BAB II

LANDASAN TEORI

2.1 Konsep Dasar Kecerdasan Buatan

2.1.1 Pengertian Kecerdasan Buatan

Pada mulanya manusia menggunakan komputer hanya untuk mengolah data yang ada, untuk menghasilkan informasi dalam pengambilan keputusan. Seiring perkembangan komputer, maka kegunaan komputer semakin besar, sampai kini dapat mengolah pengetahuan, sehingga proses pengambilan keputusan dapat lebih cepat.

Apabila komputer mengerjakan pengolahan data maka manusia harus mengkonversikan data tersebut menjadi informasi yang dapat digunakan dalam mengolah pengetahuan untuk menghasilkan keputusan. Bila komputer mengerjakan pengolahan informasi, juga termasuk pengolahan data, maka manusia cukup mengerjakan pengolahan pengetahuan untuk menghasilkan keputusan. Akan tetapi apabila komputer dapat mengerjakan pengolahan pengetahuan, maka akan sangat sedikitlah bagian pekerjaan yang harus dikerjakan oleh manusia, termasuk untuk menerapkan hasilnya untuk keperluan tertentu.

Kecerdasan buatan didefinisikan sebagai kecerdasan yang ditunjukan oleh suatu entitas buatan. Kecerdasan dibuat, diciptakan dan dimasukkan ke dalam suatu mesin (komputer) agar dapat melakukan pekerjaan seperti yang dapat dilakukan manusia. Kecerdasan buatan bertujuan untuk mengembangkan suatu program komputer yang dapat memecahkan suatu masalah tertentu, dengan cara

yang dapat dianggap cerdas jika dilakukan oleh manusia. Penelitian dalam kecerdasan buatan menyangkut pembuatan mesin untuk mengotomatisasikan tugas-tugas yang membutuhkan perilaku cerdas. Termasuk contohnya adalah pengendalian, perencanaan dan penjadwalan, kemampuan untuk menjawab dignosa dan pertanyaan pelanggan, serta pengenalan tulisan tangan, suara dan wajah.

Menurut beberapa ahli kecerdasan buatan didefinisikan sebagai berikut :

1. Menurut Rich and knight [1991] [10] : "*Kecerdasan buatan (artificial intelligence) merupakan sebuah studi tentang bagaimana membuat computer 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 tingkah laku cerdas dalam sistem kecerdasan komputer.

Kecerdasan buatan dilihat dari berbagai sudut pandang adalah sebagai berikut :

1. Sudut pandang Kecerdasan (Intelligence)

   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.

   Domain penelitian adalah sebagai berikut :

a. Mundane task

i. Persepsi (vision and speech)

ii. Bahasa alami (understanding, generation and translation)

iii. Pemikiran yang bersifat commonsense

iv. Robot control

b. Format task

i. Permainan atau games

ii. Matematika (geometri, logika, kalkulus, integral, pembuktian)

c. Expert task

i. Analisis finansial

ii. Analisis medis

iii. Analisis ilmu pengetahuan

iv. Rekayasa (desain, pencarian, kegagalan, perencanaan, manufaktur)

3. Sudut pandang Bisnis

Kumpulan peralatan yang sangat powerful dan metodologis dalam menyelesaikan masalah-masalah bisnis.

4. Sudut pandang Pemrograman (*Programming*)

Kecerdasan buatan termasuk didalamnya adalah studi tentang pemrograman simbolik, pemecahan masalah, proses pencarian (*search*).

2.1.2 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 mencocokkan 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 intelligence* (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 r*obotics* kecerdasan buatan bertindak seolah-olah melakukan pekerjaan fisik yang biasa dikerjakan manusia, bahkan robot dapat melakukan pekerjaan yang tidak dapat dilakukan oleh manusia atau berbahaya bagi manusia. Semua tindakan dirancang dan disusun dalam satu urutan atau algoritma tertentu. Dengan bantuan robot, pekerjaan yang berulang kali dan rumit dapat dilakukan dengan baik.

6. *Education*

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

2.1.3 Tujuan Kecerdasan Buatan

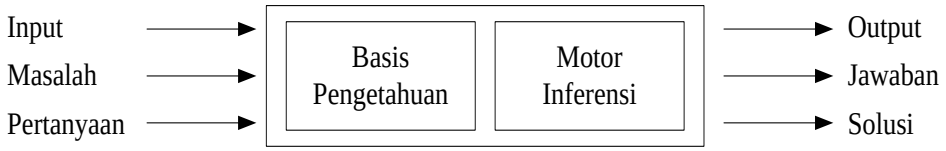
Tujuan dari kecerdasan buatan menurut Winston dan Prendergast :

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 kecerdasarn 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.

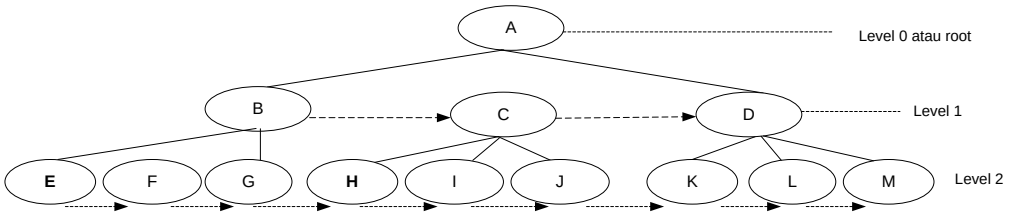


Gambar 2.1 Proses Motor Inferensi

2.1.4 Metode Pencarian Pada Kecerdasan Buatan

Hal penting dalam menentukan keberhasilan sistem berdasar kecerdasan adalah kesuksesan dalam pencarian dan pencocokan. Pada dasarnya ada dua metode pencarian, yaitu pencarian buta (*blind search*) dan pencarian terbimbing (*heuristic search*). Namun metode pencarian yang digunakan pada tulisan ini adalah pencarian buta khususnya pencarian melebar pertama (*breadth first search*).

Pada metode *breadth first search*, semua node pada level n akan dikunjungi terlebih dahulu sebelum mengunjungi node-node pada level n+1. Pencarian dimulai dari node akar lalu ke level 1 dari kiri ke kanan, kemudian berpindah ke level berikutnya demikian pula dari kiri ke kanan hingga ditemukannya solusi atau tujuan. Adapun penggambaran struktur pohon metode *breadth first search* adalah sebagai berikut:



Gambar 2.2 Metode pencarian *breadth first search*

2.2 Bahasa Alami (*Natural Language*)

Bahasa sebagai bagian yang penting dalam kehidupan manusia. Dalam bentuk tulis dapat berupa catatan dari pengetahuan yang didapat oleh umat manusia dari satu generasi ke generasi berikutnya, sedangkan dalam bentuk lisan merupakan sarana komunikasi antar individu dalam suatu masyarakat. Tujuan dalam bidang *Natural Language* adalah melakukan proses pembuatan model komputasi dari bahasa, sehingga dapat terjadi suatu interaksi antara manusia dan komputer dengan perantaraan bahasa alami. Model komputasi ini dapat berguna untuk keperluan ilmiah misalnya meneliti sifat-sifat dari suatu bentuk bahasa alami maupun untuk keperluan sehari-hari dalam hal ini memudahkan komunikasi antara manusia dengan komputer.

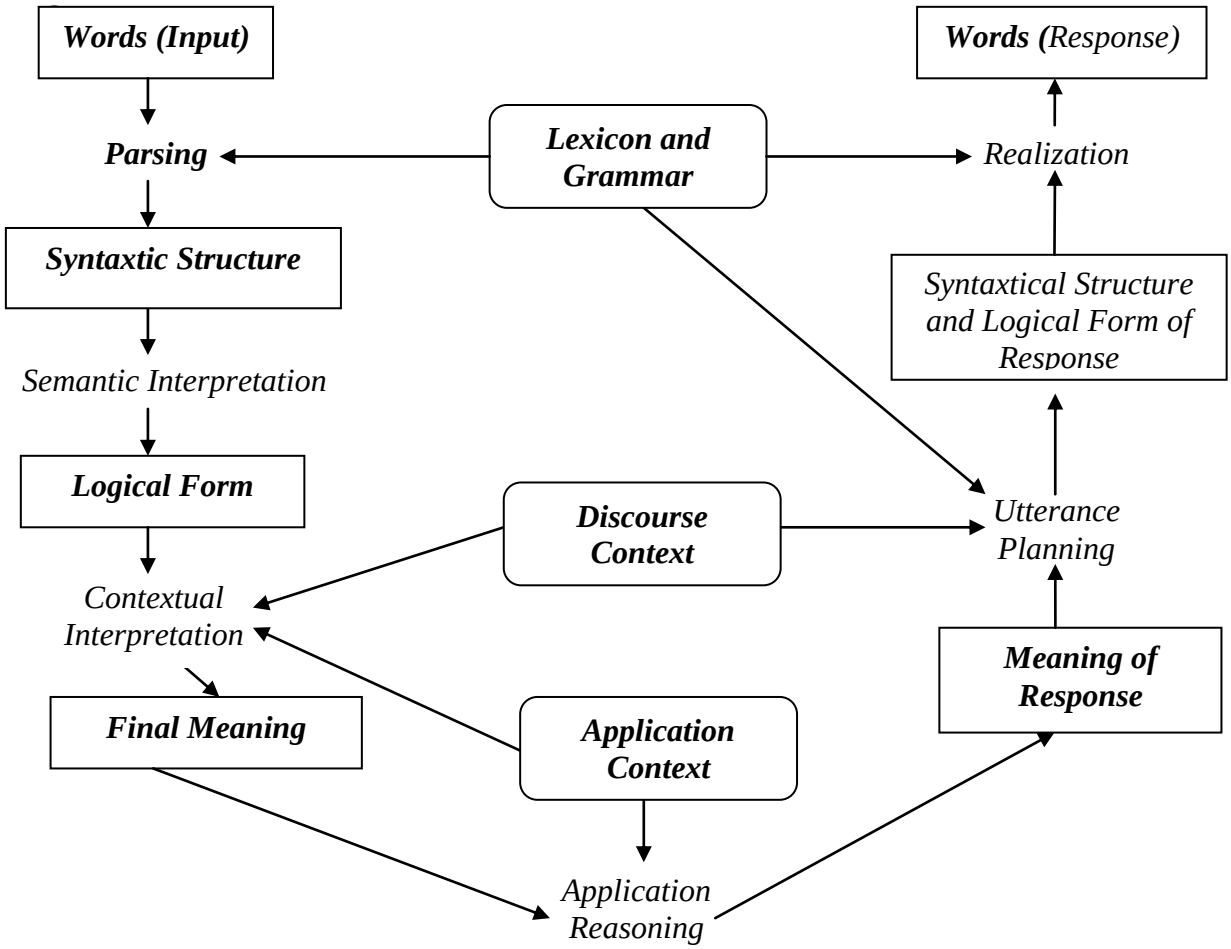
2.2.1 Pengertian Natural Language Processing

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 kta 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* atau interpretasi 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 2.2.



Gambar 2.3 Organisasi Sebuah Sistem NLP

2.2.2 Aplikasi Dalam Bidang *Natural Language*

Jenis aplikasi yang bisa dibuat pada bidang bidang *natural language* adalah *text - based application* dan *dialogue - based applications*.

1. *Text - based application*

   Mencakup segala macam aplikasi yang melakukan proses terhadap text tertulis seperti misalnya buku, berita di surat kabar, e-mail dan lain sebagainya. Contoh penggunaan dari text - based application ini adalah :

   a. mencari topik tertentu dari buku yang ada pada perpustakaan
   b. memberikan respon atas input yang diberikan
   c. mencari isi dari surat atau e-mail
   d. menterjemahkan dokumen dari satu bahasa ke bahasa yang lain

2. *Dialogue - based application*

   Idealnya pedekatan ini melibatkan bahasa lisan atau pengenalan suara, akan tetapi bidang ini juga memasukkan interaksi dengan cara memasukkan teks pertanyaan melalui keyboard. Aplikasi yang sering ditemui untuk bidang ini adalah :

   a. sistem tanya jawab, dimana *natural language* digunakan dalam mendapatkan informasi dari suatu database.
   b. sistem otomatis pelayanan melalui telepon
   c. control suara pada peralatan elektronik
   d. sistem problem - solving yang membantu untuk melakukan penyelesaian masalah yang umum dihadapi dalam suatu pekerjaan.

2.2.3 *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. Mentransformasi menjadi sebuah *token* dan menentukan jenis *token*-nya.
4. Mengirimkan *token*
5. Membuang/mengabaikan *blank* dan komentar dalam program
6. Menangani kesalahan
7. Menangani tabel simbol

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.

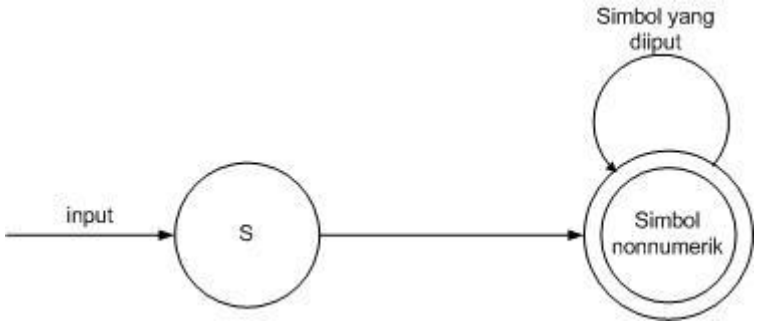
Simbol-simbol berikut adalah simbol terminal :

1. huruf kecil awal alfabet, misalnya : a, b, c
2. simbol operator, misalnya : +, -, dan ´
3. simbol tanda baca, misalnya : (,), dan ;

Sedangkan simbol-simbol berikut adalah simbol non terminal :

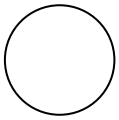
1. huruf besar awal alfabet, misalnya : A, B, C
2. huruf S sebagai simbol awal

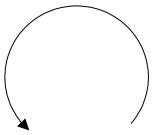
Pada saat *scanner* membaca input, tools yang digunakan untuk menggambarkan perpindahan dari posisi satu ke posisi lainnya adalah diagram transisi.

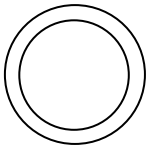


Gambar 2.4 Diagram transisi

Keterangan:

 : state/keadaan awal input suatu kalimat

 : *looping*/perulangan pembacaan simbol

 : state/keadaan akhir suatu kalimat

2.2.4 *Parser* (Analisis Sintaksis)

Parser atau *syntactic analyzer* pada kompilator bahasa pemrograman berfungsi untuk memeriksa kebenaran kemunculan setiap *token*. Pada sistem yang akan dibuat, 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 bank data agar didapatkan makna kalimat yang sesungguhnya. Dengan kata lain, tahap analisa semantik terjadi di bagian bank data. Kemampuan dari *parser* untuk mengolah *token* dan bekerja sama dengan bank data inilah yang paling menentukan tingkat kecerdasan dari sistem yang akan dibuat.

Proses parsing dapat dibagi menjadi dua bagian besar metode yaitu *Top Down parsing* dan *Bottom Up parsing*.

2.2.4.1 *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.

2.2.4.2 *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 kesalahan pada saat parsing misalnya string tidak sesuai, maka dilakukan *backtrack/backup*. Metode ini membuat pohon parsing secara top down yaitu dengan cara mencoba segala kemungkinan untuk setiap simbol non-terminal. Contoh suatu bahasa dengan aturan produksi sebagai berikut:

$S \rightarrow aAd \mid aB$

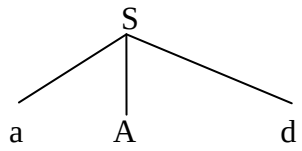
A → b | c

B → ccd | ddc

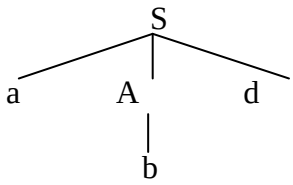
Misal akan dilakukan parsing untuk string ‘accd’. Maka analisis sintaks terhadap string tersebut dengan menggunakan metode *Brute Force* adalah:

S

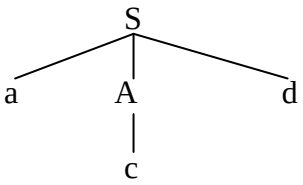
Input :
Hasil :
Sisa : accd
Penjelasan : Gunakan aturan prooduksi S pertama. Masukkan simbol terkiri sebagai input



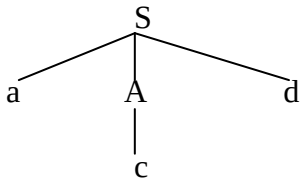
Input : a
Hasil : a
Sisa : ccd
Penjelasan : Input = Hasil. Gunakan aturan produksi A pertama



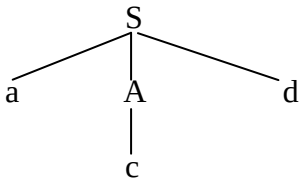
Input : ac
Hasil : ab
Sisa : accd
Penjelasan : Input ≠ Hasil. Backtrack. Gunakan aturan produksi A alternatif kedua



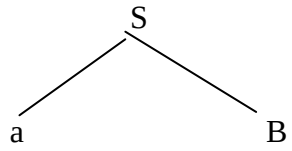
Input : ac
Hasil : a
Sisa : accd
Penjelasan : Input ≠ Hasil. Backtrack. Gunakan aturan produksi A alternatif pertama



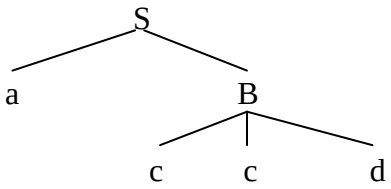
Input : ac
Hasil : ac
Sisa : cd
Penjelasan : Input = Hasil. Karakter berikutnya adalah simbol terminal. Tidak ada lagi aturan produksi A alternatif. Hasil dibandingkan dengan input.



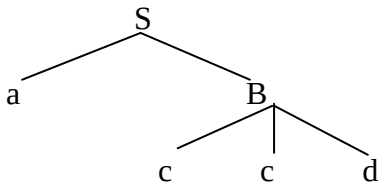
Input : acc
Hasil : acd
Sisa : c
Penjelasan : Input ≠ Hasil. Tidak ada lagi aturan produksi A alternatif. Backtrack. Gunakan aturan produksi S alternatif pertama.



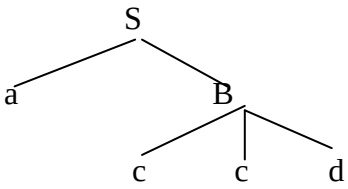
Input : a
Hasil : a
Sisa : ccd
Penjelasan : Input = Hasil. Gunakan aturan produksi B pertama



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.

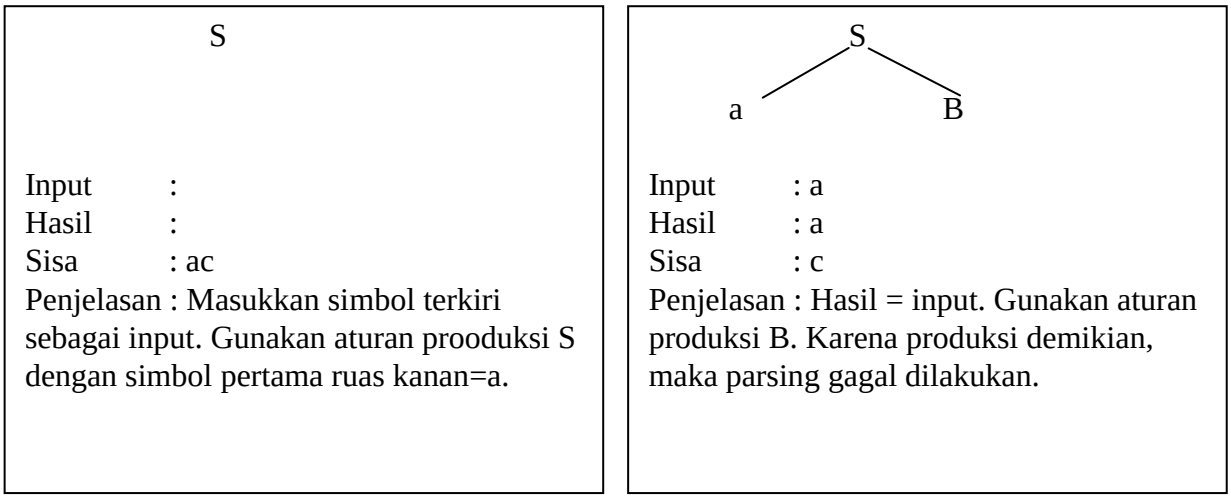
Jika tidak ada produksi yang demikian maka dikatakan bahwa parsing tidak dapat dilakukan. Ketentuan produksi yang digunakan metoda *recursive descent* adalah : Jika terdapat dua atau lebih produksi dengan ruas kiri yang sama maka karakter pertama dari semua ruas kanan produksi tersebut tidak boleh sama. 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*

2.2.5 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.

2.3 Konsep Dasar Komunikasi Data

Kata komunikasi berasal dari kata berbahasa Inggris *communications*. Kata *communications* berasal dari bahasa Latin *communicare* yang berarti saling berbagi (*share*). Komunikasi Data merupakan bagian dari teknologi informasi, dimana komputer melakukan pengiriman data berupa informasi yang disajikan oleh isyarat digital biner terhadap komputer yang lain atau juga pengiriman data terhadap terminal. Proses ini melibatkan suatu pengirim (*transmitter*), penerima (*receiver*) dan sebuah media transmisi untuk tempat mengalirnya informasi. Sehingga jaringan telekomunikasi diartikan sebagai suatu system yang terbentuk dari interkoneksi fasilitas-fasilitas yang dirancang untuk membawa trafik dari beragam sumber telekomunikasi.

Perkembangan teknologi telekomunikasi terasa semakin cepat, terutama dengan pesatnya kemajuan teknologi komputer dan informatika. Dan seiring dengan bertambahnya tahun, komunikasi tanpa kabel (*wireless*) cukup diminati di berbagai Negara sabagai salah satu solusi untuk mencukupi kebutuhan sarana telekomunikasi. Teknologi informasi dan komunikasi (infokom) berkembang semakin pesat didorong oleh *Internet Protocol* (IP) dengan berbagai aplikasi baru dan berbagai layanan multimedia. Infrastuktur infokom terdiri dari *public switched data network* (PSDN) dan *public switched telepon network* (PSTN), namun hingga kini tulang punggung infokom masih banyak berpijak pada jaringan PSTN. PSTN atau yang biasa disebut jaringan atau saluran telepon

konvemsional yang menggunakan kabel. PSTN secara umum diatur oleh standar-standar teknis yang dibuat oleh ITU-T dan menggunakan pengalamatan E.163/E.164 (secara umum dikenal dengan nomor telepon). PSTN dilengkapi dengan telekomunikasi kabel interlokal dan local.

2.3.1 Short Message Service (SMS)

SMS merupakan sebuah layanan yang banyak diaplikasikan pada komunikasi tanpa kabel, memungkinkan dilakukannya pengiriman pesan-pesan dalam bentuk *alphanumeric* antara terminal pelanggan dengan sistem eksternal seperti email, paging, voice mail, dan lain-lain. Isu SMS pertama kali muncul di belahan Eropa pada sekitar tahun 1991 bersama sebuah teknologi komunikasi *wirelsess* yang saat ini cukup banyak penggunaannya yaitu *Global System for Mobile* (GSM) .

Pada proses pengiriman SMS dari handphone, SMS tersebut tidak langsung dikirim ke handphone tujuan, tetapi terlebih dahulu dikirim ke SMS center (SMSC), kemudian dengan sistem store and forward SMS tersebut dikirimkan ke handphone tujuan. Layanan SMS merupakan sebuah layanan yang bersifat *nonreal time* dimana sebuah *short message* dapat di submit ke suatu tujuan, tidak peduli apakah tujuan tersebut aktif atau tidak. Bila dideteksi bahwa tujuan tidak aktif, maka sistem akan menunda pengiriman ke tujuan hingga tujuan aktif kembali. Pada dasarnya sistem SMS akan menjamin *delivery* (terkirim) dari suatu *short message* hingga sampai ke tujuan. Kegagalan pengiriman yang bersifat sementara seperti tujuan tidak aktif akan selalu teridentifikasi sehingga pengiriman *short message* akan selalu dilakukan kecuali apabila diberlakukan aturan bahwa *short*

message yang telah melampaui batas waktu tertentu harus dihapus dan dinyatakan gagal terkirim.

2.3.2 Karakteristik SMS

Selain sebagai media pengirim dan penerima pesan *alphanumeric*, SMS juga dapat digunakan sebagai pengangkat muatan biner (*binary payload*) dan mengimplementasikan tumpukan (*stack*) wap lewat *Short Message Service Centre* (SMSC). Ada beberapa karakteristik pesan SMS yang penting yaitu :

a. Prinsip kerja SMS ini adalah bahwa setiap jaringan mempunyai suatu *Service Centre* (SC). Pesan tidak langsung dikirimkan ke tempat tujuan, melainkan disimpan terlebih dahulu di SC. SC juga dijadikan sebagai interface antara *Public Land Mobile Network* (PLMN).
b. Transmisi SMS dapat terjadi meskipun *Mobile Station* (MS) sedang melakukan komunikasi dengan MS lain. Hal ini dimungkinkan karena kanal radio untuk transmisi *voice* telah ditentukan selama durasi pemanggilan sedangkan pesan SMS merambat pada kanal radio dengan memanfaatkan jalur *signaling*.
c. Pengiriman pesan yang menggunakan kanal signaling memiliki dua tipe yaitu:
    1. SMS *point to point* : menyediakan mekanisme untuk mengirimkan pesan hanya dari satu MS ke MS tertentu.
    2. SMS *broadcast* (*point to multi point*) : pengiriman SMS ke beberapa MS sekaligus.
d. Pesan dijamin sampai atau tidak sampai sama sekali, selayaknya email, sehingga apabila terjadi kegagalan sistem, timeout atau hal lain yang

menyebabkan pesan tidak diterima akan diberi informasi atau report yang menyatakan pesan gagal dikirim.

e. Berbeda dengan fungsi *call* (pemanggilan) sekalipun saat mengirimkan pesan, MS tidak aktif atau diluar jangkauan *service area*, bukan berarti pengiriman pesan akan gagal, namun pesan akan masuk ke antrian dulu selama belum *time out*, pesan akan segera dikirimkan jika MS sudah aktif atau sudah berada di *service area*.

2.3.3 Elemen Dasar Jaringan SMS

Elemen dasar suatu jaringan SMS seperti berikut ini :

1. *Short Messaging Entities* (SME) disebut juga ESME, suatu piranti yang dapat menerima dan mengirim pesan pendek, entitas dalam system SMS yang dapat berada pada jaringan, berupa perangkat brgerak, dan merupakan *service centre* yang berada diluar jaringan.
2. *Short Message Service Centre* (SMSC), entitas yang bertanggung jawab untuk menyimpan, routing, dan meneruskan short message antar SME dan piranti bergerak (*mobile phone*). Terminology SMSC mengacu pada sesuatu yang berupa *hardware* dan *software*.
3. *Signaling System 7* (SS7), protokol *signaling* yang umum digunakan dalam jaringan telepon seluluer.
4. *Base Station System* (BSS), kesatuan system yang bertanggung jawab mengatur transmisi sinyal elektromagnetik untuk membawa data dari MSC ke perangkat telepon bergerak. *Base station* terdiri dari *Base Station Controller* (BSC) dan *Base Transceiver Station* (BTS).

5. MSC, sebuah system yang melakukan *switching* dan mengontrol panggilan telepon dalam suatu jaringan komunikasi bergerak. Berfungsi mengirimkan *short message* ke suatu tujuan tertentu melalui *base station* yang sesuai.
6. *Home Location Register* (HLR), database yang digunakan sebagai tempat penyimpanan permanen data, pengelolaan, dan profil layanan pelanggan.
7. *Visitor Location Register* (VLR), database tempat menyimpan informasi temporal berisi data pelanggan yang berasal dari suatu HLR yang roaming ke HLR lainnya.

2.3.4 Elemen Layanan

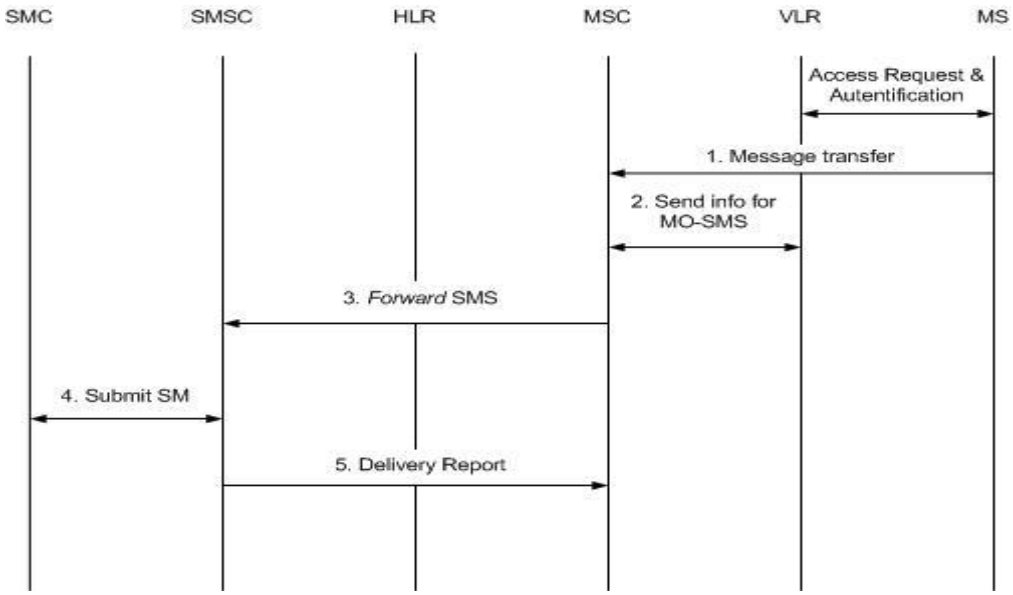
SMS terdiri dari beberapa elemen layanan yang relevan terhadap penerimaan dna pengiriman pesan pendek yaitu:

1. *Message expiration,* SMS akan menyimpan dan mencoba mengirimkan kembali pesan yang mengalami kegagalan sampai pengiriman tersebut berhasil.
2. *Priority,* untuk member tanda pesan-pesan yang penting dan membedakannya dari pesan biasa.

   Sistem SMS memiliki dua layanan dasar *point-to-point* bagi pelanggan yaitu:

   a. *Mobile-Oriented* (MO) *Short Message*, dikirimkan dari *mobile phone* yang MO-Capable ke SMSC dan dapat ditujukan ke *mobile phone* lainnya. Pada layanan ini selalu ada laporan yang dikirimkan ke mobile phone, baik yang mengkonfirmasikan pengiriman pesan pendek ke SMSC maupun

mengkonfirmasikan kegagalan pengiriman dan mengidentifikasikan penyebabnya.



Gambar 2.7 Skenario Pengiriman *Mobile Oriented Short Message* (MO-SM)

Keterangan:

⟷ Pemanggilan operasi

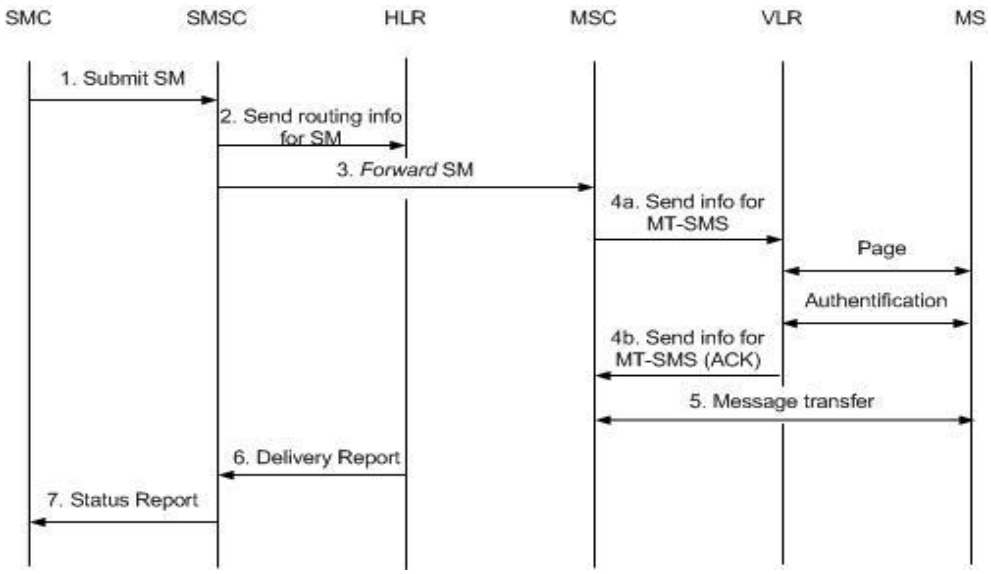
⟶ Pemanggilan operasi yang sukses

Penjelasan dari gambar 2.5 yaitu sebagai berikut:

1. MS mengirimkan SM ke MSC
2. MSC menginterogasi VLR untuk membuktikan bahwa pengiriman pesan tersebut tidak melanggar permintaan layanan pembatasan yang telah ditetapkan.
3. MSC mengirimkan pesan pendek ke SMSC dengan menggunakan operasi *forward shorth message*.
4. SMSC mengirimkan pesan pendek ke SME

5. SMSC memberitahu MSC mengenai keberhasilan operasi *forward shorth message.*
6. MSC mengembalikan hasil dari operasi MO-SM ke MS.

b. *Mobile Terminated* (MT) *Short Message,* dikirimkan dari SMSC ke *mobile phone* lainnya melalui MO-SM, pada layanan ini juga terdapat laporan yang diberikan kepada SMSC yang isinya bisa berupa konfirmasi pengiriman pesan pendek ke *mobile phone* maupun informasi kegagalan pengiriman pesan. Untuk lebih jelasnya, perhatikan gambar … yaitu skenario pengiriman MT-SM.



Gambar 2.8 Skenario Pengiriman *Mobile Terminated Short Message* (MT-SM)

Keterangan :

: Pemanggilan operasi

: Pemanggilan operasi yang sukses

Penjelasan dari gambar 2.6 yaitu sebagai berikut:

1. Pesan pendek dikirimkan dari SME ke SMSC.

2. Setelah menyelesaikan pengolahan internalnya, SMSC menginterogasi HLR dan menerima informasi *routing* untuk pelanggan *mobile*.
3. SMSC mengirimkan pesan pendek ke MSC dengan menggunakan operasi *forward shorth message.*
4. MSC mengambil informasi pelanggan dari VLR. Operasi ini dapat melibatkan prosedur autentifikasi.
5. MSC mengirimkan pesan pendek ke MS.
6. MSC mengembalikan hasil dari operasi *forward shorth message* ke SMSC.
7. Jika diminta oleh SME, SMSC akan mengembalikan laporan status yang mengindikasikan pengiriman pesan pendek.

2.3.5 Protocol Data Unit (PDU)

Dalam proses pengiriman atau penerimaan SMS, data yang dikirim maupun yang diterima oleh stasiun bergerak menggunakan salah satu dari dua mode yang ada, yaitu mode teks atau mode PDU (*Protocol Data Unit*). PDU berisi bilangan-bilangan heksadesimal yang mencerminkan bahasa input output. PDU terdiri dari beberapa *Header*. *Header* untuk kirim SMS ke SMS-*Centre* berbeda dengan SMS yang diterima dari SMS-*Centre*.

1. Format PDU untuk membaca SMS

Setiap pengiriman sms baik dari ponsel menuju operator maupun sebaliknya selalu menggunakan format PDU. Format PDU tersebut berisi informasi mengenai tanggal, nomor tujuan, nomor pengirim, nomor operator, jenis skema

sms, masa valid sms, dan beberapa hal lain tergantung jenis paketnya. Berikut ini contoh PDU yang diterima oleh ponsel:

07 91 2658050000F0 04 0C 91 265836164900 00 00 123070 41403180 04 C830FB0D

Dengan keterangan sebagai berikut:

Tabel 2.1 PDU membaca SMS

Oktet / Digit Hexa	Keterangan
07 91 2658050000F0	SCA
04	PDU Type
0C 91 265836164900	OA
00	PID
00	DCS
123070 41403182	SCTS
04	UDL
C830FB0D	UD

Penjelasan:

1. *Service Center Address* (SCA)

   SCA adalah alamat (nomor) SMSC operator pengirim. Memiliki tiga komponen, yaitu:

   a. Len

      Panjang informasi SMSC dalam octet, dalam hal ini 7 pasangan.

   b. *Type of number*

      81h = format lokal

      91h = format internasional

c. *Service center number*

Nomor SMSC operator pengirim. Karena panjang digit ganjil, maka pada digit paling belakang ditambahkan huruf F. Yang dalam hal ini menggunakan SMSC 62855000000.

- Beberapa SMSC number:

  Satelindo = 62816124 (PDU = 26181642)

  Telkomsel = 6281100000 (PDU = 2618010000)

  Excelcom = 62818445009 (PDU = 26184854009)

  IM3 = 62855000000 (PDU = 26580500000)

2. PDU Type

Nilai default PDU type untuk SMS yang diterima adalah 04 hexa.

3. *Originator Address* (OA)

Nomor pengirim. Yang terdiri dari tiga komponen, yaitu:

a. Len

Panjang digit nomor pengirim dalam hal ini 0C heksadesimal = 12 desimal.

b. *Type of number*

Format dari nomor si penerima SMS.

81h = format lokal

91h = format internasional

c. Nomor pengirim adalah alamat (nomor) handphone pengirim SMS

4. *Protocol Identifier* (PID)

   Protocol Identifier (PID) adalah format dari pengiriman pesan biasanya diatur oleh handphone, apakah pesan berupa Text, Fax, E-mail, telex dan lain-lain. Nilai default dari PID adalah ’00’ yang berarti pesan yang dikirim berupa teks standar.

5. *Data Coding Sceme* (DCS)

   Rencana dari pengkodean data untuk menentukan SMS yang akan dikirim apakah berupa SMS teks standar, bliking SMS, atau flash SMS. Nilai default DCS adalah ’00’ yang menunjukkan bahwa SMS yang dikirim berupa teks standar.

6. *Service Center Time Stamp* (SCTS)

   Adalah waktu dari penerimaan pesan oleh SMSC penerima. Terdiri atas tahun (yy), bulan (MM), tanggal (dd), jam (hh), menit (mm), detik (ss) serta zone waktu. Dalam hal ini waktu penerimaannya adalah 21-03-07 dan jam 14:04:13, sedangkan 82 disini maksudnya adalah 28 unit, dimana 1 unit = 15 menit, maka 28 unit = 28*15/60 = 7 jam, sehingga menjadi GMT + 07.00 = WIB.

7. *User Data Length* (UDL)

   Adalah panjang pesan SMS yang diterima dalam bentuk teks standar. Dalam hal ini 4 huruf.

8. User Data (UD)

   Adalah pesan yang diterima dalam format heksadesimal yang diterjemahkan ke dalam 8 bit lalu dirubah ke ASCII didapat pesan ‘Halo’.

2. PDU untuk mengirimkan SMS

PDU yang diterima HP berbeda dengan PDU untuk mengirim, namun tetap menggunakan aturan yang sama. Perbedaannya terutama pada *field* PDU yang dikandung, baik nilai maupun fungsinya. Berikut ini contoh PDU yang dikirm:

07 91 2658050000F0 11 00 0C 91 261822988804 00 00 FF 04 C830FB0D

Tabel 2.2 PDU pengiriman SMS

Oktet / Digit Hexa	Keterangan
07 91 2658050000F0	SCA
11	PDU Type
00	MR
0C 91 261822988804	DA
00	PID
00	DCS
FF	VP
04	UDL
C830FB0D	User Data

Penjelasan:

3. *Service Center Address* (SCA)

   SCA adalah alamat (nomor) SMSC. Memiliki tiga komponen, yaitu:

   4. Len

      Panjang informasi SMSC dalam octet, dalam hal ini 7 pasangan.

   5. Type of number

      81h = format lokal

      91h = format internasional

6. *Service center number*

   Nomor SMSC operator pengirim. Karena panjang digit ganjil, maka pada digit paling belakang ditambahkan huruf F.

7. PDU Type

   Nilai default PDU type untuk SMS yang dikirim adalah 11 hexa

8. Message Reference (MR)

   Message Reference adalah acuan dari pengaturan pesan SMS. Untuk membiarkan pengaturan SMS diatur sendiri oleh handphone tujuan, maka nilai yang diberikan adalah ’00’.

9. Destination Address

   Adalah nomor tujuan, yang terdiri dari:

   10. Len

       Menunjukkan panjang digit nomor tujuan, dalam hal ini 0C heksadesimal = 12 desimal.

   11. *Type of number*

       81h = format local

       91h = format internasional

   12. Nomor tujuan

       Adalah alamat (nomor) handphone yang akan dikirim SMS.

   13. *Protocol Identifier* (PID)

   Protocol Identifier (PID) adalah format dari pengiriman pesan biasanya diatur oleh handphone, apakah pesan berupa Text, Fax, E-mail, telex dan lain-lain. Nilai default dari PID adalah ’00’ yang berarti pesan yang dikirim berupa teks standar.

14. *Data Coding Sceme* (DCS)

    Rencana dari pengkodean data untuk menentukan SMS yang akan dikirim apakah berupa SMS teks standar, bliking SMS, atau flash SMS. Nilai default DCS adalah ’00’ yang menunjukkan bahwa SMS yang dikirim berupa teks standar.

15. Validity Period (VP)

    Validity period adalah lama waktu pesan SMS disimpan di SMSC apabila pesan tersebut gagal diterima oleh handphone penerima. FF berarti waktu maksimum.

16. *User Data Length* (UDL)

    Adalah panjang pesan SMS yang diterima dalam bentuk teks standar. Dalam hal ini 4 huruf.

17. User Data (UD)

    Adalah pesan yang diterima dalam format heksadesimal yang diterjemahkan ke dalam 8 bit lalu dirubah ke ASCII didapat pesan ‘Halo’.

2.4 Metode Analisis Yang digunakan

2.4.1 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.

2.4.2 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.*

2.4.3 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.

2.4.4 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 : $\{(e_1, e_{2,\ldots,} e_n) \mid (e_1 \in E_{1,} e_2 \in E_2 ,\ldots, e_n \in E_n)\}$, 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.

2.5 Perangkat Lunak Pengembang

2.5.1 Borland Delphi

Borland delphi merupakan suatu bahasa pemrograman yang memberikan berbagai fasilitas pembuatan aplikasi visual. Keunggulan bahasa

pemrograman ini terletak pada produktivitas, kualitas, pengembangan perangkat lunak, kecepatan kompilasi, pola desain yang menarik serta diperkuat dengan pemrogramannya yang terstruktur. Keunggulan lain dari Delphi adalah dapat digunakan untuk merancang program aplikasi yang memiliki tampilan seperti program aplikasi lain yang berbasis windows. Untuk pemrograman *database*, Borland Delphi menyediakan dan mendukung beberapa format *database*, seperti Microsoft Acces, Oracle, MySQL dan lain-lain.Dengan banyaknya format *database* yang mampu didukung oleh Borland Delphi, serta dengan terintegrasinya komponen-komponen untuk berinteraksi dengan *database* tersebut, menjadikan bahasa pemrograman ini menjadi salah satu bahasa pemrograman yang banyak diminati di kalangan *programer* dalam membangun aplikasi yang menggunakan *database*.

2.5.2 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 *optimezer*nya 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*).

2.6 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.

2.6.1 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

2.6.2 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).