

BAB 2

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi media informasi Perguruan Tinggi menggunakan *ChatBot Line* di Kota Bandung.

2.2 Perguruan Tinggi

Pendidikan tinggi merupakan jenjang pendidikan setelah pendidikan menengah yang mencakup program pendidikan diploma, sarjana, magister, spesialis, dan doktor yang diselenggarakan oleh pendidikan tinggi. Pendidikan tinggi diselenggarakan dengan sistem terbuka. Perguruan Tinggi adalah satuan pendidikan yang menyelenggarakan pendidikan tinggi dan dapat berbentuk akademi, politeknik, sekolah tinggi, institut, atau universitas. [7]

2.3 Sistem Pakar

Sistem pakar adalah suatu program yang bertindak sebagai penasihat atau konsultan pintar dengan mengambil pengetahuan yang disimpan dalam domain tertentu. Seorang pemakai yang belum berpengalaman dalam mendiagnosa suatu masalah dapat memecahkan masalah yang sulit dan mengambil keputusan yang benar.

Secara umum sistem pakar adalah sistem yang berusaha mengadopsi pengetahuan manusia ke komputer, agar komputer dapat menyelesaikan masalah seperti yang biasa dilakukan para ahli. Sistem pakar tidak untuk menggantikan kedudukan seorang pakar tetapi untuk memberikan pengetahuan dan pengalaman pakar tersebut. Sistem pakar dikembangkan pertama kali oleh komunitas AI tahun 1960an. Sistem pakar yang pertama adalah *General Purpose Problem Solver (GPS)* yang dikembangkan oleh Newel Simon. [8]

2.3.1 Konsep Dasar Sistem Pakar

Konsep dasar sistem pakar mengandung sebagai berikut :

1. Keahlian

Keahlian adalah suatu kelebihan penguasaan pengetahuan dibidang tertentu yang diperoleh dari pelatihan, membaca atau dari pengalaman. Bentuk pengetahuan yang termasuk keahlian:

- a) Fakta-fakta pada lingkup permasalahan tertentu.
- b) Teori-teori pada lingkup permasalahan tertentu.
- c) Aturan-aturan berkenaan dengan lingkup permasalahan tertentu.
- d) *Meta-knowledge* (pengetahuan tentang pengetahuan).

2. Ahli/Pakar

Seorang ahli adalah seseorang yang mampu menjelaskan suatu tanggapan, mempelajari hal-hal baru seputar topik permasalahan, menyusun kembali pengetahuan jika dipandang perlu, memecahkan masalah dengan cepat dan tepat.

3. Pengalihan Keahlian

Tujuan dari sistem pakar adalah untuk mentransfer keahlian dari seorang pakar ke dalam komputer kemudian ke masyarakat. Proses ini meliputi 4 kegiatan. Yaitu perolehan pengetahuan (dari para ahli atau sumber-sumber lainnya). representasi pengetahuan ke komputer, kesimpulan dari pengetahuan dan pengalihan pengetahuan ke pengguna.

4. Mengambil Keputusan

Hal yang unik dari sistem pakar adalah kemampuan untuk menjelaskan dimana keahlian tersimpan dalam basis pengetahuan. Kemampuan komputer untuk mengambil kesimpulan dilakukan oleh komponen yang dikenal dengan mesin inferensi yaitu meliputi prosedur tentang pemecahan masalah.

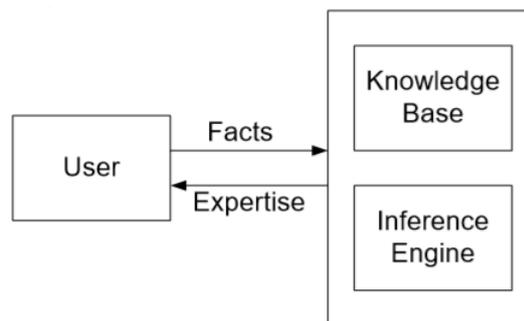
5. Aturan (*Rule*)

Sistem pakar dibuat merupakan sistem yang berdasarkan pada aturan-aturan dimana program disimpan dalam bentuk aturan-aturan sebagai

prosedur pemecahan masalah. Aturan tersebut biasanya berbentuk IF-THEN.

6. Kemampuan menjelaskan

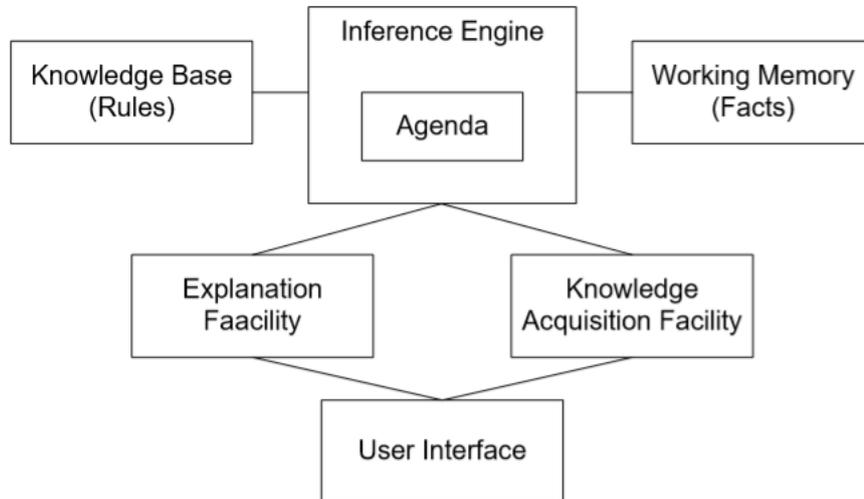
Keunikan lain dari sistem pakar adalah kemampuan dalam menjelaskan atau memberi saran rekomendasi serta juga menjelaskan mengapa beberapa tindakan saran tidak direkomendasikan.



Gambar 2.1 Konsep Dasar Fungsi Sistem Pakar

Gambar 2.1 menggambarkan konsep dasar suatu sistem pakar *knowledge base*. Pengguna menyampaikan fakta atau informasi untuk sistem pakar dan kemudian menerima saran dari pakar atau jawaban ahlinya. Bagian dalam sistem pakar terdiri dari 2 komponen utama, yaitu *knowledge based* yang berisi pengetahuan kepakaran dan *inference engine* yang menggambarkan kesimpulan-kesimpulan tersebut merupakan respon dari sistem pakar atas permintaan pengguna.

2.3.2 Struktur Sistem Pakar



Gambar 2.2 Struktur Sistem Pakar

Berikut adalah penjelasan mengenai struktur sistem pakar yang terdapat pada Gambar 2.2 :

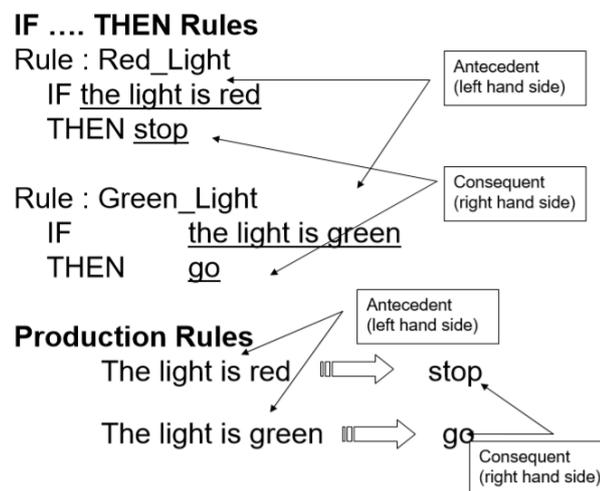
- a) *User Interface* (antarmuka) : Mekanisme komunikasi antara user dan Sistem Pakar.
- b) *Explanation Facility* (subsistem penjelasan) : digunakan untuk melacak respon dan memberikan penjelasan tentang kelakuan sistem pakar secara interaktif.
- c) *Working Memory* : database global dari fakta yang digunakan dalam prosedur.
- d) *Agenda* : Daftar prioritas prosedur yang dibuat oleh motor inferensi dan direkam dalam *working memory*
- e) *Inference engine* (motor inferensi) : program yang berisi metodologi yang digunakan untuk melakukan penalaran terhadap informasi-informasi dalam basis pengetahuan untuk memformulasikan konklusi.
- f) *Knowledge acquisition facility* : berisi pengetahuan pengetahuan yang dibutuhkan untuk memahami, memformulasikan dan menyelesaikan masalah.

2.3.3 Basis Aturan Sistem Pakar

Adapun basis aturan dalam Sistem Pakar adalah sebagai berikut :

1. Pengetahuan dalam Sistem Pakar direpresentasikan dalam bentuk IF - THEN atau dalam bentuk *Production Rules*.
2. Motor inferensi menentukan aturan awal (*rule antecedents*) yang sesuai. Bagian sisi kiri harus cocok dengan fakta yang ada di memori kerja
3. Aturan yang sesuai ditempatkan di agenda dan dapat diaktivasi
 - Aktivasi aturan akan membangkitkan fakta baru di sisi kanan
 - Aktivasi dari satu aturan adalah bagian dari aktivasi aturan yang lain.

Contoh Basis Aturan Sistem Pakar dapat dilihat pada Gambar 2.3



Gambar 2.3 Basis Aturan Sistem Pakar

2.3.4 Keuntungan Sistem Pakar

Keuntungan Sistem Pakar adalah sebagai berikut :

1. Memungkinkan orang awam bisa mengerjakan pekerjaan para ahli.
2. Bisa melakukan proses secara berulang secara otomatis.
3. Menyimpan pengetahuan dan keahlian para pakar.
4. Meningkatkan output dan produktivitas.
5. Meningkatkan kualitas.
6. Mampu mengambil dan melestarikan keahlian para pakar.
7. Mampu beroperasi dalam lingkungan berbahaya.

8. Memiliki kemampuan untuk mengakses pengetahuan.
9. Memiliki realibilitas.
10. Meningkatkan kapabilitas system computer.
11. Memiliki kemampuan untuk bekerja dengan informasi yang tidak lengkap dan mengandung ketidakpastian.
12. Sebagai media pelengkap dalam pelatihan.
13. Meningkatkan kapabilitas dalam penyelesaian masala.
14. Menghemat waktu dalam pengambilan keputusan.

2.3.5 Kelemahan Sistem Pakar

Adapun kelemahan dalam Sistem Pakar adalah sebagai berikut :

1. Biaya yang diperlukan untuk membuat dan memeliharanya sangat mahal.
2. Sulit dikembangkan. Hal ini erat kaitannya dengan ketersediaan pakar dalam bidangnya.
3. System pakar tidak 100% bernilai benar.

2.3.6 Penarikan Kesimpulan (Inferencing)

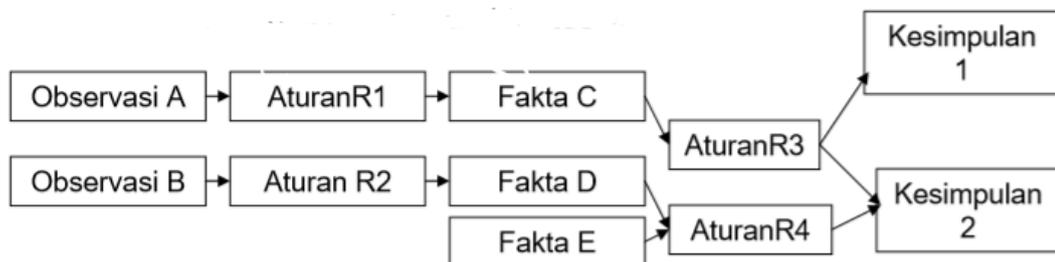
Kemampuan berfikir (*reasoning*) termasuk didalamnya proses penarikan kesimpulan (*inferencing*) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristic atau pencarian lainnya.

2.3.7 Metode Algoritma *Forward Chaining*

Dalam sistem pakar, metode *forward chaining* merupakan metode yang melakukan pelacakan ke depan, dimulai dari sekumpulan fakta dan berakhir di kesimpulan. Metode *forward chaining* bermula dari fakta-fakta yang sudah diketahui atau ditetapkan dalam suatu sistem pakar [8]. Kemudian menggunakan premis yang ditentukan oleh user, yang nantinya premis-premis itu akan disesuaikan dengan fakta-fakta tadi menggunakan suatu aturan tertentu. Hasil dari proses ini akan menghasilkan fakta baru, yang nantinya akan digunakan untuk

melanjutkan proses dan mendapatkan kesimpulan akhir setelah tidak ada lagi aturan yang premisnya cocok dengan fakta.

Alasan mengapa menggunakan *forward chaining* adalah karena terdapat banyak cara atau aturan yang berbeda untuk mendapatkan kesimpulan yang sedikit, dan ingin mendapatkan kesimpulan dari fakta-fakta yang sudah ada sebelumnya. Aturan *forward chaining* dapat dilihat pada Gambar 2.4 :



Gambar 2.4 Aturan Forward Chaining

Pencocokkan fakta atau pernyataan dimulai dari bagian sebelah kiri (IF dulu). Dengan kata lain, penalaran dimulai dari fakta terlebih dahulu untuk menguji kebenaran hipotesis.

Contoh :

Pada tabel di bawah ini terlihat 10 aturan yang tersimpan dalam basis pengetahuan. Fakta awal yang diberikan hanya A & F (artinya A dan F bernilai benar). Ingin dibuktikan apakah K bernilai benar (hipotesis : K) ?

R1 : if A and B then C

R2 : if C then D

R3 : if A and E then F

R4 : if A then G

R5 : if F and G then D

R6 : if G and E then H

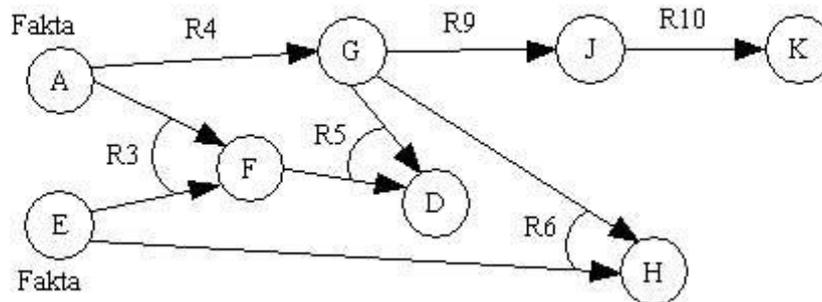
R7 : if C and H then I

R8 : if I and A then J

R9 : if G then J

R10 : if J then K

Fakta awal yang diberikan hanya A dan E, ingin membuktikan apakah K bernilai benar. Proses penalaran *forward chaining* terlihat pada Gambar 2.5 :



Gambar 2.5 Forward chaining

Berikut adalah hasil dari contoh metode *forward chaining* . Dapat dilihat pada Tabel 2.1 sebagai berikut :

Tabel 2.1 Hasil Metode *Forward chaining*

Aturan	Fakta Baru
R-3	F
R-4	G
R-5	D
R-6	H
R-9	J
R-10	K (Terbukti)

2.4 ChatBot

ChatBot adalah sebuah program komputer yang dirancang untuk mensimulasikan sebuah percakapan atau komunikasi yang interaktif kepada user (manusia) melalui bentuk teks, suara, dan atau visual. Percakapan yang terjadi

antara komputer dengan manusia merupakan bentuk respon dari program yang telah dideklarasikan pada database program pada komputer. Respon yang dihasilkan merupakan hasil pemindaian kata kunci pada inputan user dan menghasilkan respon balasan yang dianggap paling cocok, atau pola kata-kata yang dianggap paling mendekati, dari database tentunya. Dalam bahasa sehari-harinya *ChatBot* merupakan aplikasi atau program komputer yang dirancang untuk meniru manusia itu sendiri, batasan yang diambil dari *ChatBot* adalah mampu meniru komunikasi manusia. Jadi jika manusia sedang bercakap-cakap dengan program ini, seakan-akan ada 2 pribadi manusia yang saling berkomunikasi. Nyatanya tidak, manusia berkomunikasi dengan *Bot*. *Bot* sudah dirancang untuk merespon segala jenis pertanyaan dan pernyataan yang diinputkan oleh manusia (user). Hal ini terjadi karena sebelumnya sudah dideklarasikan pada database, berupa entitas-entitas kata, pola kalimat, dan berbagai jenis pernyataan dan pertanyaan.

ChatBot adalah sebuah simulator percakapan yang berupa program komputer yang dapat berdialog dengan penggunanya dalam bahasa alami. Karena *ChatBot* hanya sebuah program, dan bukan robot (*ChatBot* tidak memiliki tubuh dan tidak memiliki mulut sehingga tidak dapat berbicara seperti manusia), maka yang dimaksud dengan dialog antar manusia sebagai pengguna dengan *ChatBot* dilakukan dengan cara mengetik apa yang akan dibicarakan dan *ChatBot* akan memberikan respon. Orang yang membuat dan mengembangkan program *ChatBot* disebut *bot* master.

Kata *bot* diambil dari kata *robot*, yang dapat diartikan sebagai sebuah program komputer yang berfungsi mengumpulkan informasi atau memberikan layanan yang biasanya telah dijadwalkan sebelumnya. Sebuah *bot* (atau bisa juga dikatakan sebagai sebuah agent) biasanya mencari informasi di internet mengumpulkan informasi yang sesuai dengan yang kita inginkan, dan memberikannya pada kita secara periodik atau sesuai dengan perintah kita.

Bot biasa digunakan dalam IRC (*Internet Chat Groups*) di mana *bot* tersebut melakukan berbagai fungsi seperti menyapa partisipan baru dan memonitor penggunaan kata-kata yang tidak pantas. Sekarang ini, *bot* dapat mengumpulkan

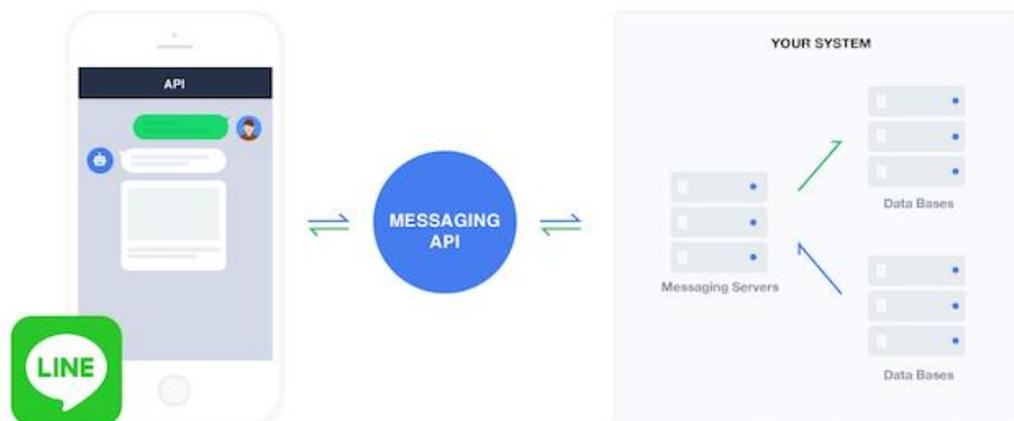
informasi, mengolah informasi tersebut, dan memberikan respon terhadap pengguna dan event yang terjadi di internet.

2.4.1 Cara Kerja ChatBot

Pada dasarnya *bot* bekerja dengan cara melihat kata kunci dalam data yang masuk dan membalasnya dengan kata kunci yang paling cocok, atau pola kata-kata yang paling mirip dari basis data tekstual. Artinya, jika pengguna mengirim suatu permintaan maka *bot* akan membalasnya dengan respon yang spesifik sesuai dengan kata kunci yang dikirim. Sebagai contoh, misalnya pengguna mengirim atau mengetikkan suatu pertanyaan “Carikan saya informasi kampus Negeri” maka dengan informasi yang sudah tersedia, *bot* akan segera merespon sesuai dengan pertanyaan tersebut “Carikan saya informasi kampus Negeri”. Disini anda hanya perlu memberikan perintah atau melakukan sebuah percakapan baik itu dalam bentuk teks maupun menu, dan *bot* akan mencari data yang anda butuhkan sesuai dengan kata kunci yang anda berikan.

2.4.2 LINE Messaging API

API (*Application Programming Interface*) merupakan kumpulan dari fungsi-fungsi yang digunakan oleh *programmer* untuk membangun sebuah aplikasi. Pada *line ChatBot* ini, api yang digunakan yaitu *Messaging API* pada *line* yang berfungsi untuk saling mengirimkan data antar aplikasi *line* dengan sistem yang digunakan yaitu web server melalui server *line* [9]



Gambar 2.6 Cara Kerja Messaging API

2.4.3 Fungsi Strpos

Fungsi `strpos()` adalah fungsi bawaan PHP yang bisa digunakan untuk mencari posisi sebuah karakter atau sebuah string di dalam string lainnya. Hasil akhir fungsi ini adalah angka yang menunjukkan posisi karakter/string yang ingin dicari.

2.4.4 Metode Jaro Winkler Distance

Jaro Winkler distance merupakan varian dari *Jaro distance metrik* yaitu sebuah algoritma untuk mengukur kesamaan antara dua string, biasanya algoritma ini digunakan di dalam pendeteksian duplikat. Semakin tinggi *Jaro-Winkler distance* untuk dua string, semakin mirip dengan string tersebut. *Jaro-Winkler distance* terbaik dan cocok untuk digunakan dalam perbandingan string singkat seperti nama orang. [10]

Rumus :

$$d_j = \frac{1}{3} \times \left(\frac{m}{S_1} + \frac{m}{S_2} + \frac{m-t}{m} \right)$$

Dimana :

m = Jumlah karakter yang sama persis
 $|S_1|$ = Panjang string 1
 $|S_2|$ = Panjang string 2
 t = Jumlah transposisi

Menghitung *Jaro-Winkler Distance*

$$d_w = d_j + (l \times p (1 - d_j))$$

Dimana :

d_j = *Jaro distance* untuk string S_1 dan string S_2

l = Panjang *prefix* (panjang karakter yang sama sebelum ditemukan ketidaksamaan) nilai maksimum 4 karakter

p = Konstanta *scaling factor* (Nilai standar untuk konstanta ini menurut *Winkler* adalah $p = 0.1$)

2.5 Web Server

Server web atau peladen web dapat merujuk baik pada perangkat keras ataupun perangkat lunak yang menyediakan layanan akses kepada pengguna

melalui protokol komunikasi HTTP atau HTTPS atas berkas-berkas yang terdapat pada suatu situs web dalam layanan ke pengguna dengan menggunakan aplikasi tertentu seperti peramban web.

2.5.1 Hosting

Hosting adalah layanan berbasis internet yang menyediakan sumber daya atau *resource* untuk disewakan sebagai tempat menyimpan data atau tempat menjalankan aplikasi atau website ditempat terpusat yang disebut dengan *server* sehingga memungkinkan organisasi atau individu menempatkan informasi di internet berupa *HTTP*, *FTP*, *EMAIL*, atau *DNS*. Dari pengertian hosting diatas dapat disimpulkan bahwa hosting adalah ibarat sebuah rumah yang disewakan dengan ukuran tertentu dan setiap bangunan rumah pasti memiliki ukuran yang berbeda. Masing-masing penyedia jasa web host menawarkan jenis hosting yang berbeda agar sesuai dengan keinginan dan kebutuhan klien. Berikut contoh hosting yang sering digunakan:

1. Shared Hosting

Shared hosting merupakan tipe web hosting yang paling banyak digunakan. Paket ini menjadi pilihan yang pas untuk blog dan website sederhana. Dengan shared hosting, dapat berbagi server dengan klien lainnya di provider hosting yang sama. Website yang *dionlinekan* di server yang sama berbagi semua *resource*, seperti *memori*, *computing power*, *disk space*, dan lain-lain.

Kelebihan:

- a. Murah
- b. Mudah digunakan oleh pemula
- c. Pre-configured server
- d. Control panel yang mudah digunakan
- e. Maintenance dan administrasi server dilakukan oleh penyedia hosting

Kekurangan:

- a. Sedikit atau sama sekali tidak dapat mengontrol konfigurasi server
- b. Apabila trafik tinggi di website pengguna lain, maka hal tersebut akan memengaruhi kinerja situs

2. VPS (*Virtual Private Server*) Hosting

Dengan menggunakan VPS (*Virtual Private Server*) hosting, masih berbagi server dengan pengguna lainnya. Akan tetapi, penyedia web hosting telah mengalokasikan beberapa bagian sendiri di server. Hal ini berarti akan mendapatkan space server yang dedicated dan memperoleh sejumlah power computing dan memory. Selain itu, VPS hosting merupakan pilihan yang tepat bagi pengguna website bisnis yang traffiknya tinggi.

Kelebihan:

- a. Space server yang dedicated (tanpa harus membeli server dedicated)
- b. Peningkatan traffic di website lain tidak berdampak bagi performa situs
- c. Akses root ke server
- d. Scalability yang mudah
- e. Dapat dikelola sendiri

Kekurangan:

- a. Sangat mahal jika dibandingkan dengan shared hosting
- b. Harus menguasai informasi dan pengetahuan teknis terkait manajemen server

3. *Cloud Hosting*

Saat ini, cloud hosting merupakan paket yang paling banyak disukai di pasaran karena literally tanpa *downtime*. Dengan cloud hosting, dapat menggunakan sekelompok server. File dan resource direplikasi di setiap server. Ketika salah satu server cloud sibuk atau bermasalah, maka traffic situs akan secara otomatis di-route ke server lainnya dalam cluster.

Kelebihan:

- a. Sedikit bahkan tidak ada downtime
- b. Jika server error, sama sekali tidak akan memengaruhi situs
- c. Alokasi resource sesuai permintaan
- d. Bayar sesuai dengan yang digunakan
- e. Lebih scalable daripada VPS

Kekurangan:

- a. Biaya sulit diperkirakan
- b. Akses root tidak selalu tersedia

4. WordPress Hosting

WordPress hosting merupakan bentuk lain dari *shared hosting* dan dibuat untuk pengguna website *WordPress*. *Server* dikonfigurasi secara khusus untuk *WordPress* dan situs dilengkapi dengan *pre-installed plugin* terhadap hal-hal penting, seperti *caching* dan keamanan. Karena konfigurasi sudah sangat optimal, maka situs akan loading lebih cepat dan terhindar dari masalah. Paket *WordPress hosting* menyertakan fitur tambahan terkait *WordPress*, tema *WordPress* yang *pre-designed*, *drag-and-drop page builder*, dan *tool developer* yang spesifik.

Kelebihan:

- a. Murah (umumnya memiliki kesamaan harga dengan *shared hosting*)
- b. Mudah digunakan oleh pemula
- c. *One-click WordPress installation*
- d. Performa yang baik untuk situs *WordPress*
- e. *Customer support* menguasai hal-hal yang berhubungan dengan *WordPress*.
- f. Terdapat plugin dan tema *WordPress* yang *pre-installed*.

Kekurangan:

Hanya bisa digunakan oleh website yang dibuat di *WordPress* (masalah bisa muncul jika ingin mengon*line*kan banyak website di server, tetapi beberapa tidak dibuat menggunakan WordPress).

5. Dedicated Server Hosting

Dengan adanya *dedicated hosting*, bisa memiliki server fisik sendiri yang secara khusus didedikasikan hanya untuk situs. Karena itulah, penggunaan *dedicated hosting* sangatlah fleksibel. Dapat mengonfigurasi sistem operasi dan software yang ingin digunakan, serta setup keseluruhan aspek hosting sesuai dengan kebutuhan.

Menyewa *dedicated server* sama bagusnya dengan memiliki server sendiri, tetapi di *dedicated server*, dibantu oleh customer support yang andal.

Kelebihan:

- a. Kontrol penuh terhadap konfigurasi server
- b. Keandalan yang tinggi (tidak perlu berbagi resource server dengan pengguna lain)
- c. Akses root ke server
- d. Keamanan terjamin

Kekurangan:

- a. Mahal
- b. Harus menguasai pengetahuan teknis terkait manajemen server

2.5.2 Domain

Domain adalah nama yang diberikan untuk mengidentifikasi sebuah jaringan tanpa menggunakan internet protokol (IP), mengingat biasanya IP menggunakan angka-angka yang sulit dihafal. Dalam dunia internet domain merupakan aspek penting dalam sebuah *website*. Fungsinya adalah sebagai alamat untuk memudahkan para pengunjung untuk mencari informasi yang diinginkan. Perkembangan domain saat ini sangat meningkat dari tahun ke tahun. Pada

awalnya, sejarah domain mulai digunakan pada tahun 1985 oleh perusahaan komputer asal Massachusetts, Amerika Serikat bernama Symbolics.com. Di akhir tahun 1985 baru ada enam domain yang terdaftar. Namun, sekarang ada ribuan bahkan jutaan domain yang ada di seluruh dunia.

Membuat website ada baiknya untuk mengenal lebih dalam apa itu domain dan seluk beluknya. Mulai dari jenis-jenis yang paling banyak digunakan, bagaimana memilih nama yang efektif hingga memilih jasa penyedia hosting untuk mendapatkan domain tersebut. Sebagai satu rangkaian dalam *website*, domain tidak berdiri sendiri. Setidaknya ada tiga jenis domain yang paling sering digunakan, yaitu *top level domain*, *second level domain*, dan *third level domain*.

1. *Top Level Domain*

Top level domain adalah bagian dari nama domain itu sendiri yang berada di bagian paling kanan setelah titik. Contohnya seperti *www.google.com*, dari contoh tersebut yang dimaksud TLD-nya adalah *.com*. TLD sendiri terbagi dua jenis yaitu *Global Top Level Domain* dan *Country Code Level Domain*. Untuk kategori *Global Top Level Domain* biasanya menggunakan domain seperti *.com*, *.net*, *.org*, *.edu* dan lain sebagainya. Domain *.com* digunakan untuk website komersial, sedangkan *.net* digunakan untuk website jaringan, dan *.edu* untuk edukasi. Ada pula domain lain seperti *.org* untuk organisasi, *.mil* untuk situs angkatan bersenjata dan lain sebagainya.

Sementara itu, untuk *Country Code Level Domain* lebih sering digunakan untuk domain lokal di negara-negara tertentu. Semisal saja *.id* untuk domain Indonesia, *.cn* untuk negara Cina dan *.my* untuk Malaysia. Masing-masing negara tidaklah sama dan hanya diambil singkatannya saja sehingga lebih mudah untuk diingat.

2. *Second Level Domain*

Jenis domain selanjutnya adalah *Second Level Domain* atau biasa disebut dengan SLD. SLD ini merujuk pada sebuah nama yang digunakan. Bila mencontoh pada domain pada poin di atas, maka untuk SLD-nya adalah 'google'. Di sini penting untuk membuat SLD yang mudah diingat atau

diketik maupun dieja oleh pengunjung. Maka dari itu, harus menggunakan nama yang familiar dan tidak terlalu panjang. Domain SLD ini bisa dibuat dengan cara mendaftar dan membelinya. Jangan lupa, lakukan pembaruan secara berkala paling tidak selama satu tahun.

3. *Third Level Domain*

Secara umum *Third Level Domain* atau TLD ini adalah nama domain yang berikan sebelum TLD dan SLD. Contohnya saja bila membuat untuk keperluan email, bisa dengan menambahkan `webmail.namadomain.com`. Begitu pula untuk hal-hal lainnya seperti blog, promo, dan sebagainya. Dalam dunia internet, TLD ini juga sering disebut dengan subdomain. Nama tersebut bisa dibuat sesuai kebutuhan setelah memiliki SLD.

2.5.3 cPanel

cPanel adalah salah satu control panel berbasis Linux yang paling banyak digunakan di akun web hosting. Melalui panel ini, bisa dengan mudah mengelola semua layanan dalam satu tempat. Saat ini, cPanel merupakan panel standar yang paling banyak digunakan dan sebagian besar web developer sudah sangat familiar dengan tool ini. Selain intuitif dan mudah digunakan, cPanel juga memungkinkan untuk mengelola akun web hosting dengan maksimal. Membuat FTP user dan alamat email baru, memonitor resource, membuat subdomain, dan menginstall software merupakan beberapa kegunaan cPanel. Berikut kelebihan cPanel dan kekurangannya.

Kelebihan cPanel:

- a. Mudah dipelajari
- b. Mudah digunakan
- c. Menghemat waktu dan uang
- d. Telah melalui uji coba
- e. Menawarkan software auto installer
- f. Tersedia berbagai tutorial dan panduan *online*

Kekurangan cPanel:

- a. Fitur yang terlalu banyak hanya akan memperlambat performa cPanel
- b. Pengaturan yang penting bisa tiba-tiba berubah secara otomatis
- c. Beberapa hosting menjalankan software yang tidak diupgrade
- d. Biayanya mahal dan jarang dipasangkan di hosting gratis

2.6 Analisis Berorientasi Objek

Analisis dan Desain Berorientasi Objek (*Object Oriented Analysis and Design*) adalah cara baru dalam memikirkan suatu masalah dengan menggunakan model yang dibuat menurut konsep. Dasar pembuatannya sendiri adalah objek yang merupakan kombinasi antara struktur data dan perilaku dalam satu entitas. Alasan mengapa harus memakai metode berorientasi objek yaitu karena perangkat lunak itu sendiri yang bersifat dinamis, di mana hal ini disebabkan karena kebutuhan pengguna berubah dengan cepat. Selain itu bertujuan untuk menghilangkan kompleksitas transisi antar tahap pada pengembangan perangkat lunak, karena pada pendekatan berorientasi objek, notasi yang digunakan pada tahap analisis perancangan dan implementasi relatif sama tidak seperti pendekatan konvensional yang dikarenakan notasi yang digunakan pada tahap analisisnya berbeda-beda. Hal itu menyebabkan transisi antar tahap pengembangan menjadi kompleks. Di samping itu dengan pendekatan berorientasi objek membawa pengguna kepada abstraksi atau istilah yang lebih dekat dengan dunia nyata, karena di dunia nyata itu sendiri yang sering pengguna lihat adalah objeknya bukan fungsinya. Beda ceritanya dengan pendekatan terstruktur yang hanya mendukung abstraksi pada level fungsional. Adapun dalam pemrograman berorientasi objek menekankan berbagai konsep seperti: *Class, Object, Abstract, Encapsulation, Polymorphism, Inheritance* dan tentunya UML (*Unified Modeling Language*). UML (*Unified Modeling Language*) sendiri merupakan salah satu alat bantu yang dapat digunakan dalam Bahasa pemrograman berorientasi objek. Selain itu UML merupakan *standard modeling language* yang terdiri dari kumpulan-kumpulan diagram, dikembangkan untuk membantu para pengembang sistem dan software agar bias menyelesaikan tugas-tugas seperti: Spesifikasi, Visualisasi, Desain Arsitektur,

Konstruksi, Simulasi dan Testing. Dapat disimpulkan bahwa UML (*Unified Modeling Language*) adalah sebuah Bahasa yang berdasarkan grafik atau gambar untuk memvisualisasikan, menspesifikasikan, membangun dan pendokumentasian dari sebuah sistem pengembangan perangkat lunak berbasis objek (*Object Oriented Programming*) [10].

Dokumentasi UML menyediakan 10 macam diagram untuk memodelkan aplikasi berorientasi objek yang 4 diantaranya adalah [11]:

1. *Activity Diagram*

Activity Diagram adalah sebuah tahapan yang lebih focus kepada menggambarkan proses bisnis dan urutan aktivitas dalam sebuah proses. Di mana biasanya dipakai pada *business modeling* untuk memperlihatkan urutan aktifitas proses bisnis. *Activity diagram* ini sendiri memiliki struktur yang mirip dengan *flowchart* atau data flow diagram pada perancangan terstruktur. *Activity diagram* dibuat berdasarkan sebuah atau beberapa *Use Case* pada *Use Case diagram* [12].

2. *Use Case Diagram*

Use Case diagram menggambarkan fungsionalitas yang diharapkan dari sebuah sistem. Di dalam *Use Case diagram* ini sendiri lebih ditekankan kepada apa yang diperbuat sistem dan bagaimana sebuah sistem itu bekerja. Sebuah *Use Case* merepresentasikan sebuah interaksi antara actor dengan sistem. *Use Case* merupakan bentuk dari sebuah pekerjaan tertentu, misalnya *login* ke dalam sistem, *posting* dan sebagainya, sedangkan seorang actor adalah sebuah entitas manusia atau mesin yang berinteraksi dengan sistem untuk melakukan pekerjaan-pekerjaan tertentu [12]. Adapun komponen-komponen dalam *Use Case diagram* diantaranya:

a. Actor

Actor merupakan suatu entitas yang berkaitan dengan sistem tapi bukan dari bagian dalam sistem itu sendiri. *Actor* berada diluar sistem namun berkaitan erat dengan fungsionalitas di dalamnya. *Actor* memiliki hubungan secara langsung terhadap fungsi utama baik terhadap salah satu atau semua

fungsionalitas utama. *Actor* juga dapat dibagi terhadap berbagai jenis atau tingkatan dengan cara digeneralisasi atau dispesifikasi tergantung kebutuhan sistemnya. *Actor* biasanya dapat berupa pengguna atau *database* yang secara pandang berada dalam suatu ruang lingkup sistem.

b. Use Case

Use Case merupakan gambaran umum dari fungsi atau proses utama yang menggambarkan tentang salah satu perilaku sistem. Perilaku sistem ini terdefinisi dari proses bisnis sistem yang akan dimodelkan. Tidak semua proses bisnis digambarkan secara fungsional pada *Use Case*, tetapi yang digambarkan hanya fungsionalitas utama yang berkaitan dengan sistem. *Use Case* menitik-beratkan bagaimana suatu sistem dapat berinteraksi baik antar sistem maupun di luar sistem.

3. *Sequence Diagram*

Sequence diagram digunakan untuk menggambarkan perilaku pada sebuah scenario. Diagram jenis ini memberikan kejelasan sejumlah objek dan pesan-pesan yang diletakkan diantaranya di dalam sebuah *Use Case*. Komponen utamanya adalah objek yang digambarkan dengan kotak segi empat atau bulat, *message* yang digambarkan dengan garis putus dan waktu yang ditunjukkan dengan *progress vertical*. Manfaat dari *Sequence* diagram adalah memberikan gambaran detail dari setiap *Use Case diagram* yang dibuat sebelumnya [12].

4. *Class Diagram*

Class diagram adalah sebuah *class* yang menggambarkan struktur dan penjelasan *class*, paket dan objek serta hubungan satu sama lain. *Class diagram* juga menjelaskan hubungan antar *class* secara keseluruhan di dalam sebuah sistem yang sedang dibuat dan bagaimana caranya agar mereka saling berkolaborasi untuk mencapai sebuah tujuan [12].

2.7 Perangkat Lunak Pengembang (Tools)

Perangkat Lunak Pengembang merupakan perangkat lunak yang membantu dalam pembuatan aplikasi sistem yang akan digunakan.

2.7.1 PHP

PHP adalah bahasa scripting yang menyatu dengan HTML dan dijalankan pada *server side*. Artinya semua sintaks yang kita berikan akan sepenuhnya dijalankan pada server sedangkan yang dikirimkan ke browser hanya hasilnya saja. contoh struktur syntax PHP sebagai berikut :

```
<?php  
  
    echo("Hallo apakabar? Nama saya PHP script");
```

2.7.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (bahasa Inggris: database management system) atau DBMS yang multithread, multi-user, dengan sekitar 6 juta instalasi di seluruh dunia. MySQL AB membuat MySQL tersedia sebagai perangkat lunak gratis di bawah lisensi GNU *General Public License* (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus dimana penggunaannya tidak cocok dengan penggunaan GPL.

Dengan menggunakan SQL, proses akses database menjadi lebih user - friendly dibandingkan dengan menggunakan dBASE atau Clipper yang masih menggunakan perintah - perintah pemrograman. MySQL merupakan software database yang paling populer di lingkungan Linux, kepopuleran ini karena ditunjang performa query dari databasenya yang saat ini bisa dikatakan paling cepat dan jarang bermasalah. MySQL ini juga sudah dapat berjalan pada lingkungan Windows.

Perintah untuk mengelola database dibagi menjadi 3 (tiga) kelompok, diantaranya :

1. Perintah untuk mendefinisikan data/DDL (Data Definition Language).

2. Perintah untuk memanipulasi data/DML (Data Manipulation Language).
3. Perintah untuk mengendalikan data/DCL (Data Control Language).

2.7.3 JSON

JSON (JavaScript Object Notation) adalah format pertukaran data yang ringan, mudah dibaca dan ditulis oleh manusia, serta mudah diterjemahkan dan dibuat (generate) oleh komputer. Format ini dibuat berdasarkan bagian dari Bahasa Pemrograman JavaScript, Standar ECMA-262 Edisi ke-3 - Desember 1999. JSON merupakan format teks yang tidak bergantung pada bahasa pemrograman apapun karena menggunakan gaya bahasa yang umum digunakan oleh programmer keluarga C termasuk C, C++, C#, Java, JavaScript, Perl, Python dll. Oleh karena sifat-sifat tersebut, menjadikan JSON ideal sebagai bahasa pertukaran-data.

2.7.4 *LINE Bot Designer*

LINE Bot Designer adalah aplikasi yang memungkinkan kita untuk membuat prototipe sebuah *ChatBot* untuk aplikasi perpesanan *LINE* dengan mudah tanpa memerlukan pengetahuan ngoding sedikitpun. Kita diberikan opsi untuk mendesain *ChatBot* dalam skenario-skenario yang diinginkan. [11]

Namun perlu diingat bahwa produk yang dihasilkan oleh *LINE Bot Designer* hanya bersifat prototipe dan belum bisa dijadikan *ChatBot* sungguhan (yang bisa berinteraksi dengan pengguna lain). Tapi, prototipe ini bisa dimanfaatkan untuk mempercepat proses pembuatan *bot* aslinya atau untuk menunjukkan konsep bisnis yang ingin dilakukan dengan memanfaatkan aplikasi *LINE* dan layanan *ChatBot*.

2.7.5 Basis Data

Basis data terdiri dari dua kata yaitu basis dan data. Basis dapat diartikan sebagai gudang tempat berkumpul. Sedangkan data adalah representasi fakta dunia nyata yang mewakili objek seperti manusia, barang, hewan, peristiwa konsep, yang nyatakan dalam bentuk angka, huruf, simbol teks, gambar, bunyi atau kombinasinya [4].

Basis data dapat didefinisikan dari beberapa sudut pandang diantaranya sebagai berikut :

1. Sekumpulan data *persistence* (data disimpan defile sekunder atau data yang tahan lama) yang saling terkait, menggambarkan suatu organisasi (enterprise).
2. Himpunan kelompok data (arsip) yang saling berhubungan yang diorganisasikan sedemikian rupa agar kelak dapat dimanfaatkan kembali dengan cepat dan mudah.
3. Kumpulan data yang saling berhubungan yang disimpan secara bersama sedemikina rupa dan tanpa pengulangan (redudansi) yang tidak perlu, untuk memenuhi berbagai kebutuhan.
4. Kumpulan file atau arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronik.

2.7.6 LINE

LINE adalah sebuah aplikasi pengirim pesan yang dapat digunakan pada berbagai platform seperti smartphone, tablet, dan komputer. Dan jika pengguna ingin menggunakan *LINE* pastikan smartphone terhubung dengan jaringan internet, sehingga pengguna dapat mengirim pesan teks, mengirim gambar, video, pesan suara, dan lain – lain.

2.7.7 API

API adalah singkatan dari *Application Programming Interface*, dan memungkinkan *developer* untuk mengintegrasikan dua bagian dari aplikasi atau dengan aplikasi yang berbeda secara bersamaan. *API* terdiri dari berbagai elemen seperti *function*, *protocols*, dan *tools* lainnya yang memungkinkan *developer* untuk membuat aplikasi. Tujuan penggunaan *API* adalah untuk mempercepat proses *development* dengan menyediakan *function* secara terpisah sehingga *developer* tidak perlu membuat fitur yang serupa. Penerapan *API* akan sangat terasa jika fitur yang diinginkan sudah sangat kompleks, tentu membutuhkan waktu untuk membuat yang serupa dengannya. Misalnya: integrasi dengan *payment gateway*.

Terdapat berbagai jenis sistem *API* yang dapat digunakan, termasuk sistem operasi, *library*, dan web.

Seperti namanya, Web *API* dalam diakses melalui protokol *HTTP*, ini adalah konsep bukan teknologi. Kita bisa membuat Web *API* dengan menggunakan teknologi yang berbeda seperti *PHP*, *Java*, *NET*, dll.

2.8.8 Webhook

Webhook adalah konsep *API* yang saat ini semakin populer digunakan. Semakin banyak yang dapat dilakukan di web, menjadikan *webhook* makin banyak digunakan. Selain itu, *webhook* sangat berguna dan mudah untuk diterapkan. *Webhook* atau yang biasa disebut *callback* adalah cara bagi suatu aplikasi untuk menyediakan aplikasi lain dengan informasi *real-time*. Lebih mudahnya, *webhook* adalah *link* URL yang ditambahkan agar data yang dikirim dapat langsung diterima di waktu sama dengan *link* URL yang sudah ditentukan.