

BAB III

LANDASAN TEORI

3.1 Aplikasi (Application)

Aplikasi (application) adalah software yang dibuat oleh suatu perusahaan komputer untuk mengerjakan tugas-tugas tertentu, misalnya Microsoft Word, Microsoft Excel (Dhanta 2009 : 32). Aplikasi adalah penerapan, penggunaan atau penambahan. Dari pengertian diatas, dapat disimpulkan bahwa aplikasi merupakan software yang berfungsi untuk melakukan berbagai bentuk pekerjaan atau tugas-tugas tertentu seperti penerapan, penggunaan dan penambahan data. Beberapa aplikasi yang digabung bersama menjadi suatu paket kadang disebut sebagai suatu paket atau suite aplikasi (application suite). Contohnya adalah Microsoft Office dan OpenOffice.org, yang menggabungkan suatu aplikasi pengolah kata, lembar kerja, serta beberapa aplikasi lainnya. Aplikasi-aplikasi dalam suatu paket biasanya memiliki antarmuka pengguna yang memiliki kesamaan sehingga memudahkan pengguna untuk mempelajari dan menggunakan tiap aplikasi. Sering kali, mereka memiliki kemampuan untuk saling berinteraksi satu sama lain sehingga menguntungkan pengguna. Contohnya, suatu lembar kerja dapat dibenamkan dalam suatu dokumen pengolah kata walaupun dibuat pada aplikasi lembar kerja yang terpisah.

3.1.1 Klasifikasi Aplikasi

Aplikasi dapat digolongkan menjadi beberapa kelas, antara lain :

a. Perangkat lunak perusahaan (enterprise)

Merupakan perangkat lunak (software) yang dirancang untuk memudahkan aktivitas dalam suatu perusahaan seperti aktivitas akuntansi dan manajemen sumber daya manusia

b. Perangkat lunak infrastruktur perusahaan

Adalah segala perangkat lunak yang mendukung alur kerja data, sistem manajemen basis data dan segala yang mendukung kinerja perusahaan.

- c. Perangkat lunak informasi kerja
Perangkat lunak untuk mendukung manajemen waktu dan sumber daya, manajemen data, dokumentasi, perangkat lunak analitik dan perangkat lunak kolaborasi.
- d. Perangkat lunak media dan hiburan
Perangkat lunak yang di bangun sebagai media digitalhiburan seperti games dan lain-lain.
- e. Perangkat lunak pendidikan
Segala perangkat lunak yang mendukung sistem pendidikan, diantaranya sebagai manajemen ruang kelas, manajemen survei manajemen pelatihan, dan perangkat lunak kesiapan penjualan.
- f. Perangkat lunak pengembangan media
Sebagai pengembang perangkat lunak dari media-media yang sudah ada, seperti Perangkat lunak seni grafis.
- g. Perangkat lunak rekayasa produk.

Yaitu perangkat lunak yang dibangun untuk sebuah kebutuhan manusia, untuk mempermudah pekerjaan manusia dalam berbagai kegiatan produksi, diataranya yaitu rekayasa perangkat keras (hardware) dan rekayasa perangkat lunak.

3.1.2 Aplikasi Android / Android Application

Aplikasi Android adalah software yang dijalankan pada sistem operasi android, dibuat oleh suatu perusahaan atau pun perorangan yang dapat kembangkan secara bebas karena aplikasi android ini berbasis opensource. Aplikasi Android dikembangkan dalam Bahasa pemrograman Java dengan menggunakan kit pengembangan perangkat lunak Android (SDK). SDK ini terdiri dari seperangkat perkakas pengembangan, termasuk debugger, perpustakaan perangkat lunak, emulator handset yang berbasis QEMU, dokumentasi, kode sampel, dan tutorial. Didukung secara resmi oleh lingkungan pengembangan terpadu (IDE) Eclipse, yang menggunakan plugin Android Development Tools (ADT). Perkakas pengembangan lain yang tersedia di antaranya adalah Native Development Kit untuk aplikasi atau ekstensi dalam C atau C++, Google App

Inventor, lingkungan visual untuk pemrogram pemula, dan berbagai kerangka kerja aplikasi web seluler lintas platform.

3.1.3 Sejarah Perkembangan Teknologi Android

Menurut Nazruddin Safaat (2014) Android adalah sebuah *system operasi* untuk perangkat *mobile* berbasis linux yang mencakup *system operasi*, *middleware* dan aplikasi. Android menyediakan *platform* terbuka bagi para pengembang untuk menciptakan aplikasi mereka. Awalnya, *Google Inc.* membeli *Android Inc.* yang merupakan pendatang baru yang membuat peranti lunak untuk ponsel/*smartphone*. Kemudian untuk mengembangkan android, dibentuklah *Open Handset Alliance*, konsorsium dari 34 perusahaan peranti keras, peranti lunak, dan telekomunikasi, termasuk *Google*, *HTC*, *Intel*, *Motorola*, *Qualcomm*, *T-Mobile*, dan *Nvidia*.

Pada saat perilisannya Android, 5 November 2007, Android bersama *Open Handset Alliance* menyatakan mendukung pengembangan *open source* pada perangkat mobile. Di lain pihak, Google merilis kode-kode Android dibawah lisensi Apache, sebuah lisensi perangkat lunak dan *open platform* perangkat seluler. Di dunia ini terdapat dua jenis distributor system operasi Android. Pertama yang mendapat dukungan penuh dari *Google* atau *Google Mail Services (GMS)* dan kedua adalah yang benar-benar bebas distribusinya tanpa dukungan langsung *Google* atau dikenal sebagai *Open Handset Distribution (OHD)*.

Sekitar September 2007 Google mengenalkan Nexus One, salah satu jenis *smartphone* yang menggunakan Android sebagai system operasinya. Telepon seluler ini diproduksi oleh HTC Corporation dan tersedia di pasaran pada 5 Januari 2010. Pada 9 Desember 2008, diumumkan anggota baru yang bergabung dalam program kerja *Android ARM Holdings*, *Atheros Communications*, diproduksi oleh *asustek Computer Inc*, *Garmin Ltd*, *Softbank*, *Sony Ericsson*, *Toshiba Corp*, dan *Vodafone Group Plc*. Seiring pembentukan *Open Handset Alliance*, *OHA* mengumumkan produk perdana mereka, Android, perangkat mobile yang merupakan modifikasi kernel *Linux 3.6*. Sejak Android dirilis telah dilakukan berbagai pembaruan berupa perbaikan bug dan penambahan fitur baru.

Pada masa saat ini kebanyakan vendor-vendor *smartphone* sudah memproduksi *smartphone* berbasis Android, vendor-vendor itu antara lain *HTC, Motorola, Samsung, LG, Acer, Philips, T-Mobile, Nexian, IMO, Asus* dan masih banyak lagi vendor *smartphone* didunia yang memproduksi Android, hal ini karena Android itu adalah *system operasi* yang *Open Source* sehingga bebas didistribusikan dan dipakai oleh vendor manapun.

Tidak hanya menjadi *system operasi* di *smartphone*, saat ini Android menjadi pesaing utama dari Apple pada *system operasi* Tablet PC. Pesatnya pertumbuhan Android selain faktor yang disebutkan diatas adalah karena Android itu sendiri adalah *platform* yang sangat lengkap baik itu *system operasinya, aplikasi dan Tool Pengembangan, Market aplikasi Android* serta dukungan yang sangat tinggi dari komunitas *Open Source* di dunia, sehingga Android terus berkembang pesat baik dari segi teknologi maupun dari segi jumlah *device* yang ada di dunia Android dipuji sebagai “*platform mobile* pertama yang lengkap, terbuka, dan bebas” (Nazruddin Safaat H., 2014).

- a. Lengkap (*Complete Platform*): para desainer dapat melakukan pendekatan yang komprehensif ketika merekasedang mengembangkan *platform* Android. Android merupakan *system operasi* yang aman dan banyak menyediakan *tools* dalam membangun *software* dan memungkinkan untuk peluang pengembangan aplikasi.
- b. Terbuka (*Open Source Platform*): *platform* Android disediakan melalui lisensi *Open Source*. Penhembang dapat dengan bebas untuk mengembangkan aplikasi. Android sendiri menggunakan Linux Kernel 3.6.
- c. Free (*Free Platform*): Android adalah *platform/aplikasi* yang bebas untuk *develop*. Tidak ada lisensi atau biaya *royalty* untuk dikembangkan pada *platform* Android. Tidak ada biaya keanggotaan diperlukan. Aplikasi untuk Android dapat didistribusikan dan diperdagangkan dalam bentuk apapun.

Android merupakan generasi baru *platform mobile, platform* yang memberikan pengembangan untuk melakukan pengembangan sesuai dengan yang diharapkannya. *System operasi* yang mendasari Android dilisensikan di bawah *GNU, General Public Lisensi Versi 2 (GPLv2)*, yang sering dikenal dengan istilah “*copyleft*” lisensi dimana setiap perbaikan pihak ketiga harus terus jatuh di bawah

terms. Android didistribusikan di bawah lisensi *Apache Software* (ASL/Apache2), yang memungkinkan untuk distribusi kedua dan seterusnya. Komersialisasi pengembang (produsen *handset* khususnya) dapat memilih untuk meningkatkan *platform* tanpa harus memberikan perbaikan mereka kepada masyarakat *Open Source*. Sebaliknya, pengembang dapat keuntungan dari perangkat tambahan seperti perbaikan dan mendistribusikan ulang pekerjaan mereka dibawah lisensi apapun yang mereka inginkan.

Pengembang memiliki beberapa pilihan ketika membuat aplikasi yang berbasis Android. Kebanyakan pengembang menggunakan Eclipse yang tersedia secara bebas untuk merancang dan mengembangkan aplikasi Android. Eclipse adalah *IDE* yang paling populer untuk pengembangan Android, karena memiliki Android *plug-in* yang tersedia untuk memfasilitasi pengembang Android. Selain itu Eclipse juga mendapat dukungan langsung dari Google untuk menjadi *IDE* pengembangan aplikasi Android, ini terbukti dengan adanya penambahan *plugins* untuk Eclipse untuk membuat *project* Android di mana *source software* langsung dari situs resminya Google. Tetapi hal diatas tidak menutup kemungkinan untuk menggunakan *IDE* yang lain seperti Netbeans untuk melakukan pengembangan.

Aplikasi Android dapat dikembangkan pada *system operasi* berikut:

- a. Windows XP Vista/7 dan 8
- b. Mac OS X (Mac OS X 10.4.8 atau lebih baru)
- c. Linux

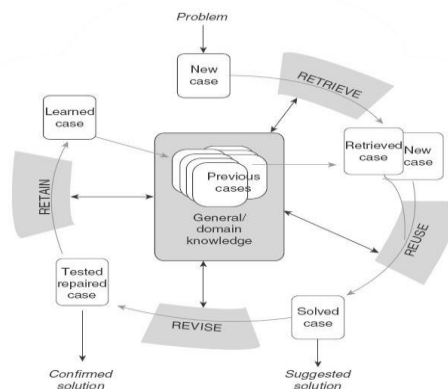
3.2 Definisi Case Based Reasoning

Case-Based Reasoning (CBR) didefinisikan sebagai sebuah metodologi untuk penyelesaian masalah dengan memanfaatkan pengalaman sebelumnya (Main dkk, 2001). *Case Based Reasoning* (CBR) merupakan sebuah paradigma utama dalam penalaran otomatis (*automated reasoning*) dan mesin pembelajaran (*machine learning*). Di dalam CBR, seseorang yang melakukan penalaran dapat menyelesaikan masalah baru dengan memperhatikan kesamaannya dengan satu atau beberapa penyelesaian dari permasalahan sebelumnya.

(Mulyana Sri dan Hartati Sri, 2009) CBR dapat memiliki makna yang berbeda, tergantung tujuan dari penalaran:penyesuaian dan penggabungan solusi

sebelumnya untuk menyelesaikan sebuah masalah baru, menjelaskan kondisi baru sesuai kondisi yang sama berdasarkan pengalaman sebelumnya, sebuah kritik terhadap solusi berdasarkan kasus sebelumnya, menemukan alasan dari kondisi sebelumnya untuk memahami situasi baru atau membangun sebuah solusi yang disepakati berdasarkan kasus sebelumnya (Mantaras dkk, 2006).

Aamodt dan Plaza (Aamodt dan Plaza, 1994) menjelaskan sebuah CBR sebagai sebuah siklus yang disingkat 4 R yaitu, *Retrieve*, *Reuse*, *Revise* dan *Retain* seperti pada gambar 2.2 berikut ini :



Gambar 3.1 Siklus *Case-based Reasoning*

Secara singkat, tahap-tahap penyelesaian masalah dengan metode *Case Based Reasoning* ada 4 yaitu sebagai berikut :

1. Retrieve (Memperoleh kembali)

Pada proses *Retrieve* ini kita mendapatkan kembali kasus yang sama atau yang mirip dengan kasus baru yang baru kita temui. Dalam proses ini, tahapan yang dapat kita lakukan adalah identifikasi masalah, memulai pencocokan, dan seleksi.

2. Reuse (Menggunakan kembali)

Pada proses *reuse* ini, sistem akan melakukan pencarian masalah pada database melalui identifikasi masalah baru. Setelah itu, sistem akan menggunakan kembali informasi permasalahan yang pernah terjadi tersebut yang memiliki kesamaan untuk menyelesaikan permasalahan yang baru. Proses *reuse* dipusatkan pada dua aspek. Pertama, perbedaan antara kasus sebelumnya dengan kasus sekarang. Kedua, bagian yang sudah diperoleh akan dikirimkan menjadi kasus.

3. **Revise (Meninjau kembali / memperbaiki)**

Pada proses *revise* ini akan dilakukan tinjauan kembali atau memperbaiki solusi- solusi yang sudah didapat pada masalah tersebut. Ada dua tugas pokok dari tahapan *revise* ini, di proses ini solusi yang sudah diperoleh dari proses *raise* akan dievaluasi kembali. Jika berhasil, maka akan langsung dilanjutkan ke proses selanjutnya yaitu proses *retain*. Jika tidak, sistem akan memperbaiki lagi solusi kasus yang diperoleh dari proses *retain* dengan menggunakan domain spesifik pengetahuan.

4. **Retain (Menyimpan)**

Pada proses *retain*, bisa dibilang proses ini adalah yang terakhir di dalam system *Case Based Reasoning*. Di dalam proses system ini akan menyimpan permasalahan yang baru lalu dimasukan ke dalam basis pengetahuan, setelah itu akan digunakan untuk menyelesaikan permasalahan yang akan datang.

3.2.1 **Kelebihan dan Kekurangan Case Based Reasoning**

1. **Kelebihan Case Based Reasoning**

- Memecahkan masalah dengan mudah karena dapat mengambil solusi dengan cepat dan tepat.
- Semakin banyak pengalaman yang tersimpan di dalam system maka system akan semakin pintar dalam menemukan solusi untuk sebuah kasus.
- Biasanya langsung fokus pada fitur terpenting pada masalah tersebut.

2. **Kekurangan Case Based Reasoning**

- Tidak menjamin solusi yang didapat itu menjadi solusi terbaik atau maksimal, karena dalam system *Case Based Reasoning* ini sangat bergantung pada kasus yang pernah terjadi, maka dari itu jika solusi dari kasus yang pernah terjadi itu salah, maka dalam hal ini tahapan *revise* sangat diperlukan untuk mengurangi tingkat kesalahannya.
- Namun dalam hal ini, jika semakin banyak pengalaman yang tersimpan di basis data, maka tidak menutup kemungkinan

dalam menyelesaikan suatu kasus baru akan menjadi lama. Itu dikarenakan, system ini akan mencari kasus-kasus yang paling mirip.

3.2.2 Contoh kasus yang berkaitan dengan CBR

Atribut yang dibutuhkan dan perhitungan studi kasus atribut penilaian dosen penerima beasiswa untuk melanjutkan pendidikan S3 adalah sebagai berikut:

Tabel 3.1 Kode dan penentuan atribut

Kode Atribut	Ketentuan atribut
A1	Usia
A2	Ipk
A3	Prestasi
A4	Kedisiplinan
A5	Loyalitas Kerja
A6	Skor toefl

Table 3.1 ini berisikan kode atribut yang akan dipakai sebagai acuan menentukan calon penerima beasiswa. Meliputi usia, nilai IPK pendidikan kuliah terakhir, prestasi apa saja yang didapat dosen selama menjalani pendidikan, kedisiplinan selama mengajar, loyalitas kerja di bidang yang dipegang, dan skor toefl yang di dapat.

Tabel 3.2 Variable dan Bobot

Variable	Bobot (Nilai)
Sesuai/Penting	1
Kurang Sesuai/sedang	0.50-0.90
Sangat tidak sesuai/tidak penting	0.01-0.49

Table 3.2 berisi bobot nilai yang diakurasi dengan symbol sesuai/penting yang bernilai 1, kurang sesuai/sedang bernilai 0.50-0.90, sangat tidak sesuai/tidak penting dengan nilai 0.01-0.49.

Contoh Kasus :

1. Data dosen calon penerima beasiswa

Dalam tabel 3.3 menjelaskan data dosen pertama yang menjadi seleksi calon dosen penerima beasiswa, dengan memasukkan data dari kampus tempat dia mengajar. Setelah diperoleh nilai yang ada akan diteruskan dengan melakukan perhitungan dengan rumus CBR yang akan dijelaskan perhitungannya dibawah :

Nama	Atribut	Nilai
Adi Prastiyo	Usia	32 tahun
	Ipk	3.50
	Toefl	480
	Kedisiplinan	Disiplin
	Loyalitas	Telah bekerja 3 th
	Prestasi	Banyak Prestasi

Tabel 3.3 Data Dosen

Perhitungan untuk menghitung apakah dosen tersebut menerima beasiswa atau tidak adalah sebagai berikut :

$$\begin{aligned}
 \text{Similarity (T,S)} &= \frac{S_1 * W_1 + S_2 * W_2 + \dots + S_n * W_n}{W_1 + W_2 + W_3 + \dots + W_n} \\
 &= \frac{\{(1*1) + (1*1) + (1*1) + (1*1) + (1*1) + (1*1)\}}{1+1+1+1+1+1} \\
 &= \frac{1+1+1+1+1+1}{6} \\
 &= \frac{6}{6} \\
 &= 1
 \end{aligned}$$

Dari perhitungan diatas diperoleh nilai akumulasi perhitungan adalah 1

2. Data dosen calon penerima beasiswa

Dalam tabel 3.4 menjelaskan data dosen pertama yang menjadi seleksi calon dosen penerima beasiswa, dengan memasukkan data dari kampus tempat dia mengajar. Setelah diperoleh nilai yang ada akan diteruskan dengan melakukan perhitungan dengan rumus CBR yang akan dijelaskan perhitungannya dibawah

Tabel 3.4 Data Dosen 2

Nama	Atribut	Nilai
Nurdin	Usia	32 tahun
	Ipk	3
	Toefl	440
	Kedisiplinan	Kurang disiplin
	Loyalitas	Telah bekerja 1 th
	Prestasi	Banyak memadai

$$\begin{aligned} \text{Similarity (T,S)} &= \frac{S1*W1+S2*W2+\dots +Sn*Wn}{W1+W2+W3\dots+WN} \\ &= \frac{\{(1*0.3)+(1*0.5)+(1*0.5) + (1*0.5) + (1*0.5) + (1*0.6)\}}{1+1+1+1+1+1} \\ &= \frac{2.9}{6} \\ &= 0.48 \end{aligned}$$

Dari perhitungan diatas diperoleh nilai akumulasi perhitungan adalah 0.48

3. Tabel hasil perhitungan penentuan beasiswa

Dari dua perhitungan diatas maka didapat suatu nilai setiap dosen yang dilihat dari perhitungan CBR. Dan dapat dilihat hasil dosen mana yang layak mendapatkan beasiswa melalui tabel 3.5.

Tabel 3.5 Tabel Hasil perhitungan

Nama	Jumlah Hasil	Keterangan
Adi Prastiyo	1	Layak
Nurdin	0.48	Tidak layak

3.2.3 Perancangan Desain Antarmuka

Desain Antarmuka atau yang lebih sering dikenal dengan istilah *Design Interface* adalah desain untuk komputer, peralatan, mesin, perangkat komunikasi *mobile*, aplikasi perangkat lunak, dan situs web yang berfokus pada pengalaman dan interaksi pengguna (Jogiyanto, 2011). Tujuan dari penggunaan *design interface* ini adalah untuk membuat interaksi pengguna sesederhana dan seefisien mungkin. Bagaimana user berinteraksi dengan komputer menggunakan tampilan antarmuka (*interface*) yang ada pada layar komputer. *Design Interface* ini sangat penting karena akan sangat berpengaruh pada *user* dalam menggunakan atau berkomunikasi dengan komputer. Apabila suatu program sulit untuk digunakan, maka hal ini akan memaksa *user* untuk melakukan suatu kesalahan saat menggunakan program tersebut.

Dari tahun ke tahun teknologi berkembang sangat pesat oleh sebab itu, perkembangan dari *Human Computer Interface (HCI)* menjadi sangat penting bagi seorang *analyst* yang merancang atau menciptakan suatu *software*.

Di mana harus memperhatikan kaidah-kaidah atau prinsip prinsip dasar seperti *User Friendly* dan *User Oriented*. Sekarang ini banyak berbagai jenis komputer sistem yang ada seperti: *desktop*, *laptop*, *notebook*, *PDA*, dan yang lainnya. Oleh sebab itu, desain antarmuka ini berperan sangat penting karena, jika berbeda *device* maka tampilan antarmuka akan berbeda juga. Seperti contoh antarmuka di laptop akan berbeda dengan antarmuka di *PDA*. Tergantung dari seberapa besar *screen* dan *support* sistemnya.

User Design Interface ini bukan lagi hanya sebatas pada tampilan. Tapi juga mencakup suara dan gerakan tubuh. *Voice Recognition* merupakan salah satu *Design Interface* yang melibatkan suara. Aplikasi yang sudah diterapkan adalah dalam *translate.google.com* sehingga kita tidak perlu untuk mengetik kata-

kata yang ingin kita terjemahkan. Contoh lain nya adalah gerakan tubuh yang ada dalam Wii. Kalian yang senang main *game* pasti tahu *device* ini di mana ada suatu remote yang dipegang guna mendeteksi setiap gerakan yang pengguna lakukan.

3.2.4 *Cascading Sytle Sheets (CSS)*

Cascading Style Sheets (CSS) adalah suatu bahasa stylesheet yang digunakan untuk mengatur tampilan suatu dokumen yang ditulis dalam bahasa markup (Kadir, 2013). Penggunaan yang paling umum dari CSS adalah untuk memformat halaman web yang ditulis dengan *HTML* dan *XHTML*. Walaupun demikian, bahasanya sendiri dapat dipergunakan untuk semua jenis dokumen *XML* termasuk *SVG* dan *XUL*. Spesifikasi CSS diatur oleh World Wide Web Consortium (W3C). CSS digunakan oleh penulis maupun pembaca halaman web untuk menentukan warna, jenis huruf, tata letak, dan berbagai aspek tampilan dokumen. CSS digunakan terutama untuk memisahkan antara isi dokumen (yang ditulis dengan *HTML* atau bahasa *markup* lainnya) dengan presentasi dokumen (yang ditulis dengan CSS). Pemisahan ini dapat meningkatkan aksesibilitas isi, memberikan lebih banyak keleluasaan dan control terhadap tampilan, dan mengurangi kompleksitas serta pengulangan pada stuktur isi. *Java Script*

Java Script adalah salah satu bagian dari bahasa pemrograman *Java* yang dibuat oleh *Sun Microsystem*. *Java Script* adalah bahasa *script* yang langsung dimasukkan ke dalam dokumen *HTML*, sehingga tidak memerlukan kompiler lagi (Kadir, 2013).

3.2.5 *Ajax*

AJAX adalah singkatan dari *Asynchronous JavaScript and XML* (Kadir, 2013). Pada dasarnya *ajax* menggunakan *XMLHttpRequest object Javascript* untuk membuat *request* ke *server* secara *asynchronous* atau tanpa melakukan *refresh* halaman *website*. Yang dibutuhkan agar *ajax* dapat berjalan adalah *javascript* harus di *enable* pada *browser* yang digunakan. Walaupun *javascript* merupakan dasar dari *Ajax*, dimana *javascript* sangat susah pada implementasi dan *maintenance*, tetapi *Ajax* memiliki struktur pemrograman yang lebih mudah untuk dipahami. Anda tinggal membuat *object XMLHttpRequest* dan memastikan

object tersebut terbentuk dengan benar. Kemudian menentukan kemana hasilnya akan ditampilkan atau dikirim.

Ajax dapat digunakan untuk melakukan banyak hal, seperti *loading* halaman *HTML* tanpa *refresh* halaman *web*, *validasi form* dan banyak lagi yang bisa dilakukan dengan *ajax*. Di tutorial ini *Ajax* akan dipadukan dengan bahasa pemrograman *PHP* yang sangat *powerfull*. *Ajax* bertugas melakukan request ke *web server* dan *PHP* yang berada di *server* akan melakukan apa yang diminta oleh *Ajax*, mengirim hasilnya ke *web browser* dan *Ajax* menampilkannya kepada *user*. Manfaat paling utama ketika menggunakan teknologi *Ajax* pada halaman *website* adalah bisa meningkatkan pengalaman pengguna / *UX*. *Website* menjadi lebih menarik, dan terasa lebih cepat karena tidak perlu berganti halaman ketika melakukan sebuah aksi. Dengan begitu berarti mengurangi *reload* seluruh halaman, dan hanya akan merefresh bagian tertentu saja dari halaman *website*. *Ajax* meningkatkan kinerja browser dan memfasilitasi kecepatan *browsing* yang lebih tinggi sehingga memberikan pengalaman pengguna yang responsif.

3.2.6 *Web Service*

Web Service adalah standard yang digunakan untuk pertukaran data antar aplikasi atau sistem. Masing-masing aplikasi yang melakukan pertukaran data bisa di tulis dengan bahasa pemrograman yang berbeda atau berjalan pada *platform* yang berbeda. Contoh implementasi dari *web service* adalah *SOAP* dan *REST*.

REST adalah singkatan dari *REST stands for REpresentational State Transfer*. Merupakan standard dalam arsitektur *web* yang menggunakan *Protocol HTTP* untuk pertukaran data. Konsep *REST* pertamakali diperkenalkan oleh Roy Fielding pada tahun 2000. Cara kerjanya, *REST server* menyediakan jalur untuk akses *resource* atau data, sedangkan *REST client* melakukan akses *resource* dan kemudian menampilkan atau menggunakannya. *Resource* yang dihasilkan sebenarnya berupa teks, namun formatnya bisa bermacam-macam tergantung keinginan *developer*, umumnya adalah *JSON* dan *XML*.

Dalam mengakses sebuah *resource*, *REST* juga menggunakan konsep *URI (Uniform Resource Identifier)* dimana ada *method* yang digunakan, *by default* adalah *GET*. Berikut ini *method-method* yang mendukung *REST* :

- *GET*, digunakan untuk *resource* yang hanya perlu dibaca saja (*read only*).
- *PUT*, digunakan digunakan untuk membuat/*create resource* baru.
- *DELETE*, digunakan digunakan untuk menghapus suatu *resource*.
- *POST*, digunakan digunakan untuk mengupdate suatu *resource*.
- *OPTIONS*, digunakan digunakan untuk mendapatkan operasi yang di *support* pada *resource*.

3.2.7 Cara Kerja *Restful Web Service*

Sebuah *client* mengirimkan sebuah data atau *request* melalui *HTT Request* dan kemudian *server* merespon melalui *HTTP Response*. Komponen dari *http request* adalah :

- *Verb, HTTP method* yang digunakan misalnya *GET, POST, DELETE, PUT* dan lain-lain.
- *URI, Uniform Resource Identifier (URI)* untuk mengidentifikasi lokasi *resource* pada *server*. *HTTP Version*, menunjukkan versi dari *HTTP* yang digunakan, contoh *HTTP*.
- *Request Header*, berisi metadata untuk *HTTP Request*. Contoh, *type client/browser*, format yang didukung oleh *client*, format dari *body* pesan, setting *cache* dan lain-lain.
- *Request Body*, konten dari data.

Sedangkan komponen dari *http response* adalah:

- *Status/Response Code*, mengindikasikan status *server* terhadap *resource* yang di *request*. misal : 404, artinya *resource* tidak ditemukan dan 200 *response OK*.
- *HTTP Version*, menunjukkan versi dari *HTTP* yang digunakan, contoh *HTTP*
- *Response Header*, berisi metadata untuk *HTTP Response*. Contoh, *type server*, panjang *content*, *type content*, waktu *response*, dan lain-lain.
- *Response Body*, konten dari data yang diberikan.

Dalam arsitektur *REST*, seharusnya tidak boleh menyimpan *state* atau penanda dari *client* di *server*. Hal ini disebut sebagai *stateless* atau *statelessness*. Contohnya pada kasus *session*, dimana *session* merupakan penanda *client* yang disimpan di *server*. Sehingga *session* pada arsitektur *REST* tidak diperbolehkan.

Keuntungan dari *stateless* antara lain :

- *Web services* dapat melayani masing-masing *request* secara *independent*.
- *Web services* tidak perlu mempertahankan *state user*, hal ini tentu membuat desain *application* lebih sederhana.
- *HTTP* merupakan protokol yang bersifat *stateless*.

Sedangkan kerugian dari *stateless* yaitu *web services* membutuhkan informasi tambahan pada masing-masing *request* untuk menterjemahkan *request* dan *state* dari *client*. Target utama dari *stateless* sebenarnya adalah untuk *scale-up* concurrent akses terhadap aplikasi. Karena aplikasi tidak perlu menyimpan *state* dari *client* sehingga meningkatkan jumlah *service* terhadap *request* pada satu waktu.

3.3 Pemasaran

Pemasaran adalah salah satu kegiatan pokok yang perlu dilakukan oleh perusahaan baik itu perusahaan barang atau jasa dalam upaya untuk mempertahankan kelangsungan hidup usahanya. Hal tersebut disebabkan karena pemasaran merupakan salah satu kegiatan perusahaan, di mana secara langsung berhubungan dengan konsumen. Maka kegiatan pemasaran dapat diartikan sebagai kegiatan manusia yang berlangsung dalam kaitannya dengan pasar. Kotler (2012) mengemukakan berarti bekerja dengan pasar sasaran untuk mewujudkan pertukaran yang potensial dengan maksud memuaskan kebutuhan dan keinginan manusia. Sehingga dapat dikatakan bahwa keberhasilan pemasaran merupakan kunci kesuksesan dari suatu perusahaan. Menurut Stanton (2013), definisi pemasaran adalah suatu sistem keseluruhan dari kegiatan-kegiatan bisnis yang ditujukan untuk merencanakan, menentukan harga, mempromosikan dan

mendistribusikan barang atau jasa yang memuaskan kebutuhan baik kepada pembeli yang ada maupun pembeli potensial.

Berdasarkan definisi tersebut di atas, dapat ditarik kesimpulan bahwa pemasaran merupakan usaha terpadu untuk menggabungkan rencana-rencana strategis yang diarahkan kepada usaha pemuas kebutuhan dan keinginan konsumen untuk memperoleh keuntungan yang diharapkan melalui proses pertukaran atau transaksi. Kegiatan pemasaran perusahaan harus dapat memberikan kepuasan kepada konsumen bila ingin mendapatkan tanggapan yang baik dari konsumen. Perusahaan harus secara penuh tanggung jawab tentang kepuasan produk yang ditawarkan tersebut. Dengan demikian, maka segala aktivitas perusahaan, harusnya diarahkan untuk dapat memuaskan konsumen yang pada akhirnya bertujuan untuk memperoleh laba.

3.4 Teori Perusahaan

Manajemen pemasaran jika kita perhatikan kata tersebut berasal dari dua kata yang pertama kata “manajemen” dan yang kedua kata “pemasaran”, jadi pengertian manajemen pemasaran berasal dari gabungan dua pengertian. Kata manajemen sering di definisikan sebagai sebuah proses rangkaian kegiatan perencanaan, pengorganisasian, penggerakan dan *controlling* (pengawasan) yang dilakukan untuk mencapai target atau tujuan yang telah ditentukan sebelumnya, dan definisi pemasaran yaitu suatu kegiatan didalam perekonomian yang dapat membantu menciptakan nilai ekonomi, nilai tersebut menentukan harga produk atau jasa. Faktor terpenting dalam menciptakan nilai tersebut adalah produksi, pemasaran dan konsumsi. Jadi pemasaran merupakan penghubung antaran kegiatan produksi dan konsumsi.

Manajemen Pemasaran adalah salah satu kegiatan pokok yang dilakukan oleh perusahaan untuk mempertahankan kelangsungan perusahaan. Tujuan dari manajemen pemasaran adalah berkembang dan mendapatkan laba yang lebih baik. Manajemen pemasaran harus dilakukan sebelum barang-barang diproduksi dan tidak berakhir saat telah melakukan penjualan. Manajemen pemasaran harus juga memberikan kepuasan kepada konsumen, agar penjualan terus dapat dilakukan (Handoko, 2014).

Menurut Kotler (2012) manajemen Pemasaran adalah kegiatan penganalisaan, perencanaan, pelaksanaan, dan pengawasan program-program yang bertujuan menimbulkan pertukaran dengan pasar yang dituju dengan maksud untuk mencapai tujuan perusahaan. Terdapat 3 (tiga) unsur yang penting dalam manajemen pemasaran diantaranya sebagai berikut:

1. Orientasi pada konsumen atau pembeli

Pada umumnya produsen menghasilkan produk untuk upaya memenuhi kebutuhan dan kepuasan konsumen, dengan tujuan untuk mendapatkan keuntungan. Konsumen ialah orientasi paling utama yang harus di pertimbangkan dalam segala macam bentuk strategi bisnis. Demikian juga dalam proses marketing atau penjualan, maka dalam bisnis selalu mengedepankan pihak konsumen yang merupakan strategi dari bagian konsep pemasaran yang perlu di perhatikan. Sebab konsumen bisa di bilang sebagai raja, dan pihak produsen atau perusahaan adalah pelayan yang harus melayani kebutuhan raja (konsumen) yaitu dengan berbagai macam tawaran yang menarik, baik itu dari segi produk yang dihasilkan maupun dari segi pelayanan yang di sediakan. Manajemen pemasaran menyetujui konsep pemasaran yang terbaik dengan menjadikan konsumen sebagai prioritas yang paling utama.

2. Perencanaan kegiatan-kegiatan pemasaran secara menyeluruh

Manajemen pemasaran menyetujui adanya pengaturan secara dinamis sebagai bentuk penyusunan kegiatan-kegiatan dari pemasaran secara menyeluruh. Segala keperluan dalam bidang pemasaran, strategi, implementasi, penganalisaan, *controlling* (pengawasan) dan lain-lain yang berkaitan dengan kegiatan pemasaran harus dilakukan secara tersusun, dan diatur secara rinci atau detail dan jelas, sehingga bisa mempermudah dalam proses pelaksanaan dan *controlling* (pengawasan). Manajemen pemasaran menyetujui suatu upaya yang sistematis yang jelas sehingga dapat dilakukan penganalisaan dan *controlling* (pengawasan) terhadap hasil yang di dapat. Evaluasi terhadap suatu manajemen pemasaran salah satunya juga di dasarkan atas kegiatan penyusunan secara menyeluruh tersebut.

3. Kepuasan konsumen

Kepuasan konsumen menjadi unsur yang penting juga yang harus di perhatikan dalam menyusun strategi pemasaran. Manajemen pemasaran yang bagus yaitu menyetujui adanya hasil keputusan para konsumen yang maksimal sebagai akibat dari proses pemasaran yang berjalan. Kepuasan dari konsumen tidak hanya diukur dan dilihat dari bagaimana kualitas produk yang dihasilkan, akan tetapi dari bagaimana cara maupun strategi pemasaran yang dijalankan.

3.5 *Hypertext Preprocessor (PHP)*

Menurut Riyanto (2015:34) “*PHP* adalah pemrograman *interpreter* yaitu proses penerjemahan baris kode sumber menjadi kode mesin yang dimengerti komputer secara langsung pada saat baris kode dijalankan”. *Php* disebut juga pemrograman *Server Side Programming*, hal ini dikarenakan seluruh prosesnya dijalankan pada *server*. *PHP* adalah suatu bahasa dengan hak cipta terbuka atau yang juga dikenal dengan *open source* yaitu pengguna data mengembangkan kode-kode fungsi sesuai kebutuhannya.

Menurut Shalahuddin (2013:22) “*PHP (Perl Hypertext Preprocessor)* adalah bahasa *server-side-scripting* yang menyatu dengan *HTML* untuk membuat halaman web yang dinamis”. Dengan menggunakan program *PHP*, sebuah *website* akan lebih interaktif dan dinamis. Kelebihan-kelebihan dari *PHP* yaitu:

1. *PHP* merupakan sebuah bahasa *script* yang tidak melakukan sebuah kompilasi dalam penggunaannya. Tidak seperti halnya bahasa pemrograman aplikasi yang lainnya.
2. *PHP* dapat berjalan pada *web server* yang dirilis oleh Microsoft, seperti *IIS* atau *PWS* juga pada *apache* yang bersifat *open source*.
3. Karena sifatnya yang *open source*, maka perubahan dan perkembangan *interpreter* pada *PHP* lebih cepat dan mudah, karena banyak *milis-milis* dan *developer* yang siap membantu pengembangannya.
4. Jika dilihat dari segi pemahaman, *PHP* memiliki referensi yang begitu banyak sehingga sangat mudah untuk dipahami.
5. *PHP* dapat berjalan pada 3 *operating sistem*, yaitu: *Linux*, *unix*, dan *windows*, dan juga dapat dijalankan secara *runtime* pada suatu *console*.

Sedangkan kekurangan dari penggunaan bahasa pemrograman *PHP* adalah sebagai berikut:

1. Tidak terpisahnya script logik dan tampilan membuat *PHP* tidak ideal untuk pengembangan aplikasi skala besar. Tapi hal ini dapat diakali dengan menggunakan framework berkonsep *MVC (Model-View-Controller)* seperti *CodeIgniter* dll.
2. Variabel global *PHP* dapat berasal dari pengunjung web baik dari form *HTML* maupun dari *Address Bar browser*, sehingga apabila programmernya ceroboh maka pengguna web dapat mengirimkan script-script yang akan merubah respon dari web.
3. *Source Code* dapat dilihat oleh *client*..
4. Pemrograman berorientasi objek belum didukung sepenuhnya oleh *PHP*, tapi setau saya para pengembang *PHP* terus bekerja untuk menutupi kekurangan-kekurangan yang ada pada *PHP*.
5. *PHP* Tidak mengenal *Package*.
6. Jika tidak di encoding, maka kode *PHP* dapat dibaca semua orang & untuk mengencodinya dibutuhkan tool dari *Zend* yang mahal sekali biayanya.
7. *PHP* memiliki kelemahan keamanan. Jadi Programmer harus jeli & berhati-hati dalam melakukan pemrograman & Konfigurasi *PHP*.

3.6 *SQL*

***SQL* adalah** suatu bahasa (language) yang digunakan untuk mengakses data di dalam sebuah database relasional (Bunafit, 2015). *SQL* sering juga disebut dengan istilah query, dan bahasa *SQL* secara praktiknya digunakan sebagai bahasa standar untuk manajemen database relasional. Hingga saat ini hampir seluruh server database atau software database mengenal dan mengerti bahasa *SQL*. Dalam penggunaan *SQL* terdapat beberapa perintah yang berguna untuk mengakses dan memanajemen data yang terdapat dalam database. Jenis perintah *SQL* secara umum dibagi kepada tiga sub perintah, yaitu *DDL (Data Definition Language)*, *DML (Data Manipulation Language)*, dan *DCL (Data Control Language)*. Ketiga sub perintah tersebut sangat perlu untuk dipahami bagi anda yang ingin menguasai bahasa sql dan mahir dalam pembuatan database.

DDL adalah sub perintah dari bahasa *SQL* yang digunakan untuk membangun kerangka sebuah database, dalam hal ini database dan table. Terdapat tiga perintah penting dalam *DDL*, yaitu *CREATE*, *ALTER*, *DROP*. *DML* adalah sub perintah dari bahasa *SQL* yang digunakan untuk memanipulasi data dalam database yang telah dibuat. Terdapat empat perintah penting dalam *DML*, yaitu *INSERT*, *SELECT*, *UPDATE*, dan *DELETE*. *DCL* adalah sub bahasa *SQL* yang berfungsi untuk melakukan pengontrolan data dan server databasenya, seperti manipulasi user dan hak akses (priviledges). Yang termasuk perintah dalam *DCL* ada dua, yaitu *GRANT* dan *REVOKE*.

3.7 *MySQL*

MySQL dikembangkan oleh perusahaan Swedia bernama *MySQL* AB yang pada saat ini bernama Tcx Data Konsult AB sekitar tahun 1994-1995, namun cikal bakal kodenya sudah ada sejak tahun 1979. Awalnya Tcx merupakan perusahaan pengembang *software* dan konsultan *database*, dan saat ini *MySQL* sudah diambil alih oleh *Oracle Corp*. Kepopuleran *MySQL* antara lain karena *MySQL* menggunakan *SQL* sebagai bahasa dasar untuk mengakses databasenya sehingga mudah untuk digunakan, kinerja *query* cepat, dan mencukupi untuk kebutuhan *database* perusahaan-perusahaan yang berskala kecil sampai menengah, *MySQL* juga bersifat *open source* (tidak berbayar). Keunggulan dari *MySQL* (Riyanto, 2014) :

1. Portabilitas. *MySQL* dapat berjalan stabil pada berbagai sistem operasi seperti *Windows*, *Linux*, *FreeBSD*, *Mac Os X Server*, *Solaris*, *Amiga*, dan masih banyak lagi.
2. Perangkat lunak sumber terbuka. *MySQL* didistribusikan sebagai perangkat lunak sumber terbuka, dibawah lisensi *GPL* sehingga dapat digunakan secara gratis.
3. *Multi-user*. *MySQL* dapat digunakan oleh beberapa pengguna dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.
4. *Performance tuning*, *MySQL* memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak *SQL* per satuan waktu.

5. Ragam tipe data. *MySQL* memiliki ragam tipe data yang sangat kaya, seperti *signed / unsigned integer, float, double, char, text, date, timestamp*, dan lain-lain.

Sedangkan kekurangan dari penggunaan *MySQL* (Riyanto, 2014) :

1. Untuk koneksi ke bahasa pemrograman *visual* seperti *VB(Visual Basic)*, *delphi*, dan *foxpro*, *MySQL* kurang support, karena koneksi ini menyebabkan *field* yang dibaca harus sesuai dengan koneksi dari program *visual* tersebut, dan ini yang menyebabkan *MySQL* jarang dipakai dalam program *visual*.
2. Data yang ditangani belum begitu besar. Tidak cocok baik menyimpan data maupun untuk memproses data. Program hanya dapat *running* di *windows*. Itupun untuk *windows* lama, untuk *windows* terbaru seperti *Vista, Visual Basic* sangat tidak stabil berjalan di *platform* tersebut, banyak kantor yang dapat menghubungkan laptop pegawai mereka untuk bergabung dalam sistem jaringan kantor.

3.8 **Hypertext Markup Language(HTML)**

Abdul Kadir (2014 :3) *HTML (Hypertext Markup Language)* adalah platform independen, yang berarti dapat membuat, atau kode, file HTML pada satu jenis komputer dan kemudian menggunakan browser pada jenis lain dari komputer untuk melihat bahwa file sebagai halaman web. Dokumen HTML disusun oleh elemen – elemen. Elemen merupakan istilah dari komponen-komponen dasar pembentuk dokumen HTML. Untuk menandai berbagai elemen dalam suatu dokumen HTML bisa menggunakan tag. Tag HTML terdiri atas sebuah kurung sudut kiri (<. Tanda lebih kecil), sebuah nama tag, dan sebuah kurung sudut kanan (>, Tanda lebih besar). Tag pada umumnya berpasangan (misal <H1> dengan </H1>, teks yang berpasangan selalu diawali dengan karakter garis miring). Tag – tag pertama menunjukkan tag awal yang berarti awal elemen, dan yang kedua menunjukkan tag akhir, berarti elemen akhir).

Elemen yang dibutuhkan untuk membuat suatu dokumen HTML dinyatakan dengan tag <html>, <head>, dan <body> berikut tag – tag pasangannya. Setiap dokumen terdiri atas tag head dan body. Elemen head berisi

informasi tentang dokumen – dokumen tersebut, dan elemen body, berisi tentang teks yang sebenarnya yang tersusun dari link, grafik, paragraph, dan elemen lainnya.

3.9 XAMPP

Riyanto (2014 : 12) Xampp merupakan paket PHP dan MySQL berbasis open source, yang dapat digunakan sebagai tool pembantu pengembangan aplikasi berbasis PHP. Xampp mengkombinasikan beberapa paket perangkat lunak berbeda kedalam satu paket. Memanfaatkan xampp sebagai database karena xampp menyediakan aplikasi database MySQL dengan interface lebih mudah dalam pengoperasiannya, tool – tool yang disediakan cukup lengkap dan memenuhi kebutuhan perancangan database selain xampp merupakan aplikasi gratis.