terjadi di dalam fitur perakitan *Website*. Sistem pembuatan keputusan akan mengkondisikan apabila sebuah komponen dipilih oleh *user* untuk dipasang di *template* perakitan PC, maka akan ada beberapa komponen tersedia yang tidak dapat dipilih lagi atau diakses karena sudah tidak cocok.

Pengembangan fitur ini tidak bisa dilakukan dengan sembarang cara. Oleh karena itu, metode AHP digunakan di dalam perancangannya. Jadi solusi dari permasalahan ini tertuju kepada pengambilan keputusan sistem. Menurut Devi, Zulfa, Munawaroh, dan Rizky di dalam penelitiannya yang berjudul Sistem Pendukung Keputusan Penjurusan SMA Menggunakan Metode *Analytical hierarchy process* (2013), AHP adalah metode untuk memecahkan suatu situasi yang rumit dan tidak terstruktur ke dalam beberapa komponen ke dalam susunan hierarki . Metode ini bekerja dengan memberi nilai subjektif tentang pentingnya

setiap variabel secara relatif. Lalu, variabel yang memiliki prioritas tertinggi akan dipilih untuk mempengaruhi hasil di dalam situasi tersebut.

Rizky menjelaskan secara teknis bahwa proses pengambilan keputusan ini bekerja dengan memilih suatu alternatif terbaik. Hal tersebut bekerja dengan penstrukturan masalah yang dilanjutkan dengan penentuan alternatif. Modul utamanya adalah dengan memiliki sebuah hierarki fungsional dengan *input* utamanya berupa persepsi manusia.

Kelebihan AHP

1. Memvalidasi kriteria pemilihan sampai ke yang paling penting
2. Merupakan sebuah model pengambilan keputusan yang komprehensif

Berikut adalah prinsip yang mendasar analisis logis dari AHP. Penjelasan dari masing-masing prinsip ini akan dilanjutkan dalam metodologi penelitian.

1. Dekomposisi
2. Prinsip penetapan prioritas

## **2.9. Penggunaan Usability Testing**

Sebuah sistem harus menjalani tahapan pengujian untuk dikaji kualitasnya. Pengujian berfungsi untuk mencari kekurangan dan kesalahan dari *website*. Dengan demikian, hasil akhir dari *website* berjalan sesuai dengan rencana atau desain awal. Tahapan ini sendiri sudah menjadi bagian dari prototyping. Lalu bentuk dari teknik ini bervariasi tergantung jenis sistem. Usability *testing* menjadi pilihan dari teknik

pengujian yang digunakan untuk *website*. Teknik ini digunakan untuk memastikan bahwa fungsi yang dimiliki oleh sistem yang sedang dikembangkan dapat bekerja dengan benar, efisien, efektif, dan memuaskan.

*Usability testing* dapat dijelaskan sebagai uji coba untuk mengetahui bahwa sistem yang dikembangkan dapat digunakan secara layak. Teknik *usability testing* dijalankan apabila sistem sudah memasuki tahap jadi untuk mengidentifikasi masalah sebelum digunakan secara penuh (Gaffney, 2014). Teknik ini lalu digunakan untuk mengetahui faktor yang membuat pengguna mengalami kesulitan dalam menggunakan fungsi di dalam sistem. Kemudian *usability testing* juga membantu pengembang untuk membuat rekomendasi saat tahap pengembangan. Teknik ini bertujuan untuk memahami bagaimana pengguna berinteraksi dengan produk pengembang dan ditingkatkan dengan hasil yang diterima (Smith, 2008). Hasilnya, performa sistem dapat ditingkatkan berdasarkan informasi yang dihasilkan pengujian yang menggunakan *usability testing*.

*Usability testing* memiliki 5 prinsip yang menjadi komponen utamanya. Prinsip tersebut terdiri dari:

1. *Efficiency*

Menunjukkan seberapa cepat *user* dalam menyelesaikan kegiatan yang diberikan.

2. *Satisfaction*

Menjelaskan tingkat kenyamanan pengguna pada saat menggunakan tampilan dari *website*.

### 3. *Memorability*

Kemampuan *user* untuk mengingat bagaimana cara menggunakan *website* apabila sudah tidak menyentuhnya dalam jangka waktu yang lama.

### 4. *Learnability*

Menunjukkan tingkat kemudahan dari *user* dalam menjalankan suatu fungsi saat pertama kali menggunakan sistem ataupun *website* biarpun hanya sekedar dari sisi tampilan saja.

### 5. *Errors*

Menentukan jumlah dari kesalahan yang dilakukan oleh *user* saat menggunakan dan menguji *website*.

## 2.9.1. Metode System Usability Scale (SUS)

System Usability Scale merupakan salah satu bagian dari teknik *usability testing*. Metode ini menilai hasil uji dengan menggunakan skala likert. Metode SUS bekerja dengan memberikan kuesioner yang terdiri dari 10 pertanyaan kepada sampel atau responden dari penelitian. Respondens memiliki 5 pilihan untuk setiap pertanyaan dari setuju sampai sebaliknya. Metode ini juga dapat digunakan kepada *hardware* dan produk jasa. Prinsip *usability testing* yang diukur dalam SUS adalah *efficiency*, *satisfaction*, dan *effectiveness*.

U N I V E R S I T A S  
M U L T I M E D I A  
N U S A N T A R A

