BAB II LANDASAN TEORI

2.1 Ilmu gizi

Gizi (*Nutrition*) merupakan suatu proses organisme menggunakan makanan yang dikonsumsi secara normal melalui proses digesti, absorpsi, transportasi, penyimpanan, metabolisme dan pengeluaran zat-zat yang tidak digunakan, untuk mempertahankan kehidupan, pertumbuhan dan fungsi normal dari organ-organ, serta menghasilkan energi.

Zat Gizi (*Nutrients*) adalah ikatan kimia yang diperlukan tubuh untuk melakukan fungsinya, yaitu menghasilkan energi, membangun dan memelihara jaringan serta mengatur proses-proses kehidupan. Ilmu Gizi (*Nutrience Science*) adalah ilmu yang mempelajari segala sesuatu tentang makanan dalam hubungannya dengan kesehatan optimal/ tubuh.

Makanan adalah bahan selain obat yang mengandung zat-zat gizi dan atau unsur-unsur/ikatan kimia yang dapat diubah menjadi zat gizi oleh tubuh, yang berguna bila dimasukkan ke dalam tubuh. Bahan makanan adalah makanan dalam keadaan mentah.

Status gizi adalah keadaan tubuh sebagai akibat konsumsi makanan dan penggunaan zat-zat gizi.

2.1.1 Pengelompokan zat gizi menurut kebutuhan

Komponen terbesar dari susunan diet, berfungsi untuk menyuplai energi dan zat-zat esensial (pertumbuhan sel/ jaringan), pemeliharaan aktivitas tubuh. Karbohodrat (hidrat arang), lemak, protein, vitamin dan mineral.

1. ***Makronutrien***

   Golongan makronutrien terdiri dari :

   a. Karbohidrat – Glukosa; serat.

   b. Lemak/ lipida – Asam linoleat (omega-6); asam linolenat (omega-3).

   c. Protein – Asam-asam amino; leusin; isoleusin; lisin; metionin; fenilalanin; treonin; valin; histidin; nitrogen nonesensial.

2. ***Mikronutrien***

   Golongan mikronutrien terdiri dari :

   a. Mineral: Kalsium; fosfor; natrium; kalium; sulfur; klor; magnesium; zat besi; selenium; seng; mangan; tembaga; kobalt; iodium; krom fluor; timah; nikel; silikon, arsen, boron; vanadium, molibden.

   b. Vitamin: Vitamin A (retinol); vitamin D (kolekalsiferol); vitamin E (tokoferol); vitamin K; tiamin; riboflavin; niacin; biotin; folasin/folat; vitamin B6; vitamin B12; asam pantotenat; vitamin C.

   c. Air

2.1.2 Fungsi zat gizi

Zat gizi memiliki 3 fungsi utama, yaitu:

1. Memberi energi (zat pembakar)

Karbohidrat, lemak dan protein, merupakan ikatan organik yang mengandung karbon yang dapat dibakar dan dibutuhkan tubuh untuk melakukan kegiatan/aktivitas.

2. Pertumbuhan dan pemeliharaan jaringan tubuh (zat pembangun)

   Protein, mineral dan air, diperlukan untuk membentuk sel-sel baru, memelihara, dan menganti sel yang rusak.

3. Mengatur proses tubuh (zat pengatur)

   Protein, mineral, air dan vitamin. Mineral dan vitamin sebagai pengatur dalam proses-proses oksidasi, fungsi normal saraf dan otot serta banyak proses lain yang terjadi dalam tubuh, seperti dalam darah, cairan pencernaan, jaringan, mengatur suhu tubuh, peredaran darah, pembuangan sisa-sisa/ ekskresi dan lain-lain proses tubuh.

2.1.3 Efek ketidakstabilan zat gizi

1. Akibat Gizi Kurang pada Proses Tubuh

   Kekurangan gizi secara umum (makanan kurang dalam kuantitas dan kualitas) menyebabkan gangguan pada proses-proses:

   a. Pertumbuhan
   b. Produksi tenaga
   c. Pertahanan tubuh
   d. Struktur dan fungsi otak
   e. Perilaku

2. Akibat Gizi Lebih pada Proses Tubuh

Gizi lebih menyebabkan kegemukan atau obesitas. Kelebihan energi yang dikonsumsi disimpan di dalam jaringan dalam bentuk lemak. Kegemukan adalah salah satu faktor terjadinya berbagai penyakit degeneratif seperti:

a. Hipertensi
b. Diabetes mellitus
c. Jantung koroner
d. Hati dan kandung empedu.

2.1.4 Perhitungan kesesuain gizi

Menghitung kesesuaian gizi memiliki beberapa cara. Acuan yang digunakan melalui kesepakatan "Konsensus Pengendalian dan Pencegahan Diabetes Mellitus tipe 2 di Indonesia 2011" sebagai berikut:

1. Perhitungan antropometri

   Perhitungan antropometri merupakan salah satu cara untuk mencari keadaan ideal seseorang berdasarkan tinggi badan yang dimiliki. Antropometri juga dapan menjadi salah-satu acuan untuk menyatakan keadaan gizi seseorang. Rumus dasar yang digunakan untuk menghitung berat badan yang disesuaikan oleh tinggi badannya.

   Berdasarkan konsensus pengendalian dan pencegahan Diabetes Mellitus tipe 2 di Indonesia 2011, perhitungan Berat Badan Ideal (BBI) dengan rumus *Brocca* yang dimodifikasi adalah sebagai berikut:

   a. BBI = 90% x (TB dalam cm – 100) x 1 Kg
   b. Bagi Pria dengan Tinggi Badan (TB) dibawah 160 cm dan wanita dibawah 150 cm , rumus dimodifikasi menjadi:

BB Normal : BBI $\pm$ 10%

Kurus : < BBI – 10%

Gemuk : >BBI + 10%

2. Penentuan status dan kebutuhan gizi berdasarkan antropometri

Status gizi akan dihitung berdasarkan keadaan antropometri pasien yang akan disesuaikan kebutuhan gizinya hingga mendekati keadaan ideal antropometrinya. Perhitungan dan penentuan status gizi ini lebih dikenal dengan perhitungan IMT (Indeks Masa Tubuh).

$$\text{IMT} = \frac{\text{BB (kg)}}{\text{TB}^2 \text{ (m)}}$$

Keteranga:

BB: Berat Badan

TB: Tinggi Badan

Range IMT dinyatakan pada tabel sebagai berikut:

Tabel II.1 Range IMT

No	Besar IMT	Kondisi Gizi
1	18,5	Berat badan kurang
2	18,5 – 22, 9	Berat badan normal
3	> 23	Berat badan lebih
4	23 – 24,9	Pre-Obesitas
5	25 – 29,9	Obesitas 1
6	> 30	Obesitas 2

3. Perhitungan kebutuhan kalori

a. Jenis kelamin

Wanita : 25 kal/kg

Laki-laki : 30 kal/kg

b. Usia

Usia 40-59 tahun, kalori dikurangi 5%

Usia 60-69 tahun, kalori dikurangi 10%

Usia diatas 70 tahun dikurangi 20%

c. Aktifitas fisik atau pekerjaan

Bed rest : kal ditambah 10 %

Ringan : kal ditambah 20%

Sedang : kal ditambah 30%

Berat : kal ditambah 50%

d. Berat badan

Bila kegemukan : dikurangi 20-30%

Bila kurus : di tambah 20-30%

Untuk penurunan berat badan kalori minimal yang diberikan sebesar:

- Wanita : 1000-1200 kKal perhari
- Laki-laki : 1200-1600 kKal perhari

4. Pengaturan pola makan normal

Pengaturan pola makan normal yaitu untuk *user* normal (tidak mengidap diabetes) yang ingin mengolah menu hariannya sesuaikan dengan hasil dari poin 1 dan 2.

a. Perhitungan kebutuhan karbohidrat

$$\text{Karbo} = \frac{\text{A\% x total kKal}}{4}$$

b. Perhitungan kebutuhan lemak

$$\text{Lm} = \frac{\text{B\% x total kKal}}{9}$$

c. Perhitungan kebutuhan protein

$$Pr = \frac{\text{C\% x total kKal}}{4}$$

Keterangan: A = [55...65]

B = [20...30]

C = [10...15]

5. Pengaturan pola makan diabetes

   Pengaturan pola makan untuk penderita diabetes disesuaikan berdasarkan hasil dari poin 1, 2, dan aturan diet diabetes.

   Rumus yahg digunakan sama dengan poin 4, namun menghindari penggunaan gula murni maupun *glukosa simple*.

6. Informasi kandungan gizi menu kreasi

   Hal ini menghasilkan kandungan gizi dari sebuah menu kreasi/ menu yang dibuat sendiri oleh *user*. Dihitung dengan menjumlahkan jenis makanan yang dikonsumsi dan mengalikannya dengan jumlah porsi yang dimakan.

2.2 Diabetes

Diabetes mellitus adalah suatu penyakit gangguan kesehatan di mana kadar gula dalam darah seseorang menjadi tinggi karena gula dalam darah tidak dapat digunakan oleh tubuh. Diabetes Mellitus (DM) dikenal juga dengan sebutan penyakit gula darah atau kencing manis yang mempunyai jumpah penderita yang cukup banyak di Indonesia juga di seluruh dunia.

Pada orang yang sehat karbohidrat dalam makanan yang dimakan akan diubah menjadi glokosa yang akan didistribusikan ke seluruh sel tubuh untuk dijadikan energi dengan bantuan insulin. Pada orang yang menderita kencing manis, glukosa sulit masuk ke dalam sel karena sedikit atau tidak adanya zat insulin dalam tubuh. Akibatnya kadar glukosa dalam darah menjadi tinggi yang nantinya dapat memberikan efek samping yang bersifat negatif atau merugikan.

Kadar gula yang tinggi akan dibuang melalui air seni. Dengan demikian air seni penderita kencing manis akan mengandung gula sehingga sering dilebung atau dikerubuti semut. Selanjutnya orang tersebut akan kekurangan energi/ tenaga, mudah lelah, lemas, mudah haus dan lapar, sering kesemutan, sering buang air kecil, gatal-gatal, dan sebagainya.

Kandungan atau kadar gula penderita diabetes saat puasa adalah lebih dari 126 mg/dl dan saat tidak puasa atau normal lebih dari 200 mg/dl. Pada orang normal kadar gulanya berkisar 60-120 mg/dl. Penyakit yang akan ditimbulkan oleh penyakit gula darah ini adalah gangguan penglihatan mata, katarak, penyakit jantung, sakit ginjal, impotensi seksual, luka sulit sembuh dan membusuk/gangren, infeksi paru-paru, gangguan pembuluh darah, stroke dan sebagainya.

Tidak jarang bagi penderita yang parah bisa amputasi anggota tubuh karena pembusukan. Oleh sebab itu sangat dianjurkan melakukan perawatan yang serius bagi penderita serta melaksanakan/menjalani gaya hidup yang sehat dan baik bagi yang masih sehat maupun yang sudah sakit.

2.2.1 Tipe diabetes mellitus

Diabetes mellitus tidak murni 100% penyakit keturunan. Diabetes dapat pula diakibatkan oleh kebiasaan buruk seperti merokok, minuman keras, narkotika, faktor stres serta pola makan yang tidak teratur.

1. Diabetes tipe 1

Terjadi karena kerusakan sel-sel beta pankreas yang mengakibatkan terjadinya *defisiensi* (penurunan) produksi insulin. *Defisiensi* insulin dapat menyebabkan terjadinya *hiperglicemia* (peningkatan kadar gula *glukosa* darah), *polyuria* (sering buang air kecil), *polydipsia*, kehilangan berat badan, kekurangan cairan (*dehidrasi*), ketidakseimbangan *elektrolit*, dan *ketoasidosis*.

Tipe ini disebut juga dengan penyakit diabetes ketergantungan insulin atau insulin-dependent diabetes mellitus (IDDM) dan disebut juga diabetes pada remaja. Diabetes tipe 1 umum terjadi karna faktor keturunan dengan jumlah penderita mencapai 5 – 10% dari keseluruhan jumlah penderita diabetes.

2. Diabetes tipe 2

Diabetes mellitus tipe 2 merupakan tipe diabetes yang paling banyak, jumlah penderitanya mencapai 80% dari keseluruhan jumlah penderita diabetes. Faktor resiko dari terjadinya diabetes mellitus tipe 2 adalah obesitas (kegemukan), usia lanjut, riwayat keluarga dengan diabetes, riwayat diabetes saat kehamilan, mempunyai riwayat rusaknya keseimbangan glukosa, dan jarang melakukan aktifitas fisik.

3. Diabetes kehamilan

Diabetes kehamilan adalah suatu kondisi di mana terjadi ketidaktahanan terhadap glukosa (intolerance glucose) pada wanita hamil. Kondisi diabetes melitus ini hanya terjadi selama masa kehamilan. Niasanya terdiaknosis pada kehamilan trismester 2 dan 3, yaitu mulai bulan ke-4 kehamilan sampai kehamilan berakhir. Kejadian diabetes mililitus ini hanya 4% dari seluruh kehamilan.

4. Diabetes tipe khusus

Diabetes tipe khusus adalah diabetes mellitus yang terjadi karena faktor genetik, pembedahan, obat;obatan, gizi salah (*malnutrition*), infeksi, dan penyakit lainnya. Jumlah penderita diabetes tipe khusus hanya 1 – 2% dari keseluruhan penderita diabetes mellitus.

5. Diabetes karena rusaknya keseimbangan glukosa

Diabetes karena rusaknya keseimbangan glukosa diklasifikasikan menjadi dua, yaitu sebagai berikut.

a. *Impaired fasting glucose* (IFC) yang dijelaskan dengan kadar glukosa darah puasa lebih besar dari 110 mg/dl, tetapi kurang dari 126 mg/dl.
b. *Impaired glukose tolerance* (IGT) yang dijelaskan dari hasil tes toleransi glukosa lebih besar dari 140 mg/dl, tetapi kurang dari 200 mg/dl.

Walaupun IFG dan IGT secarra klinis bukan diabetes yang sesungguhnya, tetapi penyakit ini dapat berkembangn menjadi penyakit diabetes dan penyakit jantung.

2.3 Kecerdasan Buatan

Kecerdasan buatan adalah salah satu bidang ilmu komputer yang mendayagunakan komputer sehingga dapat berperilaku cerdas seperti manusia. Ilmu komputer tersebut mengembangkan perangkat lunak dan perangkat keras untuk menirukan tindakan manusia. Aktifitas manusia yang ditirukan seperti penalaran, penglihatan, pembelajaran, pemecahan masalah, pemahaman bahasa alami dan sebagainya. Sesuai dengan definisi tersebut, maka teknologi kecerdasan buatan dipelajari dalam bidang–bidang seperti : Robotika (*Robotics*), Penglihatan komputer (*Computer Vision*), Pengolahan bahasa alami (*Natural Language Processing*), Pengenalan pola (*Pattern Recognition*), Sistem Syaraf Buatan (*Artificial Neural System*), Pengenalan Suara (*Speech Recognition*), dan Sistem Pakar (*Expert System*).

Kecerdasan Buatan menyelesaikan permasalahan dengan mendayagunakan komputer untuk memecahkan masalah yang komplek dengan cara mengikuti proses penalaran manusia. Salah satu teknik kecerdasan buatan yang menirukan proses penalaran manusia adalah Sistem pakar. (Hartati, 2008)

2.3.1 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

   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

   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

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

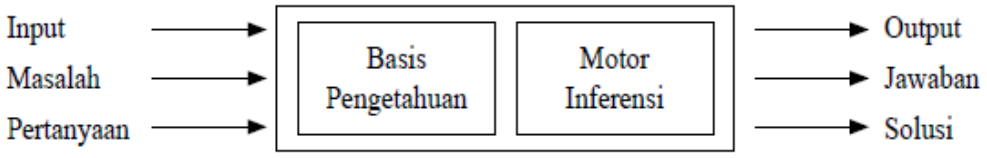
2.3.2 Tujuan Kecerdasan Buatan

Tujuan dari kecerdasan buatan menurut Winston dan Prendergast dalam *Decision support systems and intelligent systems*, 2005 :

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 II.1 Proses Motor Inferensi adalah :

1. Basis Pengetahuan (*Knowledge Base*) berisi fakta-fakta, teori, pemikiran dan hubungan antara satu dengan lainnya.
2. Motor Inferensi (*Inference Engine*) adalah kemampuan menarik kesimpulan berdasarkan pengalaman.

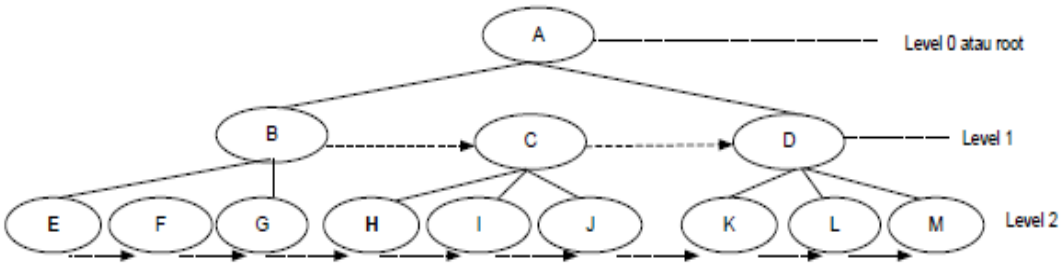


Gambar II.1 Proses motor inferensi

2.3.3 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 II.2 Metode pencarian *breadth first search*

2.4 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.

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 di*input*kan 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*".

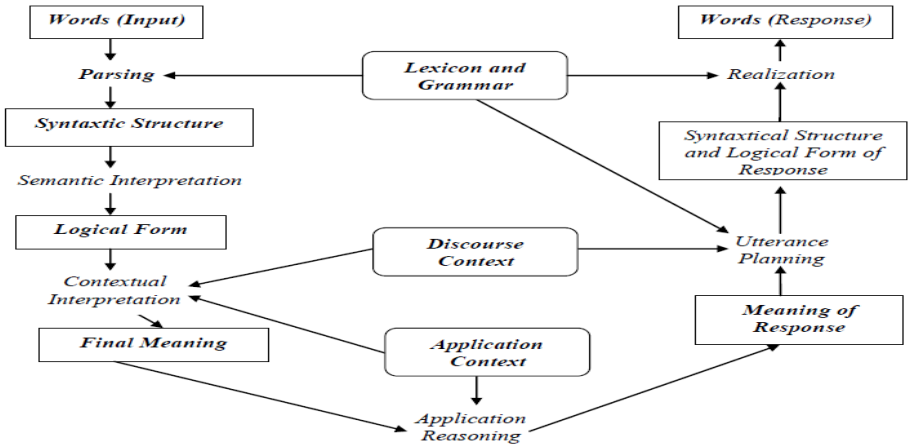
NLP tidak bertujuan untuk mentransformasikan bahasa yang diterima dalam bentuk teks atau suara menjadi data digital dan/atau sebaliknya pula; melainkan bertujuan untuk memahami arti dari kalimat yang diberikan dalam bahasa alami dan memberikan respon yang sesuai, misalnya dengan melakukan

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.

2.4.1 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.



Gambar II.3 Organisasi Sebuah Sistem NLP

2.4.2 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.

2.4.3 *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

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.

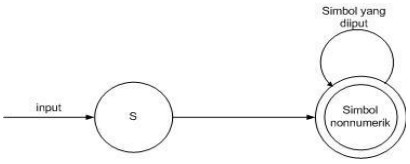
Dalam grammar, anggota alfabet dinamakan simbol terminal atau token. Kalimat adalah string yang tersusun atas simbol-simbol terminal. Bahasa adalah himpunan kalimat-kalimat. Anggota bahasa bisa berupa tak berhingga hingga kalimat. Simbol-simbol berikut adalah simbol terminal :

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 II.4 Diagram transisi

2.4.4 *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*.

2.4.4.1 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.

2.4.4.2 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.

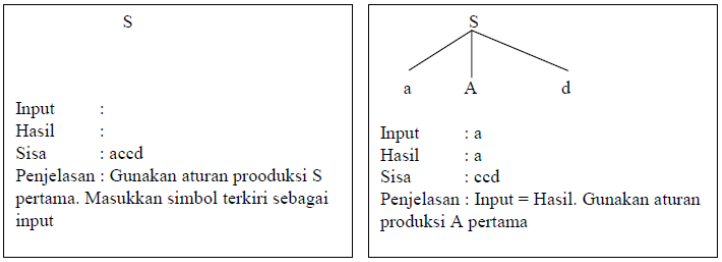
Contoh suatu bahasa dengan aturan produksi sebagai berikut:

S → aAd | aB

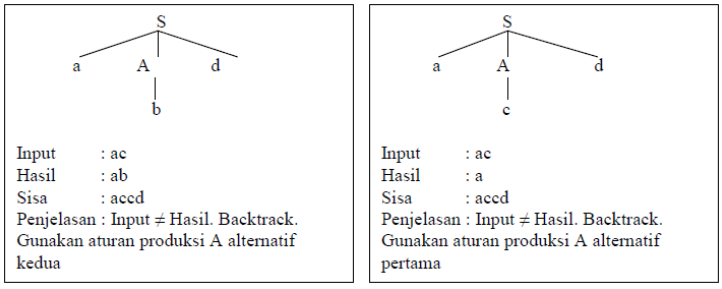
A → b | c

B → ccd | ddc

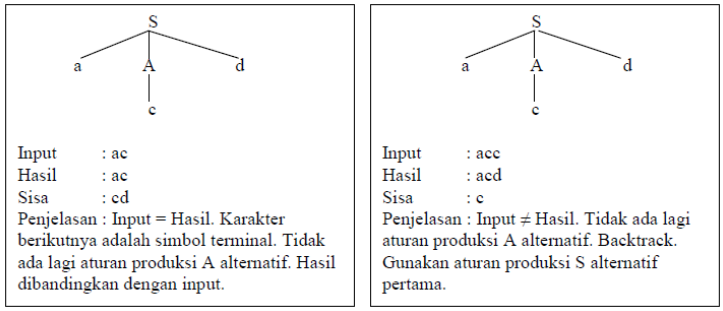
Misal akan dilakukan parsing untuk string "accd". Maka analisis sintaks terhadap string tersebut dengan menggunakan metode *Brute Force* adalah:



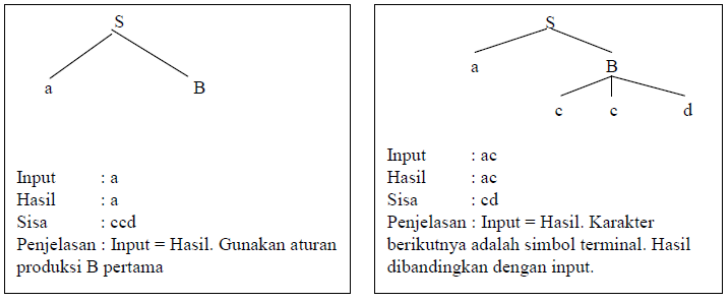
Gambar II.5 Metode *Bruce Force* 1



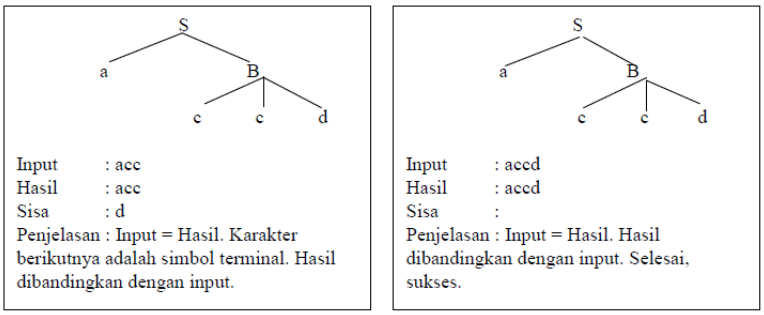
Gambar II.6 Metode *Bruce Force* 2



Gambar II.7 Metode *Bruce Force* 3



Gambar II.8 Metode *Bruce Force* 4



Gambar II.9 Metode *Bruce Force* 5

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.

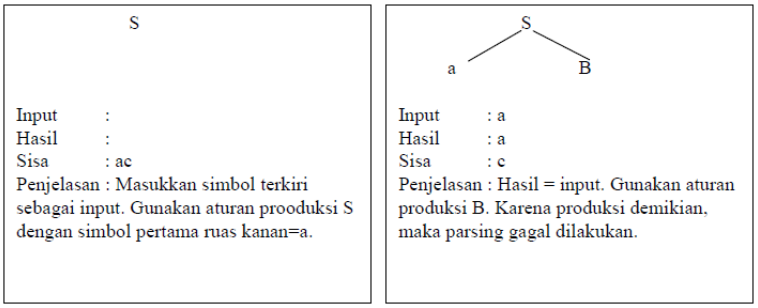
Jika tidak ada produksi yang demikian maka dikatakan bahwa parsing tidak dapat dilakukan. Ketentuan produksi yang digunakan metoda *recursive descent* adalah jika terdapat dua atau lebih produksi dengan ruas kiri yang sama maka karakter pertama dari semua ruas kanan produksi tersebut tidak boleh sama. 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 II.10 Metode *Recursive Descent*

2.4.5 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.

2.5 Teori Bahasa dan Automata

Teori bahasa dan automata merupakan cabang ilmu dari teori komputasi pada ilmu komputer. Beberapa teori komputasi berasal dari bahasa dan rekayasa sistem, tetapi kebanyakan dari matematika.

Finite state automata dan ekspresi reguler awalnya dikembangkan berdasarkan pemikiran *neural* network dan *switching circuit. Finite state automata* (FSA) merupakan *tool* yang sangat berguna dalam perancangan *lexical analiyzer*.

2.6 Konsep Dasar Analisis Sistem

2.6.1 Flowmap

Bagan alir atau *flowmap* adalah bagan yang menunjukan alir di dalam program atau prosedur sistem secara logika. Bagan alir digunakan terutama

untuk alat bantu komunikasi dan dokumentasi. Bagan alir dokumen atau sering disebut juga *Flow Map* merupakan bagan alir yang menunjukan arus dari aliran formulir termasuk tembusan-tembusannya.

2.6.2 ERD (*Entity Relationship Diagram*)

Entity Relationship Diagram atau biasa dikenal dengan diagram E-R secara grafis menggambarkan isi sebuah *database*. Diagram ini memiliki dua komponen utama yaitu *entity* dan relasi. Untuk melambangkan fungsi diatas maka digunakan simbol-simbol yang bisa dilihat pada daftar simbol.

Elemen-elemen *Entity Relationship Diagram* adalah sebagai berikut :

1. *Entity* (Entitas)

Pada E-R diagram, *entity* digambarkan dengan sebuah bentuk persegi panjang. *Entity* adalah sesuatu apa saja yang ada di dalam sistem, nyata maupun abstrak dimana data tersimpan.

Entitas diberi nama dengan kata benda dan dapat dikelompokan dalam empat jenis nama, yaitu : orang, benda, lokasi kejadian (terdapat unsur waktu di dalamnya).

2. Relationship (Relasi)

Pada E-R diagram, relationship dapat digambarkan dengan sebuah bentuk belah ketupat. Relationship adalah hubungan alamiah yang terjadi antara entitas. Pada umumnya relationship diberi nama dengan kata kerja dasar, sehinga memudahkan untuk melakukan pembacaan relasinya.

3. Atribut

Secara umum atribut adalah sifat atau karakteristik dari tiap entitas maupun tiap relationship. Maksudnya adalah sesuatu yang menjelaskan apa sebenarnya yang dimaksud entitas maupun relationship, sehingga sering dikatakan bahwa atribut adalah elemen dari setiap entitas dan relationship.

4. Kardinalitas

Kardinalitas relasi menunjukan jumlah maksimum tupel yang dapat berelasi dengan entitas yang lainnya. Dari sejumlah kemungkinan banyaknya hubungan yang terjadi dari entitas, kardinalitas relasi merujuk kepada hubungan maksimum yang terjadi dari entitas yang satu ke entitas yang lainnya dan begitu juga sebaliknya. Macam-macam kardinalitas relasi, yaitu :

a. One to one Relationship

Tingkat hubungann satu ke satu, dinyatakan dengan satu kejadian pada entitas pertama, hanya mempunyai satu hubungan dengan satu kejadian pada entitas yang kedua dan sebaliknya.



Gambar II.11 *One to One Relationship*

b. One to many Relationship

Tingkat hubungan satu ke banyak adalah untuk satu kejadian pada entitas yang pertama dapat mempunyai banyak hubungan dengan kejadian pada entitas yang kedua.



Gambar II.12 *One to Many Relationship*

c. Many To One Relationship

Untuk banyak kejadian pada entitas yang pertama hanya dapat mempunyai satu hubungan dengan kejadian pada entitas yang kedua.



Gambar II.13 *Many to One Relationship*

d. Many to many Relationship

Tingkat hubungan banyak ke banyak terjadi jika tiap kejadian pada sebuah entitas akan mempunyai banyak hubungan dengan kejadian pada entitas lainnya, baik dilihat dari sisi entitas yang pertama maupun dilihat dari sisi yang kedua. Relasi dari hubungan ini menghasilkan sebuah tabel baru yang disebut tabel hasil relasi.



Gambar II.14 *Many to Many Relationship*

5. *Key* (Kunci)

Sebuah atribut atau set atribut yang nilainya mengidentifikasikan entitas secara unik dalam suatu entitas. *Key* memiliki beberapa jenis sesuai dengan kegunaannya masing-masing, yaitu *primary key* (kunci utama), *foreign key* (kunci tamu).

2.6.3 Diagram Konteks

Diagram konteks merupakan arus data yang berfungsi untuk menggambarkan keterkaitan aliran-aliran data antar sistem dengan bagian luar

(kesatuan luar). Kesatuan luar ini merupakan sumber arus data atau tujuan data yang berhubungan dengan sistem informasi tersebut.

2.6.4 DFD (Data Flow Diagram)

Diagram Alir Data atau *Data Flow Diagram* (DFD) adalah suatu model yang menjelaskan arus data mulai dari pemasukan sampai dengan keluaran data. Tingkatan DFD dimulai dari diagram konteks yang menjelaskan secara umum suatu sistem atau batasan sistem aplikasi yang akan dikembangkan. Kemudian DFD dikembangkan menjadi DFD tingkat 0 atau level 0 dan kemudian DFD level 0 dikembangkan lagi menjadi level 1 dan selanjutnya sampai sistem tersebut tergambarkan secara rinci menjadi tingkatan-tingkatan lebih rendah lagi.

DFD merupakan penurunan atau penjabaran dari diagram konteks. Dalam pembuatan DFD harus mengacu pada ketentuan sebagai berikut :

1. Setiap penurunan level yang lebih rendah harus mempresentasikan proses tersebut dalam spesifikasi proses yang jelas.
2. Penurunan dilakukan apabila memang diperlukan.
3. Tidak semua bagian dari sistem harus ditunjukkan dengan jumlah level yang sama.

Simbol-simbol yang digunakan dalam Data Flow Diagram menurut notasi Yourdan adalah sebagai berikut :

1. Proses

Proses adalah simbol pertama data flow diagram. Proses dilambangkan dengan lingkaran, dimana proses ini menunjukan bagian dari sistem yang

mengubah satu atau lebih *input* dan *output*. Nama proses dituliskan dengan satu kata, singkatan atau kalimat sederhana. Lambang sebuah proses dapat dilihat pada gambar berikut

2. Aliran Data

Aliran Data digambarkan dengan tanda panah. Aliran data juga digunakan untuk menunjukan bagian-bagian informasi dari satu bagian ke bagian lain. Pembagian nama untuk aliran ini menunjukan sebuah arti untuk sebuah aliran. Untuk kebanyakan sistem yang dibuat, aliran data sebenarnya mengambarkan data yakni angka, huruf, pesan, floating point, dan macam-macam informasi lainnya.

3. Simpanan Data

Simpanan data digunakan sebagai penyimpanan bagi paket-paket data. Notasi penyimpanan data digambarkan dengan garis horizontal yang pararel. Simpanan data merupakan simpanan data dari data yang berupa suatu file atau *database* di sistem komputer ataupun berupa arsip atau catatan manual. Nama dari simpanan data menunjukan nama filenya.

4. Terminator

Terminator digambarkan dengan sebuah kotak yang menggambarkan kesatuan luar (*eksternal entitty*) yang berhungan dengan sistem. Kesatuan luar merupakan kesatuan (*entity*) dilingkungan luar sistem yang dapat berupa orang, Organisasi atau sistem lainnya yang berada di lingkungan luarnya yang akan memberikan *input* atau *output* dari sistem.

2.6.5 Kamus Data

Kamus data (KD) atau *data dictionary* (DD) atau disebut juga dengan istilah *sistems data dictionary* adalah catalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem informasi. Dengan menggunakan kamus data, analis sistem dapat mendefinisikan data yang mengalir di sistem dengan lengkap kamus data dibuat pada tahap perancangan sistem. Pada tahap analisis kamus data dapat digunakan sebagai alat komunikasi antara analisis sistem dengan pemakai sistem tentang informasi yang dibutuhkan oleh pemakai sistem. Pada tahap perancangan sistem kamus data digunakan untuk merancangan *input*, merancang laporan-laporan dan *database*.

Kamus data dibuat berdasarkan arus data yang ada di DFD. Arus data di DFD sifatnya adalah global, hanya ditunjukkan nama arus datanya saja. Keterangan lebih lanjut tentang struktur dari suatu arus data di DFD secara lebih terinci dapat dilihat di kamus data.

2.7 Konsep Dasar Basis Data

2.7.1 Data

Data adalah sumber informasi yang bentuknya masih mentah. Data pula menggambarkan suatu kejadian yang sedang terjadi, dimana data tersebut diolah dan diterapkan dalam sistem menjadi *input* yang berguna dalam suatu sistem. Secara rinci dari definisi data adalah sebagai berikut :

1. Data adalah penggambaran dari sesuatu dan kejadian yang kita hadapi.
2. Data bisnis adalah penggambaran dari suatu organisasi tentang sesuatu dan kejadian yang terjadi.

3. Data adalah kenyataan yang menggambarkan suatu kejadian-kejadian dan kesatuan nyata. Kejadian adalah sesuatu yang terjadi pada saat tertentu. Sebagai contoh, dalam dunia bisnis kejadian-kejadian nyata yang sering terjadi adalah perubahan dari suatu nilai yang disebut dengan transaksi.
4. Data yang telah diolah menjadi bentuk yang lebih berguna dan lebih berarti bagi yang menerimanya.
5. Sesuatu yang nyata dapat mengurangi derajat ketidakpastian tentang sesuatu keadaan atau kejadian.

2.7.2 Basis Data

Basis data adalah kumpulan informasi yang disimpan di dalam komputer secara sistematik sehingga dapat diperiksa menggunakan suatu program komputer untuk memperoleh informasi dari basis data tersebut. Perangkat lunak yang digunakan untuk mengelola dan memanggil kueri (*query*) basis data disebut sistem manajemen basis data (*database management system*, DBMS).

Basis data digunakan karena memiliki keuntungan sebagai berikut:

1. Mengurangi redundansi
2. Data dapat di-*share* antar aplikasi
3. Dapat dilakukan standardisasi data
4. Batasan *security* dapat diterapkan
5. Mengelola integritas data (akurasinya terjamin)
6. Independensi data (objektif DBS), basis data dapat berkembang tanpa mempengaruhi aplikasi yang telah ada

Secara definitif, basis data merupakan suatu objek terstruktur. Objek terstruktur tersebut terdiri atas data dan *metadata*. Data pada basis data merupakan informasi deskriptif yang benar-benar tersimpan, misalnya ‘Nama’ atau ‘Alamat’. Sedangkan metadata merupakan bagian yang menjelaskan tentang struktur data tersebut dalam basis data, misalnya *field* untuk ‘Nama’ dan ‘Alamat’, panjang *field*, atau tipe data untuk masing-masing *field*.

Konsep dasar dari basis data adalah kumpulan dari catatan-catatan, atau potongan dari pengetahuan. Sebuah basis data memiliki penjelasan terstruktur dari jenis fakta yang tersimpan di dalamnya, penjelasan ini disebut skema. Skema menggambarkan obyek yang diwakili suatu basis data, dan hubungan di antara obyek tersebut.

Ada banyak cara untuk mengorganisasi skema, atau memodelkan struktur basis data ini dikenal sebagai model basis data atau model data. Model yang umum digunakan sekarang adalah model relasional, yang menurut istilah layman mewakili semua informasi dalam bentuk tabel-tabel yang saling berhubungan dimana setiap tabel terdiri dari baris dan kolom. Model yang lain seperti model hierarkis dan model jaringan menggunakan cara yang lebih eksplisit untuk mewakili hubungan antar tabel. Istilah basis data mengacu pada koleksi dari data-data yang saling berhubungan, dan perangkat lunaknya seharusnya mengacu sebagai sistem manajemen basis data (*database management system/DBMS*).

Basis Data (*Database)* dapat dibayangkan sebagai sebuah lemari arsip yang ditempatkan secara berurutan untuk memudahkan dalam pengambilan kembali data tersebut. Basis Data menunjukkan suatu kumpulan data yang

dipakai dalam suatu lingkungan perusahaan atau instansi-instansi. Penerapan basis data dalam sistem informasi disebut sistem basis data (*database sistem).*

Jadi dapat disimpulkan bahwa Basis Data merupakan kumpulan data yang (arsip) yang saling berhubungan yang disimpan secara bersama sedemikian rupa dan tanpa pengulangan (*redudansi*) yang tidak perlu, untuk memenuhi berbagai kebutuhan. Diartikan sebagai kumpulan file/ tabel/ arsip yang saling berhubungan yang disimpan dalam media penyimpanan elektronis.

2.7.3 Normalisasi

Normalisasi adalah suatu teknik untuk mengorganisasi data ke dalam tabel-tabel untuk memenuhi kebutuhan pemakai di dalam suatu organisasi.

2.7.3.1 Tujuan Normalisasi

1. Menghilangkan kerangkapan data
2. Mengurangi kompleksitas
3. Mempermudah pemodifikasian data

2.7.3.2 Proses Normalisasi

Data diuraikan dalam bentuk tabel, selanjutnya dianalisis berdasarkan persyaratan tertentu ke beberapa tingkat. Apabila tabel yang diuji belum memenuhi persyaratan tertentu, maka tabel tersebut perlu dipecah menjadi beberapa tabel yang lebih sederhana sampai memenuhi bentuk yang optimal.

2.7.3.3 Tahapan Normalisasi

1. Bentuk Tidak Normal
2. Menghilangkan perulangan group
3. Bentuk Normal Pertama (1NF)

4. Menghilangkan ketergantungan sebagian
5. Bentuk Normal Kedua (2NF)
6. Menghilangkan ketergantungan transitif
7. Bentuk Normal Ketiga (3NF)
8. Menghilangkan anomali-anomali hasil dari ketergantungan fungsional
9. Bentuk Normal Boyce-Codd (BCNF)
10. Menghilangkan Ketergantungan Multivalue
11. Bentuk Normal Keempat (4NF)
12. Menghilangkan anomali-anomali yang tersisa
13. Bentuk Normal Kelima

2.7.4 Ketergantungan Fungsional

Atribut Y pada relasi R dikatakan tergantung fungsional pada atribut X (R. X→R. Y), jika dan hanya jika setiap nilai X pada relasi R mempunyai tepat satu nilai Y pada R.

2.7.5 *Database* Management System (DBMS)

Database Management System (DBMS) adalah suatu sistem perangkat lunak yang digunakan untuk memanipulasi atau memproses basis data. Sedangkan istilah *relational database management system* digunakan untuk menyebut suatu perangkat lunak yang dapat menangani basis data relasional dan berkomunikasi dengan *engine* basis data tersebut .

Diperlukan suatu sistem untuk diintegrasikan data file kedalam suatu file sehingga bias melayani berbagai *user* yang berbeda. Perangkat keras serta

prosedur yang mengelola *database* merupakan suatu *database* manajemen sistem. DBMS memungkinkan untuk memebentuk dan meremajakan file-file, memilih, mendatakan dan menyortir data, dan untuk menghasilkan laporan-laporan.

2.8 Perangkat Lunak Pendukung

Perangkat lunak pendukung yang digunakan pada pengembangan aplikasi pembantu pengolahan pola makan berdasarkan kebutuhan gizi (anamanesa gizi), sebagai berikut

2.8.1 Borland Delphi 7.0

Delphi merupakan generasi lanjutan dari Turbo Pascal yang telah dikenal sebagai bahasa pemrograman yang terstruktur yang diluncurkan oleh *Borland International Incoorporation* pada tahun 1983.

Borland Delphi memiliki kelebihan dibandingkan dengan aplikasi pemrograman visual berbasis windows yang lain diantaranya :

a. Borland Delphi menyediakan fasilitas yang luas mulai dari fungsi membuat form hingga untuk menggunakan format file berbasis data yang popular seperti Dbase dengan paradoks.
b. Dalam Borland Delphi template aplikasi dan template format yang dapat digunakan untuk membuat semua aplikasi dengan lebih cepat.
c. Borland Delphi dapat diatur sesuai dengan kebutuhan dan menunjukkan bagaimana memiliki Borland Delphi di lingkungan dengan pekerjaan lebih produktif.
d. Program terkompilasi dan windows menyatakan dapat mengkompilasi program tetapi sebenarnya hanya dapat mengkompilasi dan kemudian

menghubungkan interpreter dan program kode dalam sebuah file sehingga didapat eksekusi yang lambat.

e. Kelebihan dalam menggunakan Delphi 7.0 yaitu Delphi 7.0 merupakan bahasa pemrograman dengan fasilitas-fasititas yang menjadikannya memiliki struktur dan format yang lebih efisien dan efektif untuk pemrograman sehingga dapat dengan mudah membuat suatu aplikasi yang sesuai dengan apa yang kita butuhkan.

Delphi memiliki beberapa kelebihan yang tidak dimiliki oleh bahasa lain, diantaranya :

1. *Form* dan komponen-komponennya dapat dipakai ulang dan dikembangkan.
2. Tersedia *application template* dan *form template*.
3. Memikili lingkungan pengembangan visual yang dapat diatur sesuai kebutuhan.
4. Menghasilkan file terkompilasi yang berjalan lebih cepat
5. Kemampuan mengakses data dari bermacam-macam format.

2.8.2 MySQL

MySQL adalah sebuah perangkat lunak sistem manajemen basis data SQL (*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 dibawah lisensi GNU General Public License (GPL), tetapi mereka juga menjual dibawah lisensi komersial untuk kasus-kasus

dimana penggunaannya tidak cocok dengan penggunaan GPL. Pada Gambar 2.15 tampak logo MySQL.



Gambar II.15 Logo MySQL

MySQL adalah *Relational Database Management System* (RDBMS) yang didistribusikan secara gratis dibawah lisensi GPL (General Public License). Dimana setiap orang bebas untuk menggunakan MySQL, namun tidak boleh dijadikan produk turunan yang bersifat *closed source* atau komersial. MySQL sebenarnya merupakan turunan salah satu konsep utama dalam *database* sejak lama, yaitu SQL (*Structured Query Language*).

SQL adalah sebuah konsep pengoperasian *database*, terutama untuk pemilihan atau seleksi dan pemasukan data, yang memungkinkan pengoperasian data dikerjakan dengan mudah secara otomatis. Keandalan suatu sistem *database* (DBMS) dapat diketahui dari cara kerja optimizer-nya dalam melakukan proses perintah-perintah SQL, yang dibuat oleh *user* maupun program-program aplikasinya. Sebagai *database* server, MySQL dapat dikatakan lebih unggul dibandingkan *database* server lainnya dalam query data. Hal ini terbukti untuk query yang dilakukan oleh *single user*, kecepatan query MySQL bisa sepuluh kali lebih cepat dari PostgreSQL dan lima kali lebih cepat dibandingkan interbase. Selain itu MySQL juga memiliki beberapa keistimewaan, antara lain sebagai berikut:

1. Portability

   MySQL dapat berjalan stabil pada berbagai sistem operasi seperti Windows, Linux,FreeBSD, Mac Os X Server, Solaris, Amiga, dan lainnya.

2. *OpenSource*

   MySQL didistribusikan secara *opensource* (gratis), dibawah lisensi GPL sehingga dapat digunakan secara cuma-cuma.

3. *Multiuser*

   MySQL dapat digunakan oleh beberapa *user* dalam waktu yang bersamaan tanpa mengalami masalah atau konflik.

4. *Performance tuning*

   MySQL memiliki kecepatan yang menakjubkan dalam menangani query sederhana, dengan kata lain dapat memproses lebih banyak SQL per satuan waktu.

5. *Column types*

   MySQL memiliki tipe kolom yang sangat kompleks, seperti signed/ unsigned integer, float, double, char, text, date, timestamp, dan lain-lain.

6. *Command dan functions*

   MySQL memiliki operator dan fungsi secara penuh yang mendukung perintah Select dan Where dalam query.

7. *Security*

   MySQL memiliki beberapa lapisan sekuritas seperti level subnetmask, nama host, dan izin akses *user* dengan sistem perizinan yang mendetail serta password terenkripsi.

8. *Scalability dan limits*

   MySQL mampu menangani *database* dalam skala besar, dengan jumlah records lebih dari 50 juta dan 60 ribu tabel serta 5 milyar baris. Selain itu batas indeks yang dapat ditampung mencapai 32 indeks pada tiap tabelnya.

9. *Connectivity*

   MySQL dapat melakukan koneksi dengan client menggunakan protokol TCP/IP, Unix soket (UNIX), atau Named Pipes (NT).

10. *Localisation*

    MySQL dapat mendeteksi pesan kesalahan pada client dengan menggunakan lebih dari dua puluh bahasa. Meskipun demikian, bahasa Indonesia belum termasuk didalamnya.

11. *Interface*

    MySQL memiliki *interface* (antar muka) terhadap berbagai aplikasi dan bahasa pemrograman dengan menggunakan fungsi API (Application Programming *Interface*).

12. *Clients dan tools*

    MySQL dilengkapi dengan berbagai tool yang dapat digunakan untuk administrasi *database*, dan pada setiap tool yang ada disertakan petunjuk online.

13. *Struktur tabel*

    MySQL memiliki struktur tabel yang lebih fleksibel dalam menangani ALTER *TABEL*, dibandingkan *database* lainnya semacam PostgreSQL ataupun Oracle.