BAB II

LANDASAN TEORI

2.1 Landasan Teori

Landasan teori merupakan penjelasan berbagai konsep dasar dan teori-teori yang berkaitan dalam pembangunan aplikasi *Natural language Processing* untuk pembuatan *chat bot* berbasis web.

2.1.1 Kecerdasan Buatan

Kecerdasan Buatan adalah salah satu cabang Ilmu pengetahuan berhubungan dengan pemanfaatan mesin untuk memecahkan persoalan yang rumit dengan cara yang lebih manusiawi. Hal Ini biasanya dilakukan dengan mengikutI atau mencontoh karakteristik dan analogi berpikir dari kecerdasan atau Inteligensia manusia, dan menerapkannya sebagai algoritma yang dikenal oleh komputer. Dengan suatu pendekatan yang kurang lebih fleksibel dan efisien dapat diambil tergantung dari keperluan, yang mempengaruhi bagaimana wujud dari perilaku kecerdasan buatan. AI biasanya dihubungkan dengan Ilmu Komputer, akan tetapi juga terkait erat dengan bidang lain seperti Matematika, Psikologi, Pengamatan, Biologi, Filosofi, dan yang lainnya. Kemampuan untuk mengkombinasikan pengetahuan dari semua bidang ini pada akhirnya akan bermanfaat bagi kemajuan dalam upaya menciptakan suatu kecerdasan buatan.

Pengertian lain dari kecerdasan buatan adalah bagian ilmu komputer yang membuat agar mesin komputer dapat melakukan pekerjaan seperti dan sebaik

yang dilakukan manusia. Pada awal diciptakannya, komputer hanya difungsikan sebagai alat hitung saja. Namun seiring dengan perkembangan jaman, maka peran komputer semakin mendominasi kehidupan manusia. Komputer tidak lagi hanya digunakan sebagai alat hitung, lebih dari itu, komputer diharapkan untuk dapat diberdayakan untuk mengerjakan segala sesuatu yang bisa dikerjakan oleh manusia.

Menurut beberapa ahli kecerdasan buatan didefinisikan sebagai berikut :

1. Menurut Rich and knight [1991] [10] : ”*Kecerdasan buatan (artificial intellegence) merupakan sebuah studi tentang bagaimana membuat komputer melakukan hal-hal yang pada saat ini dapat dilakukan lebih baik oleh manusia”.*
2. Menurut Andri Kristanto (2003) : Kecerdasan buatan merupakan bagian dari ilmu pengetahuan komputer yang khusus ditujukan dalam perancangan otomatisasi tingah laku cerdas dalam sistem kecerdasan komputer.

Kecerdasan buatan dilihat dari berbagai sudut pandang adalah sebagai berikut :

1. Sudut pandang Kecerdasan (Intellegence)

   Kecerdasan buatan adalah bagaimana membuat mesin yang ”cerdas” dan dapat melakukan hal-hal yang sebelumnya dapat dilakukan oleh manusia.
2. Sudut pandang Penelitian

   Studi bagaimana membuat agar komputer dapat melakukan sesuatu sebaik yang dilakukan oleh manusia.

2.1.2 Konsep Kecerdasan Buatan

Ada beberapa konsep yang harus dipahami dalam kecerdasan buatan, diantaranya (kusrini, 2006) :

1. *Turing Test* – Metode Pengujian Kecerdasan

   Turing Test merupakan sebuah metode pengujian kecerdasan yang dibuat oleh Alan Turing. Proses uji ini melibatkan seorang penanya (manusia) dan dua obyek yang ditanyai. Yang satu adalah seorang manusia dan yang satunya adalah sebuah mesin yang akan diuji. Penanya tidak dapat melihat langsung kepada obyek yang ditanyai. Penanya diminta untuk membedakan mana jawaban komputer dan mana jawaban manusia berdasarkan jawaban kedua obyek tersebut. Jika penanya tidak dapat membedakan mana jawaban mesin dan mana jawaban manusia maka Turing berpendapat bahwa mesin yang diuji tersebut dapat diasumsikan CERDAS.

2. Pemrosesan Simbolik

   Komputer semula didesain untuk memproses bilangan atau angka-angka (pemrosesan numerik). Sementara manusia dalam berfikir dan menyelesaikan masalah lebih bersifat simbolik, tidak didasarkan pada sejumlah rumus atau melakukan komputasi matematika. Sifat penting dari AI adalah bahwa AI merupakan bagian dari ilmu komputer yang melakukan proses secara simbolik dan non-algoritmik dalam penyelesaian masalah.

3. *Heuristic*

   Istilah heuristic diambil dari bahasa Yunani yang berarti menemukan. Heuristic merupakan suatu strategi untuk melakukan proses pencarian (*search*) ruang *problem* secara selektif, yang memandu proses pencarian yang kita lakukan disepanjang jalur yang memiliki kemungkinan sukses paling besar.

4. Penarikan Kesimpulan (*Inferencing*)

   AI mencoba membuat mesin memiliki kemampuan berfikir atau mempertimbangkan (*reasoning*). Kemampuan berfikir (*reasoning*) termasuk didalamnya proses penarikan kesimpulan (*inferencing*) berdasarkan fakta-fakta dan aturan dengan menggunakan metode heuristic atau pencarian lainnya.

5. Pencocokan Pola (*Pattern Matching*)

   AI bekerja dengan metode pencocokan pola (*pattern matching*) yang berusaha untuk menjelaskan objek, kejadian (*event*) atau proses, dalam hubungan logika atau komputasional.

2.1.3 Pattern Matching

Pattern Matching atau Pencocokan Pola adalah suatu metode yang digunakan untuk mencocokan suatu pola tertentu (kumpulan huruf) dengan suatu kumpulan kata (teks) atau string. Pada bidang sains komputer metode pattern matching sangat banyak digunakan antara lain Editor Teks, Mesin Pencari Web, Analisis Gambar dan lain-lain. String dapat kita asumsikan

sebagai kumpulan dari beberapa karakter yang membentuk suatu kesatuan. (Budiasa, 2009).

2.1.3.1 Algoritma Brute Force

Algoritma Brute Force merupakan algoritma pencocokan string dengan melakukan pencarian semua kemunculan string *pattern* yang berada di string teks.

Cara Kerja Algoritma Brute Force :

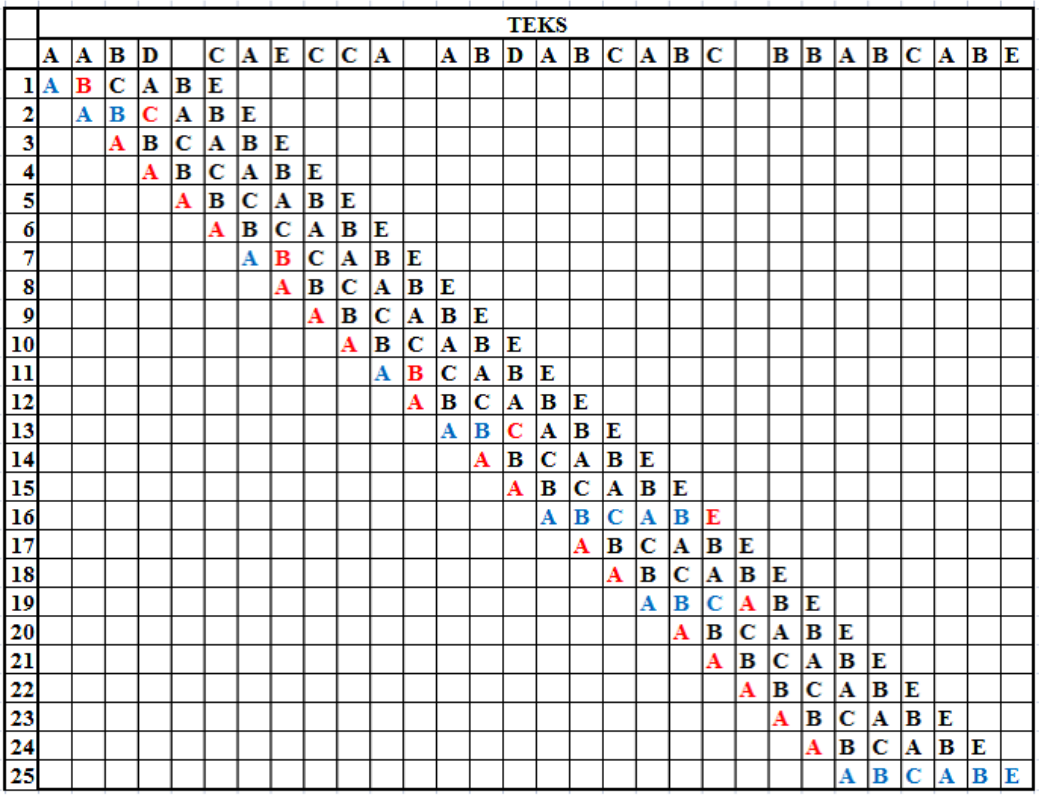
1. Mula-mula *pattern* dicocokan pada awal teks.
2. Dengan bergerak dari kiri ke kanan, bandingkan setiap karakter di dalam *pattern* dengan karakter yang bersesuaian di dalam teks sampai :
   a. Semua karakter yang dibandingkan cocok atau sama (pencarian berhasil), atau
   b. Dijumpai sebuah ketidakcocokan karakter (pencarian belum berhasil)
3. Bila *pattern* belum ditemukan kecocokannya dan teks belum habis, geser *pattern* satu karakter ke kanan dan ulangi langkah 2.

Persoalan pencarian string dirumuskan sebagai berikut :

1. Teks (*text*), yaitu (*long*) string yang panjangnya n karakter
2. Pattern, yaitu string dengan m karakter (m < n) yang akan dicari di dalam teks.

Contoh cara kerja algoritma Brute Force ini adalah sebagai berikut :

- Teks : AABD CAECCA ABDABCABC BBABCABE
- Pattern : ABCABE



Gambar 2.1 Contoh Algoritma Brute Force

2.1.4 Aplikasi Kecerdasan Buatan

Kecerdasan buatan berupa aplikasi-aplikasi yang dapat diterapkan pada komputer besar (*main frame*), komputer mikro, atau *personal computer* (PC). Beberapa aplikasi kecerdasan buatan diantaranya sebagai berikut :

1. *General problem solving*

   Adalah suatu tahapan proses atau langkah-langkah yang berurutan untuk mencapai suatu tujuan, dengan cara melacak dan mengkombinasikan berbagai cara atau metode sehingga menghasilkan solusi terbaik.

2. *Expert system* atau sistem pakar

   Adalah suatu program yang bertindak sebagai penasehat 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 dengan benar.

3. Vision

   Merupakan aplikasi yang dapat mengenali gambar yang diterimanya dari kamera, scanner, dan alat input lainnya dengan cara mencocokan dan melacak gambar apa yang diterimanya melalui kamera atau scanner sebagai masukan.

4. *Natural Language Processing* atau pengolahan bahasa alami

   Merupakan salah satu cabang dari *artificial intellegence* (AI) yang berhubungan dengan pemrosesan bahasa alami oleh komputer yang mencakup metode-metode *speech recognition*, *speech synthesizer* (*text-to-speech*), *parsing*, penterjemahan bahasa, *QA system*, dan kemampuan bahasa alami lainnya pada sebuah mesin atau komputer.

5. Robotics

   Pada aplikasi *robotics* kecerdasan buatan bertindak seolah-olah melakukan pekerjaan fisik yang biasa dikerjakan manusia, bahkan *robot* dapat melakuakn pekerjaan yang tidak dapat dilakukan oleh manusia atau berbahaya bagi manusia. Semua tindakan dirancang dan disusun dalam

satu urutan atau algoritm tertentu. Dengan bantuan *robot*, pekerjaan yang berulang kali dan rumit dapat dilakukan dengan baik.

6. *Education*

   Merupakan aplikasi kecerdasan buatan dalam bidang pendidikan yang bertindak sebagai *partner* bagi pelajar atau mahasiswa dalam mempelajari suatu bidang.

2.1.5 Tujuan Kecerdasan Buatan

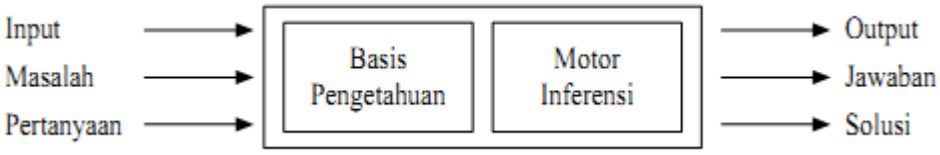
Tujuan dari kecerdasan buatan menurut Winston dan Prendergast :

1. Membuat mesin menjadi lebih pintar (tujuan utama)
2. Memahami apa itu kecerdasan (tujuan ilmiah)
3. Membuat mesin lebih bermanfaat (tujuan *entrepreneurial*)

   Dua bagian utama yang dibutuhkan untuk aplikasi kecerdasan buatan seperti yang terlihat pada Gambar 2.1 Proses Motor inferensi adalah :

   a. Basis Pengetahuan (*Knowledge Base*) berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.

   b. Motor Inferensi (*Inference Engine*) adalah kemampuan menarik kesimpulan berdasarkan pengalaman.



Gambar 2.2 Proses Motor Inferensi

2.1.6 *Chat Bot*

Chat Bot 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 Chat Bot merupakan Aplikasi atau Program komputer yang dirancang untuk meniru manusia itu sendiri, batasan yang diambil dari Chat Bot 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.

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. Beberapa jenis *bot* bekerja

secara *silent* di internet tanpa sepengetahuan kita namun ada pula *bot* yang bekerja secara interaktif yang dapat kita ajak berbicara yaitu *chat bot*.

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

Chat bot yang pertama adalah ELIZA yang dibuat pada tahun 1964 sampai 1966 oleh Professor Joseph Weizenbaum di MIT (*Massachusetts Institute of Technology*), dengan tujuan untuk mempelajari komunikasi *natural language* antara manusia dengan mesin. Eliza bertindak seolah-olah dia adalah seorang psikolog yang dapat menjawab pertanyaan-pertanyaan dari pasien dengan jawaban yang cukup masuk akal atau menjawabnya dengan pertanyaan balik.

ELIZA adalah pelopor atau dapat disebut sebagai nenek moyangnya *chat bot.* ELIZA (juga dikenal sebagai Dr. Eliza) terkenal sebagai program *chat* yang “memiliki profesi” sebagai seorang psikiater. ELIZA mensimulasikan percakapan antara seorang psikiater dengan pasiennya menggunakan metode biasa yang dapat mencerminkan perasaan pasien dengan mengajukan pertanyaan-pertanyaan seperti: "*How do you ...*", "*Why do you feel like ...*", "*What do you think about ...*".

Program ini akan mencari pola kata-kata tertentu pada input yang diberikan oleh pengguna, dan kemudian memberikan output yang sesuai.

ELIZA dapat memberikan respon yang tepat hanya jika pernyaataan pengguna ditulis dalam ejaan yang benar. Meskipun begitu, pengguna dapat berbicara secara santai dengan ELIZA.

ALICE (*Artificial Linguistic Internet Computer Entity*) adalah sebuah program *chat bot* yang diciptakan oleh Dr. Richard Wallace. ALICE mampu menciptakan suasana seperti komunikasi antara dua manusia seperti biasanya, ALICE bahkan dapat berbohong, menyebarkan gosip, dan mengetahui banyak kejadian yang terjadi di dunia nyata.

Ada banyak lagi *chat bot* lain yang menarik seperti Erin the Bartender, Madame Celeste, Mega HAL, dan lain-lain.

2.1.7 Bahasa Alami (*Natural Language*)

Pada prinsipnya bahasa alami adalah suatu bentuk representasi dari suatu pesan yang ingin dikomunikasikan antar manusia. Bentuk utama representasinya adalah berupa suara/ucapan (*spoken language*), tetapi sering pula dinyatakan dalam bentuk tulisan.

Bahasa dapat dibedakan menjadi (1) Bahasa Alami, dan (2) Bahasa Buatan. Bahasa alami adalah bahasa yang biasa digunakan untuk berkomunikasi antar manusia, misalnya bahasa Indonesia, Sunda, Jawa, Inggris, Jepang, dan sebagainya.

Bahasa buatan adalah bahasa yang dibuat secara khusus untuk memenuhi kebutuhan tertentu, misalnya bahasa pemodelan atau bahasa pemrograman komputer.

Chomsky adalah orang yang pertama kali merepresentasikan bahasa sebagai rangkaian simbol. Chomsky berhasil memperlihatkan bahwa bahasa apapun dapat direpresentasikan dengan suatu cara yang universal. Pemikiran Chomsky yang merepresentasikan bahasa sebagai kumpulan simbol-simbol dan aturan yang mengatur susunan simbol-simbol tersebut telah membuka peluang untuk melakukan pemrosesan bahasa secara simbolik dengan teknologi komputer, sehingga melahirkan bidang ilmu Natural Language Processing (NLP).

Linguistik adalah bidang ilmu yang khusus mengkaji tentang bagaimana bahasa distrukturkan dan digunakan. Bidang ilmu ini salah satunya membahas tentang perbendaharaan kata dan leksikon.

Perbendaharaan kata adalah sekumpulan kata-kata dan frase-frase yang digunakan dalam bahasa tertentu. Sebagai bagian dari pengkajian bahasa, linguis mendefinisikan semua kata-kata dan frase-frase yang digunakan secara umum kemudian mengorganisasikannya ke dalam sebuah leksikon. Leksikon adalah kamus yang mendaftar kata-kata bahasa itu secara alfabet.

2.1.7.1 Pengertian *Natural Language Processing*

Pengolahan bahasa alami (NLP) adalah bidang ilmu komputer dan linguistik berkaitan dengan interaksi antara komputer dan manusia. Seluler

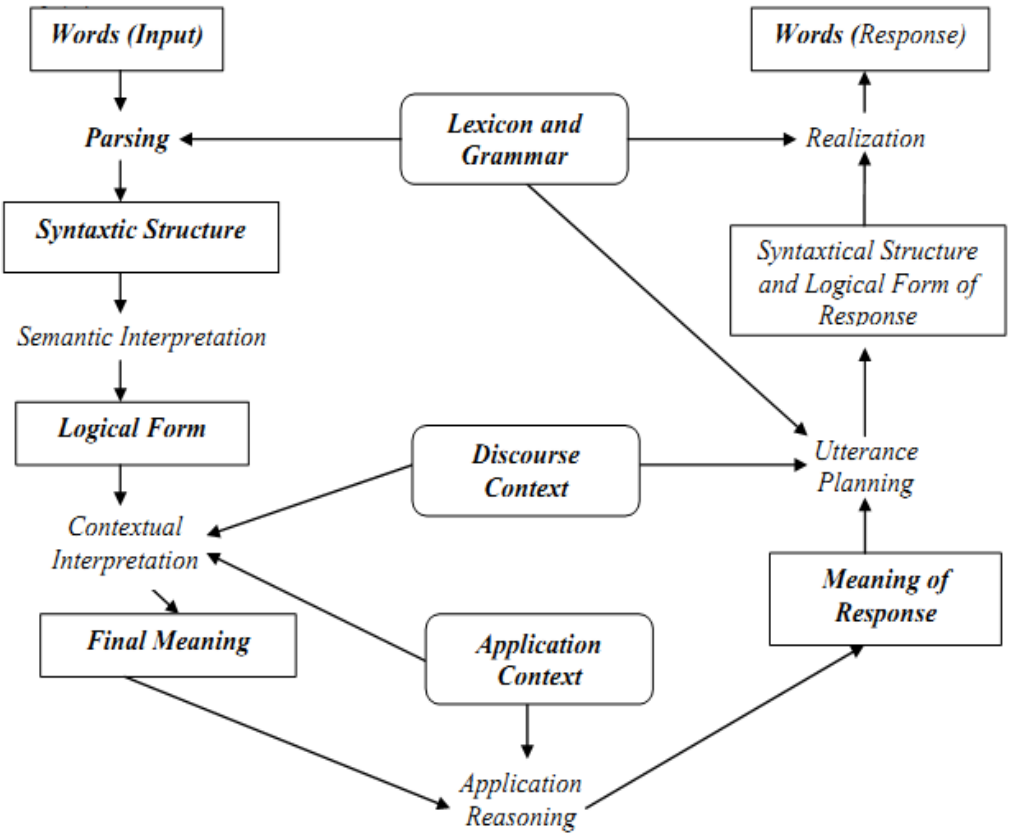
generasi bahasa sistem komputer yang mengubah informasi dari database ke dalam bahasa manusia yang dapat dibaca.

Natural language processing, biasanya disingkat dengan NLP, mencoba membuat komputer mampu memahami suatu perintah yang dituliskan dalam bentuk bahasa sehari-hari dan diharapkan komputer juga merespon dalam bahasa yang mirip dengan bahasa natural. Setelah komputer bisa memahami perintah dalam bahasa natural, maka diharapakan sistem komputer juga dapat memberikan respon dalam bahasa natural pula.

Sebuah sistem natural language harus memperhatikan pengetahuan terhadap bahasa itu sendiri baik dari segi kata yang digunakan bagaimana kata-kata tersebut digabung untuk menghasilkan suatu kalimat, apa arti suatu kata, apa fungsi sebuah kata dalam sebuah kalimat dan sebagainya. Natural Language processor tidak memperdulikan bagaimana suatu kalimat diinputkan ke komputer. Tugasnya adalah mengekstrak informasi dari kalimat. Inti dari sistem NLP adalah parser. Parser adalah bagian dari program atau system yang membaca setiap kalimat, kata demi kata, untuk menentukan "*what is what*".

NLP tidak bertujuan untuk mentransformasikan bahasa yang diterima dalam bentuk teks atau suara menjadi data digital dan/atau sebaliknya pula; melainkan bertujuan untuk memahami arti dari kalimat yang diberikan dalam bahasa alami dan memberikan respon yang sesuai, misalnya dengan melakukan suatu aksi tertentu atau menampilkan data tertentu. Untuk mencapai tujuan ini dibutuhkan tiga tahap proses. Proses yang pertama ialah parsing atau analisa sintaksis yang memeriksa kebenaran struktur kalimat berdasarkan suatu *grammar*

(tata bahasa) dan *lexicon* (kosa kata) tertentu. Proses kedua ialah *semantic* interpretation atauinterpretasi semantik yang bertujuan untuk merepresentasikan arti dari kalimat secara context-independent untuk keperluan lebih lanjut. Sedangkan proses ketiga ialah contextual interpretation atau interpretasi kontekstual yang bertujuan untuk merepresentasikan arti secara context dependent dan menentukan maksud dari penggunaan kalimat. Gambaran organisasi dari sebuah sistem NLP yang lengkap ditunjukkan pada gambar 2.3.



Gambar 2.3 Organisasi Sebuah Sistem NLP

2.1.7.2 Question Answering System (QA System)

Question answering system (QA system) adalah sistem yang mengijinkan user menyatakan kebutuhan informasinya dalam bentuk *natural language question* (pertanyaan dalam bahasa alami), dan mengembalikan kutipan teks singkat atau bahkan frase sebagai jawaban. Ketersediaan sumber informasi yang luas dan bervariasi, serta adanya perkembangan yang pesat dari teknik *Natural Language Processing* (NLP), *Information Extraction* (IE), dan *Information Retrieval* (IR) sangat mempengaruhi perkembangan dari QA *system*, yang mana dulunya hanya dapat menjawab pertanyaan-pertanyaan yang terbatas pada suatu bidang (*domain*) tertentu dengan berdasarkan pada sumber informasi yang terstruktur seperti database, hingga kini dapat menjawab berbagai jenis pertanyaan dengan bersumber pada informasi dari sebuah koleksi teks yang tidak terstruktur. Umumnya arsitektur *question answering system* yang berbasis teks dibangun atas enam tahapan proses, yaitu analisis pertanyaan, preprocessing koleksi dokumen, pemilihan kandidat dokumen, analisis kandidat dokumen, ekstraksi jawaban, dan pemberian respon.

Aplikasi-aplikasi *QA system* (baik yang dapat diakses melalui internet maupun tidak) yang dikembangkan dengan ciri khasnya masing-masing memiliki urutan proses yang tidak jauh berbeda satu dengan lainnya. Jawaban yang dikembalikan oleh sebuah *QA system* sebagai respon terhadap pertanyaan perlu dievaluasi untuk menilai performansi sistem.

2.1.8 Scanner (Analisis Leksikal)

` Scanner merupakan salah satu bagian dari kompilator bahasa pada komputer yang bertugas melakukan analisis leksikal. Analisis leksikal adalah proses pengidentifikasian semua besaran yang membangun suatu bahasa pada program sumber. Scanner menerima input berupa stream karakter kemudian memilah program sumber menjadi satuan leksik yang disebut dengan token. Token ini akan menjadi input bagi parser. Tugas dari scanner adalah sebagai berikut:

1. Melakukan pembacaan kode sumber dengan merunut karakter demi karakter
2. Mengenali besaran leksik
3. Mentransformasikan menjadi sebuah *token* dan menentukan jenis *token*-nya.
4. Mengirimkan token
5. Membuang/mengabaikan blank dan komentar dalam program
6. Menangani kesalahan
7. Menangani table symbol

Di dalam aplikasi NLP sistem cerdas yang akan dibuat, yang dimaksud dengan program sumber yang diolah oleh *scanner* adalah berupa kalimat input dari pengguna dalam bentuk sms.

Ketika scanner menerima input berupa *stream* karakter kemudian memilah menjadi satuan leksik, satuan leksik tersebut terdiri atas simbol-simbol satuan yang jika dikombinasikan akan mempunyai arti yang berbeda-beda. Simbol-

simbol yang bisa dipergunakan dalam sebuah bahasa tentunya terbatas jumlahnya, yang membentuk sebuah himpunan dan disebut sebagai abjad (*alphabet*).

Tata bahasa (grammatika) adalah sekumpulan dari himpunan variabel-variabel, simbol-simbol terminal, simbol non-terminal, simbol awal yang dibatasi oleh aturan-aturan produksi. Aturan produksi adalah pusat dari tata bahasa yang menspesifikasikan bagaimana suatu tata bahasa melakukan transformasi suatu string ke bentuk lainnya.

Dalam pembicaraan grammar, anggota alfabet dinamakan simbol terminal atau token. Kalimat adalah string yang tersusun atas simbol-simbol terminal. Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa berupa tak berhingga hingga kalimat.

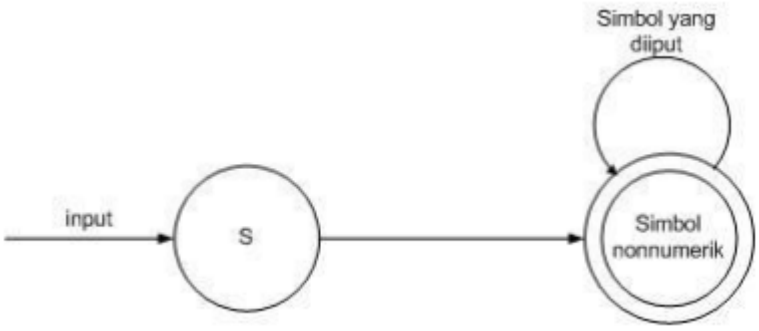
Simbol-simbol berikut adalah symbol terminal :

1. huruf kecil alphabet, misalnya :a, b, c
2. symbol operator, misalnya : +, -, dan ‘
3. symbol tanda baca, misalnya : (,), dan ;

Sedangkan simbol-simbol berikut adalah simbol non terminal :

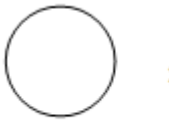
1. huruf besar awal alphabet, misalnya : A, B, C
2. huruf S sebagai simbol awal

Pada saat scanner membaca input, tools yang digunakan untuk menggambarkan perpindahan dari posisi satu ke posisi lainnya adalah diagram transisi.

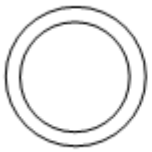


Gambar 2.4 Diagram transisi

Keterangan :


: state/keadaan awal input suatu kalimat


: *looping*/perulangan pembacaan simbol


: state/keadaan akhir suatu kalimat

2.1.9 *Parser* (Analisis Sintaksis)

Parser atau *syntactic analyzer* pada kompilator bahasa pemrograman berfungsi untuk memeriksa kebenaran kemunculan setiap *token*. Pada QA *system*, fungsi dari *parser* ini agak berbeda karena *token* yang akan diolah semua memiliki tipe yang sama yaitu berupa kata (word). Urutan kemunculan *token* yang berupa kata-kata tersebut akan diolah dengan mengacu pada *brain file* agar

didapatkan makna kalimat yang sesungguhnya. Dengan kata lain, tahap analisa semantik terjadi di bagian *brain file*. Kemampuan dari *parser* untuk mengolah *token* dan bekerja sama dengan *brain file* inilah yang paling menentukan tingkat kecerdasan dari sebuah *chat bot*.

2.1.9.1 *Bottom-Up Parsing*

Metode ini melakukan penelusuran dari leaf/daun menuju ke root/puncak. Gramatika yang dipakai akan lebih banyak bercabang ke arah simbol non-terminal. Hal lain yang juga berkaitan erat dengan proses parsing adalah kamus atau leksikon yang digunakan. Dalam leksikon disimpan daftar kata yang dapat dikenali sebagai simbol terminal dalam grammar dan informasi yang diperlukan untuk tiap kata tersebut untuk proses parsing yang bersangkutan. *Bottom Up parser* yaitu mencari dari simbol – simbol terminal menuju ke arah pembentukan simbol awal S.

2.1.9.2 *Top-Down Parsing*

Memulai pemeriksaan dari simbol awal S dan mencoba untuk mencari bentuk simbol terminal berikutnya yang sesuai dengan jenis kata dari kalimat masukan. Ada 2 kelas metoda parsing *top-down*, yaitu kelas metoda dengan backup dan kelas metoda tanpa *backup*. Contoh metoda kelas dengan *backup* adalah metoda *Brute-Force*, sedangkan contoh metoda kelas tanpa *backup* adalah metoda *recursive descent*.

1. *Metode Brute-Force*

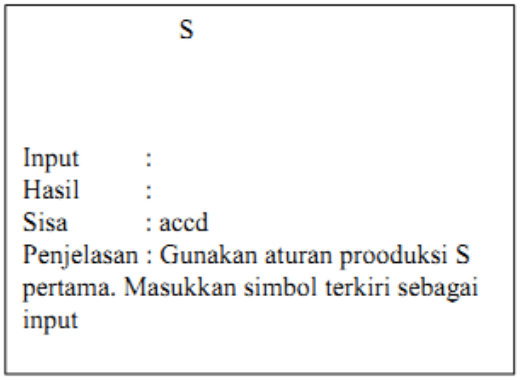
Metode Brute Force memilih aturan produksi mulai dari kiri. Apabila terjadi kalahan pada saat parsing misalnya string tidak sesuai, maka dilakukan baktrack/backup. *Metode* ini membuat pohon *parsing* secara *top down* yaitu degan cara mencoba segala kemungkinan untuk setiap simbol non-terminal. Contoh suatu bahasa dengan aturan produksi sebagai berikut:

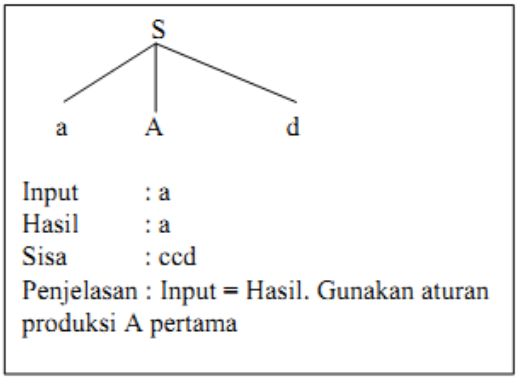
S → aAd | aB

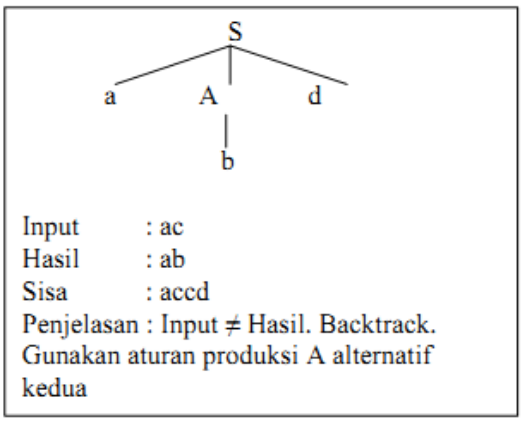
A → b | c

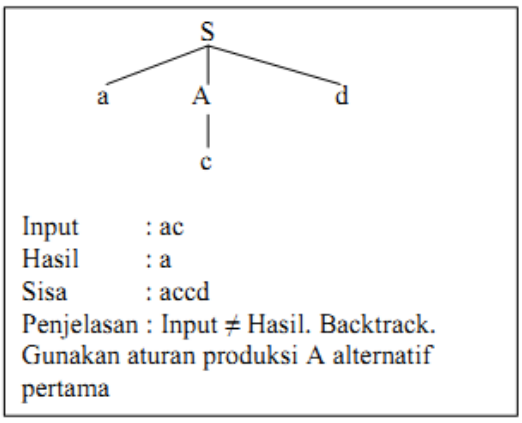
B → ccd | ddc

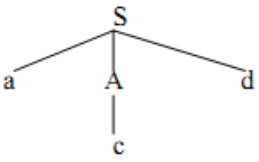
Misal akan dilakukan *parsing* untuk string 'accd'. Maka analisis sintaks terhadap string tersebut dengan menggunakan *metode Brute Force* adalah:



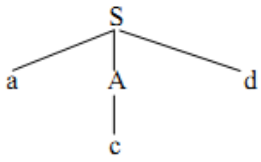




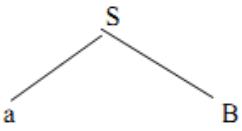




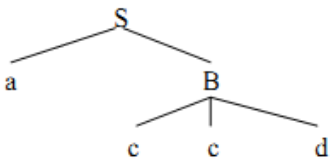
Input : ac
Hasil : ac
Sisa : cd
Penjelasan : Input = Hasil. Karakter berikutnya adalah simbol terminal. Tidak ada lagi aturan produksi A alternatif. Hasil dibandingkan dengan input.



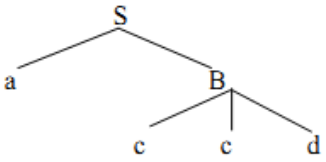
Input : acc
Hasil : acd
Sisa : c
Penjelasan : Input ≠ Hasil. Tidak ada lagi aturan produksi A alternatif. Backtrack. Gunakan aturan produksi S alternatif pertama.



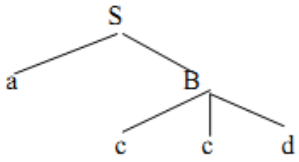
Input : a
Hasil : a
Sisa : ccd
Penjelasan : Input = Hasil. Gunakan aturan produksi B pertama



Input : ac
Hasil : ac
Sisa : cd
Penjelasan : Input = Hasil. Karakter berikutnya adalah simbol terminal. Hasil dibandingkan dengan input.



Input : acc
Hasil : acc
Sisa : d
Penjelasan : Input = Hasil. Karakter berikutnya adalah simbol terminal. Hasil dibandingkan dengan input.



Input : accd
Hasil : accd
Sisa :
Penjelasan : Input = Hasil. Hasil dibandingkan dengan input. Selesai, sukses.

Gambar 2.5 Metode *Brute Force*

2. *Metode Recursive-Descent*

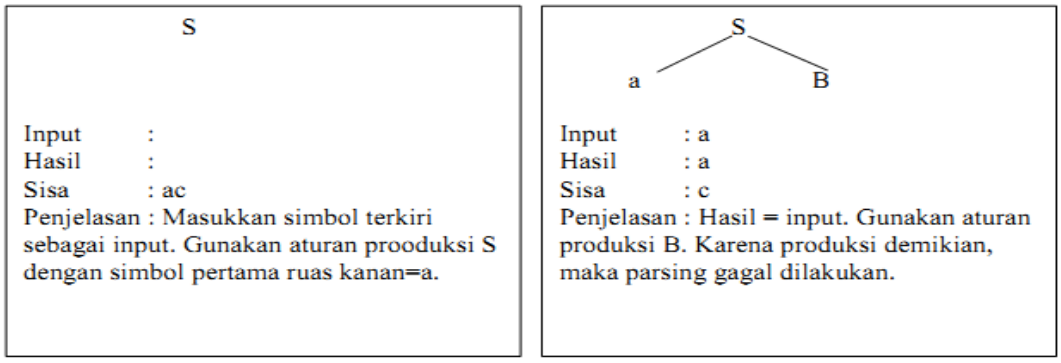
Metode Recursive-Descent adalah kelas metoda *parsing* yang tidak menggunakan produksi alternatif ketika hasil akibat penggunaan sebuah produksi tidak sesuai dengan simbol input. Jika produksi A mempunyai dua buah ruas kanan atau lebih maka produksi yang dipilih untuk digunakan adalah produksi dengan simbol pertama ruas kanannya sama dengan input yang sedang dibaca.
Jika tidak ada produksi yang demikian maka dikatakan bahwa parsing tidak dapat dilakukan. Ketentuan produksi yang digunakan metoda recursive descent adalah : Jika terdapat dua atau lebih produksi dengan ruas kiri yang sama maka karakter pertama dari semua ruas kanan produksi tersebut tidak boleh sama. Ketentuan ini tidak melarang adanya produksi yang bersifat rekursi kiri. Contoh suatu bahasa dengan aturan produksi sebagai berikut:

S → aB| A

A → a

B → b|d

Misal akan dilakukan parsing terhadap string ’ac’. Maka analisis sintaks terhadap string tersebut dengan menggunakan metode recursive descent adalah:



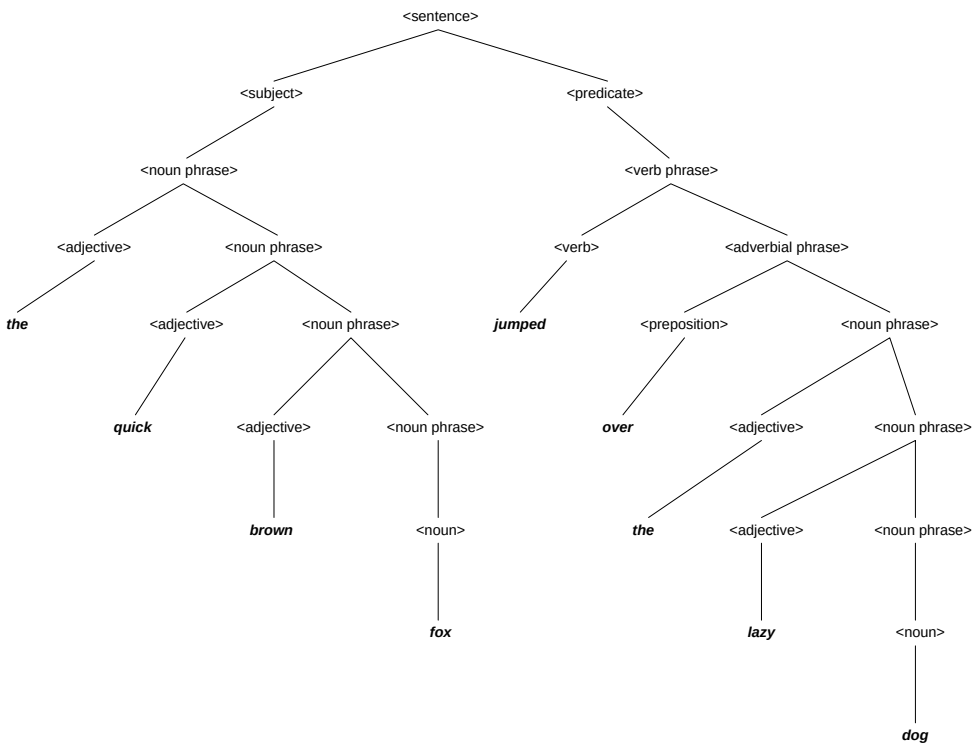
Gambar 2.6 Metode *recursive descent*

2.1.10 Pohon Sintaks

Pohon (*tree*) adalah suatu *graph* terhubung tidak sirkuler, yang memiliki satu simpul (*node*)/*vertex* disebut akar (*root*) dan dari situ memiliki lintasan ke setiap simpul.

Pohon sintaks/pohon penurunan (*syntax tree/derivation tree/ parse tree*) berguna untuk menggambarkan bagaimana memperoleh suatu *string* dengan cara menurunkan simbol-simbol variabel menjadi simbol-simbol terminal. Setiap simbol variabel akan diturunkan menjadi terminal, sampai tidak ada yang belum tergantikan. Gambar 2.2 memberikan contoh sebuah *tree* yang menguraikan kalimat dalam bahasa Inggris.

The quick brown fox jumped over the lazy dog



Gambar 2.7 Pohon Sintaks

Proses penurunan atau *parsing* bisa dilakukan dengan cara:

1. Penurunan terkiri (*leftmost derivation*): simbol variabel terkiri yang diperluas terlebih dulu
2. Penurunan terkanan (*rightmost derivation*): simbol variabel terkanan yang diperluas terlebih dulu.

2.1.11 Analisis Semantik

Analisis semantik merupakan kelanjutan dari proses *scanning* dan *parsing*. Fungsi dari analisis semantik adalah untuk menentukan makna dari serangkaian instruksi yang terdapat dalam program sumber atau masukan dari penguna. Penganalisa semantik harus mampu menentukan aksi atau respon apa yang yang akan dilakukan terhadap instruksi yang diberikan.

2.1.12 *Turing Test*

Turing Test adalah usulan untuk mengetes kemampuan sebuah mesin untuk melakukan percakapan seperti manusia. Digambarkan oleh Alan Turing di Koran pada 1950 ‘Mesin komputasi dan kecerdasan’. Hasilnya sebagai berikut : seorang manusia penilai melakukan percakapan bahasa alami dengan dua pihak lain, satu manusia dan satunya lagi mesin, jika penilai tidak dapat mengatakan yang mana mesin dan yang mana manusia, maka mesin tersebut telah berhasil melalui *test* tersebut. Diasumsikan bahwa baik manusia dan mesin mencoba untuk terlihat seperti manusia. Untuk menjaga *test* diatur lebih sederhana dan *universal* (

secara eksplisit menguji kemampuan bahasa mesin), percakapan biasanya dibatasi dengan teks saja.

2.2 Perangkat Lunak

Perangkat Lunak (*software*) merupakan suatu program yang dibuat oleh pembuat program untuk menjalankan perangkat keras komputer. Perangkat Lunak adalah program yang berisi kumpulan instruksi untuk melakukan proses pengolahan data. *Software* sebagai penghubung antara manusia sebagai pengguna dengan perangkat keras komputer, berfungsi menerjemahkan bahasa manusia ke dalam bahasa mesin sehingga perangkat keras komputer memahami keinginan pengguna dan menjalankan instruksi yang diberikan dan selanjutnya memberikan hasil yang diinginkan oleh manusia tersebut.

Perangkat lunak terdiri dari kode-kode yang dibuat atau ditulis dalam bahasa pemrograman tertentu. Banyak bahasa pemrograman yang dapat dijadikan alat bantu untuk merancang sebuah perangkat lunak antara lain bahasa PASCAL, C, C++, BASIC, Java, HTML, PHP, JavaScript, AIML, dan bahkan tersedia *tool* yang berbasis grafis seperti Visual Basic, Visual C++, Borland Delphi, dan masih banyak lagi *tool* dan bahasa pemrograman lainnya.

2.3 Metode Analisis Yang digunakan

2.3.1 Flowchart

Flowchart merupakan gambaran dalam bentuk diagram alir dari algoritma-algoritma dalam suatu program, yang menyatakan arah alur program tersebut. Hal

tersebut memungkinkan untuk memecah proses menjadi kejadian-kejadian individual atau aktifitas untuk menunjukan secara singkat hubungan diantaranya. Konstruksi flowchart memungkinkan pengertian lebih baik kepada proses dan pengertian yang lebih baik terhadap proses akan membawa kepada perbaikan pengembangan suatu sistem. Adapun simbol-simbol yang digunakan dalam flowchart, dapat dilihat dalam daftar simbol.

2.3.2 DFD (*Data Flow Diagram*)

DFD sering digunakan untuk menggambarkan suatu sistem yang telah ada atau sistem baru yang akan dikembangkan secara logika tanpa mempertimbangkan lingkungan fisik dimana data tersebut akan disimapan. DFD merupakan alat yang digunakan pada metodologi pengembangan sistem yang terstruktur, selain itu merupakan alat yang cukup popular dikarenakan dapat menggambarkan arus data dalam didalam sistem secara jelas dan terstruktur. Dalam mengembangkan suatu aliran data atau proses yang terjadi di dalam sistem data flow diagram menggunakan simbol-simbol yang memiliki arti tersendiri dalam menerangkan:

a. Eksternal Entity

Eksternal entity dapat merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, organisasi atau sistem lainnya, yang memberikan input-output dari sistem.

b. Data Flow

Arus data ini mengatur diantara proses, simpan data, dan kesatuan luar. Arus data ini menujukkan arus data yang dapat berupa masukan sistem atau hasil proses sistem.

c. Proses

Untuk *physical data flow diagram* (PDFD), data dilakukan oleh orang, mesin atau komputer. Sedangkan untuk *logical data flow diagram* (LDFD), suatu proses hanya menujukkan proses dari komputer.

d. Penyimpanan Data

Simpanan data (*data store*) merupakan tempat penyimpanan data. Simpanan data dari DFD disimbolkan dengan sepasang garis *horizontal paralel*.

Konsep dasar DFD dapat dilakukan dengan analisa *Top Down*, yaitu pemecahan sistem yang besar menjadi beberapa sub-sub sistem yang lebih kecil DFD terdiri dari :

a. Context Diagram

Diagram konteks yaitu diagram yang menunjukkan batas dan jangkauan dari sistem informasi yang dibuat. Merupakan gambaran sistem secara garis besar dengan entitas-entitas yang ada dan hanya memperlihatkan kelompok data input dan output. Konteks diagram merupakan level teratas dari diagram arus data . Diagram konteks adalah diagram tingkat atas yang merupakan diagram global dari sistem informasi yang menggambarkan aliran-aliran data dari entitas-entitas yang masuk dan yang keluar dari sistem.

b. *Middle Level*

Merupakan pemecahan dari tiap–tiap proses yang mempunyai fungsi sama. Pada *middle level* diagram 1 dipecah menjadi diagram 2,3,4 dan seterusnya yang merupakan penguraian dari diagram konteks.

c. *Lowest Level* (DFD Level Terendah),

Diagram yang menunjukkan proses yang lebih detail dari level sebelumnya.Merupakan pemecahan dari data flow yang ada pada *middle level*. Pemecahan tersebut masih tetap mempunyai fungsi yang sama dari level sebelumnya. Untuk *Lowest Level*, pemberian nomor diagram terdiri dari bagian *middle level*.

2.3.3 Kamus Data (*Data Dictionary*)

Kamus data adalah katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi.

Dengan adanya kamus data, analisis sistem dapat mdendefinisikan data yang mengalir di dalam sistem dengan lengkap. Pada tahap perancangan sistem, kamus data dapat digunakan untuk merancang input, output, dan merancang database program. Kamus data dibuat berdasarkan arus data yang ada.

2.3.4 ERD (Entity Relationship Diagram)

Model E-R didasarkan pada persepsi bahwa dunia nyata merupakan sekumpulan dari sejumlah objek dasar (entitas) dan relasi antar objek-objek data tersebut. Diagram yang menggambarkan struktur lojik keseluruhan basis data, simbol yang digunakan adalah Persegi empat, merepresentasikan himpunan entitas (untuk entitas lemah diberi garis ganda), Elips, merepresentasikan atribut, Wajik, merepresentasikan himpunan keterhubungan, Garis, menghubungkan simbol-simbol pada diagram. Label dari persegi empat, elips, dan wajik menunjukkan nama, Kardinalitas pemetaan dinyatakan dengan 2 cara : [Korth] garis berarah (1) dan garis tidak berarah (Banyak), [*Date*] menuliskan kardinalitasnya pada garis dan Peran dapat dituliskan sebagai label dari garis.

Pemakaian elemen-elemen dalam ERD ada tiga diantaranya sebagai berikut:

1. *Entity* (Entitas) adalah sebuah objek yang dapat dibedakan dari objek-objek lainnya, yang memiliki sejumlah *property* atau atribut, dimana setiap atribut memiliki sekumpulan nilai yang diizinkan yang disebut domain, himpunnan entitas yaitu kumpulan jumlah entitas yang memiliki tipe yang sama dan sebuah basis data mengandung sekumpulan himpunan entitas yang masing-masingnya memiliki sejumlah entitas dari tipe yang sama.
2. *Relationship* (relasi) merupakan hubungan antar entitas yaitu sebuah relasi menggambarkan suatu asosiasi antar sejumlah entitas, himpunan relasi (*Relationsip set*) adalah kumpulan sejumlah relasi yang memiliki tipe yang

sama yang merupakan relasi matematis terhadap dua atau lebih himpunan entitas : {(e1, e2,…, en) (e1 E1, e2 E2 ,…, en En)}, Jumlah entitas terlihat dalam 2 buah relasi disebut derajat. Kebanyakan relasi yang muncul adalah relasi binary, ada beberapa yang ternary, lebih dari itu sangat jarang, Fungsi sebuah entitas di dalam relasi disebut peran (role) dan Sebuah relasi dapat memiliki atribut.

2.4 Perangkat Lunak Pengembang

2.4.1 PHP

PHP adalah bahasa *scripting server – side*, artinya bahasa yang digunakan pada *server* dengan tanpa perlu melakukan kompilasi tetapi cukup menuliskan tulisan dalam bentuk ASCII-nya saja. PHP sangat mirip dengan bahasa C, juga mempunyai karakteristik yang mirip dengan Perl. PHP dapat digunakan untuk mengolah data dari berbagai macam *database*, contoh struktur bahasa PHP adalah sebagai berikut :

```
<?php
echo " Syntax Program";
?>
```

2.4.2 MySQL

MySQL merupakan susunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structure Query Language*). Kendala dari suatu sistem database (DBMS) dapat diketahui dari cara kerja optimezernya dalam melakukan perintah-perintah SQL yang dibuat user ataupun program-program aplikasinya"

KAD[8]. Sebagai *database* server, MySQL dapat dikatakan lebih unggul dibanding database server lainnya dalam query data. MySQL adalah salah satu dari sekian banyak sistem *database* yang merupakan terobosan solusi yang tepat dalam aplikasi *database*.

MySQL adalah multi user database yang menggunakan bahasa Strucktured Query Language (SQL). MySQL mampu menangani data yang cukup besar. Perusahaan yang mengembangkan MySQL yaitu TcX, mengaku menyimpan data lebih dari 40 database, 10.000 tabel dan sekitar 7 juta baris, totalnya kurang lebih 100 Gigabyte data. SQL adalah bahasa standar yang digunakan untuk mengakses database server. Bahasa ini pada awalnya dikembangkan oleh IBM, namun telah diadopsi 25 dan digunakan sebagai standar industri.

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).

2.5 Basis Data

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

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 elektronis.

2.5.1 Tujuan basis data

Tujuan basis data adalah sebagai berikut :

1. Kemudahan dan kecepatan dalam pengambilan kembali data atau arsip.
2. Efisiensi ruang dan waktu
3. Keakuratan data
4. Ketersediaan untuk proses pengambilan data yang diperlukan setiap saat
5. Kelengkapan data-data yang diperlukan atau yang tersimpan
6. Keamanan data
7. Kebersamaan

2.5.2 Keuntungan Basis data

1. Mereduksi redudansi yang akibatnya mengurangi inkonsistensi.
2. Data dapat dishare antar aplikasi.
3. Standarisasi data dapat dilakukan.
4. Batasan *security* dapat diterapkan.
5. Mengelola integritas (keterjaminan akurasi) data.
6. Menyeimbangkan kebutuhan yang saling konflik.
7. Independensi data (objektif DBS) : kekebalan aplikasi terhadap perubahan struktur penyimpanan dan teknik pengaksesan data (basis data harus dapat berkembang tanpa mempengaruhi aplikasi yang telah ada).