

BAB II **LANDASAN TEORI**

2.1 Logika Fuzzy

Sri Kusumadewi (2010) menerangkan bahwa Logika *fuzzy* diperkenalkan oleh Prof. Lotfi Astar Zadeh pada 1962. Logika *fuzzy* adalah metodologi sistem kontrol pemecahan masalah, yang cocok untuk di implementasikan pada sistem, mulai dari sistem yang sederhana, kecil, jaringan PC, dan sistem kontrol. Metodologi ini dapat diterapkan pada perangkat keras, perangkat lunak, atau kombinasi keduanya. Dalam logika klasik dinyatakan bahwa segala sesuatu bersifat biner, yang artinya adalah hanya mempunyai kemungkinan, “Ya atau Tidak”, “Benar atau Salah”, “Baik atau Buruk”, dan lain-lain. Oleh karena itu, semua ini dapat mempunyai nilai keanggotaan 0 dan 1. Akan tetapi, dalam logika *fuzzy* memungkinkan nilai keanggotaan berada di antara 0 dan 1. Artinya bisa saja suatu keadaan mempunyai dua nilai “Ya dan Tidak”, “Benar dan Salah”, “Baik dan Buruk” secara bersamaan, namun besar nilainya tergantung pada bobot keanggotaan yang dimilikinya. Logika *fuzzy* dapat digunakan di berbagai bidang, seperti pada sistem diagnosis penyakit (dalam bidang kedokteran); Pemodelan sistem pemasaran, riset operasi (dalam bidang ekonomi); kendali kualitas air, prediksi adanya gempa bumi, Klasifikasi dan pencocokan pola (dalam bidang teknik).

Pada teori himpunan *fuzzy*, peranan derajat keanggotaan sebagai penentu keberadaan elemen dalam suatu himpunan sangatlah penting. Nilai keanggotaan atau derajat keanggotaan atau *membership function* menjadi ciri utama dari penalaran dengan logika *fuzzy* tersebut

Sri Kusumadewi (2010) menyatakan konsep logika *fuzzy* kemudian berhasil diaplikasikan dalam bidang kontrol oleh Sugeno. Sejak saat itu aplikasi *fuzzy* berkembang kian pesat. Di tahun 1985an negara Jepang dan negara – negara di Eropa secara *agresif* membangun produk nyata sehubungan dengan konsep logika *fuzzy* yang diintegrasikan dalam produk-produk kebutuhan rumah tangga seperti *vacum cleaner*, *microwave oven* dan kamera video. Logika *fuzzy* berkembang pesat selama beberapa tahun terakhir. Terdapat lebih dari dua ribu produk dipasaran yang menggunakan konsep logika *fuzzy*, mulai dari mesin cuci

hingga kereta berkecepatan tinggi. Setiap aplikasi tentunya menyadari beberapa keuntungan dari logika *fuzzy* seperti performa, kesederhaan, biaya rendah dan produktifitasnya. Logika *fuzzy* adalah suatu cara yang tepat untuk memetakan suatu ruang *input* ke dalam suatu ruang *output*.

Kelebihan logika *fuzzy* adalah kemampuan dalam proses penalaran secara bahasa sehingga dalam perancangannya tidak memerlukan matematik yang rumit. Beberapa alasan yang dapat di utarakan mengapa kita menggunakan logika *fuzzy* diantaranya adalah mudah dimengerti, memiliki toleransi terhadap data-data yang tidak tepat, mampu memodelkan fungsi-fungsi nonlinear yang sangat kompleks, dapat membangun dan mengaplikasikan pengalaman-pengalaman para pakar secara langsung tanpa harus melalui proses pelatihan, dapat bekerja sama dengan teknik-teknik kendali secara konvensional, dan didasarkan pada bahasa alami.

Pada himpunan *fuzzy*, nilai keanggotaan terletak pada rentang 0 sampai 1. Keanggotaan *fuzzy* memberikan suatu ukuran terhadap pendapat atau keputusan, sedangkan probabilitas mengindikasikan proporsi terhadap keseringan suatu hasil bernilai benar dalam jangka panjang. Misalnya, jika nilai keanggotaan suatu himpunan *fuzzy* MUDA adalah 0,9 maka tidak perlu dipermasalahkan berapa seringnya nilai itu diulang secara individual untuk mengharapkan suatu hasil yang hampir pasti muda. Di lain pihak, nilai probabilitas 0,9 muda berarti 10% dari himpunan tersebut diharapkan tidak muda.

Himpunan *fuzzy* memiliki 2 atribut, yaitu:

- a) *Linguistik* yaitu penamaan suatu grup yang mewakili suatu keadaan atau kondisi tertentu dengan menggunakan bahasa alami seperti : MUDA, PAROBAYA, TUA.
- b) *Numeris* yaitu suatu nilai angka yang menunjukkan ukuran dari suatu variabel seperti : 10, 35, 40.

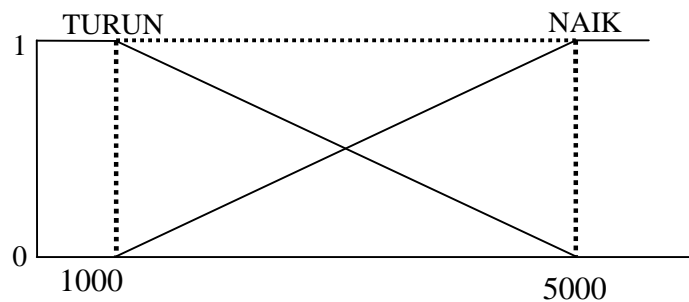
Ada beberapa hal yang perlu diperhatikan dalam sistem *fuzzy* yaitu :

- a) Variabel *Fuzzy* :
Merupakan variabel yang hendak dibahas dalam suatu sistem *fuzzy*.
contoh : penghasilan, temperatur, permintaan, umur, dan sebagainya.

b) Himpunan *Fuzzy*

Merupakan suatu kelompok yang mewakili suatu keadaan tertentu dalam variabel *fuzzy*. contoh ;

Variabel permintaan , terbagi menjadi 2 himpunan *fuzzy*, yaitu :
TURUN dan NAIK.



Gambar 2.1 Variabel permintaan

c. Semesta Pembicaraan

Merupakan keseluruhan nilai yang diperbolehkan untuk dioperasikan dalam suatu variabel *fuzzy*. Semesta pembicaraan merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai semesta pembicaraan dapat berupa bilangan positif maupun negatif. Adakalanya nilai semesta pembicaraan ini tidak dibatasi batas atasnya.

Contoh :

Semesta pembicaraan untuk variabel umur $[0 + \infty]$

Semesta pembicaraan untuk variabel temperatur $[-10 \ 90]$

a) Domain

Domain himpunan *fuzzy* adalah keseluruhan nilai yang diijinkan dalam semesta pembicaraan dan boleh dioperasikan dalam suatu himpunan *fuzzy*. Seperti halnya semesta pembicaraan, domain merupakan himpunan bilangan real yang senantiasa naik (bertambah) secara monoton dari kiri ke kanan. Nilai domain dapat berupa bilangan positif maupun negatif.

Contoh domain himpunan *fuzzy* :

TURUN : $[0 \ 5000]$

NAIK : $[1000 + \infty]$

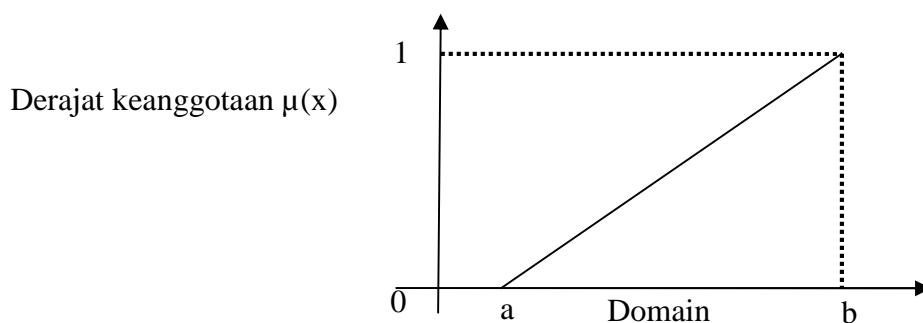
2.2 Fungsi Keanggotaan

Fungsi keanggotaan adalah suatu kurva yang menunjukkan pemetaan titik-titik *input* ke dalam nilai keanggotaan yang memiliki interval 0 sampai 1. Derajat keanggotaan sebuah variable x dilambangkan dengan simbol $\mu(x)$. Rule-rule menggunakan nilai keanggotaan sebagai faktor bobot untuk menentukan pengaruhnya pada saat melakukan inferensi untuk menarik kesimpulan. Ada beberapa fungsi keanggotaan yang sering digunakan, diantaranya adalah :

2.2.1 Grafik keanggotaan kurva linear

Pada grafik keanggotaan linear, sebuah variabel input dipetakan kederajat keanggotaannya dengan digambarkan sebagai suatu garis lurus.

Ada 2 grafik keanggotaan linear. Pertama, grafik keanggotaan kurva linear naik, yaitu kenaikan himpunan *fuzzy* dimulai pada nilai domain yang memiliki derajat keanggotaan nol [0] bergerak kekanan menuju ke nilai domain yang memiliki derajat keanggotaan lebih tinggi.

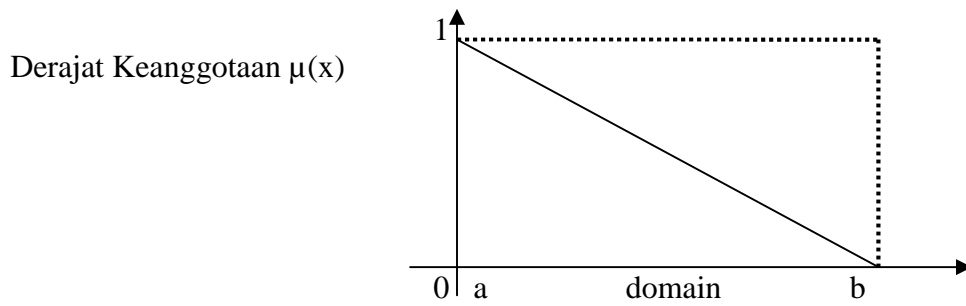


Gambar 2.2 Grafik Keanggotaan Kurva Linier Naik

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0 ; x \leq a \\ \frac{(x-a)}{(b-a)} ; a \leq x \leq b \\ 1 ; x = b \end{cases} \dots\dots\dots(2.1)$$

Kedua, Garis lurus dimulai dari nilai domain dengan derajat keanggotaan tertinggi pada sisi kiri, kemudian bergerak menurun ke nilai domain yang memiliki derajat keanggotaan lebih rendah



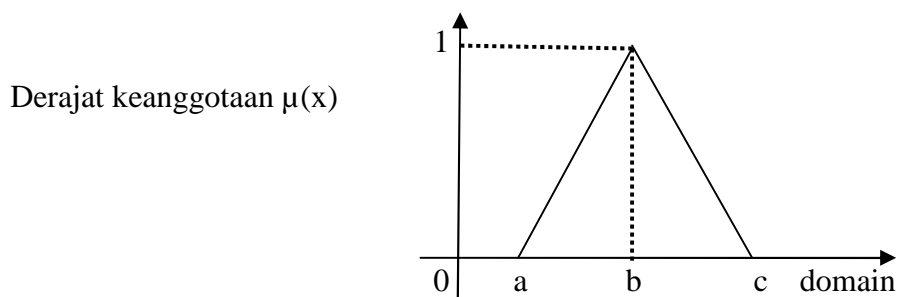
Gambar 2.3 Grafik Keanggotaan Kurva Linier Turun

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x > b \\ \frac{(b-x)}{(b-a)}; & a < x < b \\ 1; & x = a \end{cases} \dots\dots\dots(2.2)$$

2.2.2 Grafik keanggotaan kurva segitiga

Kurva segitiga pada dasarnya merupakan gabungan antara 2 garis (linier) seperti pada gambar 2.4 dibawah ini :



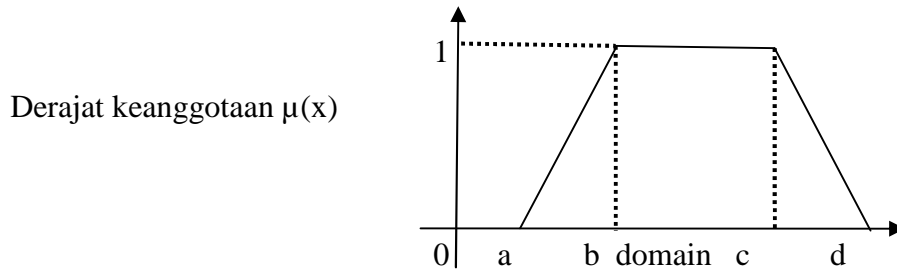
Gambar 2.4 Grafik Keanggotaan Kurva Segitiga

Fungsi Keanggotaan:

$$\mu[x] = \begin{cases} 0; & a < x < c \\ \frac{(x-a)}{(b-a)}; & a < x < b \\ \frac{(c-x)}{(c-b)}; & b < x < c \\ 1; & x = b \end{cases} \dots\dots\dots(2.3)$$

2.2.3 Grafik keanggotaan kurva trapesium

Kurva trapesium pada dasarnya seperti bentuk segitiga, hanya saja pada rentan tertentu ada beberapa titik yang memiliki nilai anggota 1



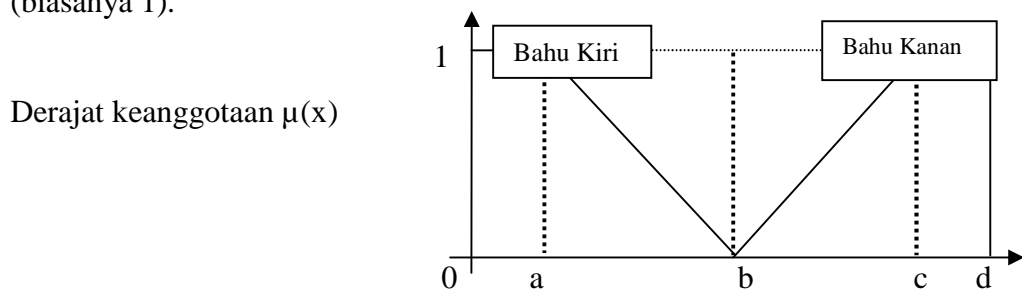
Gambar 2.5 Grafik Keanggotaan Kurva Trapesium

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 0; & x : a \text{ atau } x > d \\ \frac{(x-a)}{(b-a)}; & a \leq x \leq b \\ 1; & b \leq x \leq c \\ \frac{(c-x)}{(c-b)}; & c \leq x \leq d \end{cases} \dots\dots\dots(2.4)$$

2.2.4 Grafik keanggotaan kurva bentuk bahu

Grafik keanggotaan kurva bentuk bahu digunakan untuk mengakhiri variabel suatu daerah fuzzy yang nilai derajat keanggotaannya adalah konstan (biasanya 1).



Gambar 2.6 Grafik keanggotaan kurva bahu

Fungsi Keanggotaan :

$$\mu[x] = \begin{cases} 1; & 0 \leq x \leq a \text{ atau } c \leq x \leq d \\ \frac{(b-x)}{(b-a)}; & a \leq x \leq b \\ \frac{(x-b)}{(c-b)}; & b \leq x \leq c \end{cases} \dots\dots\dots(2.5)$$

2.3 Operasi Himpunan Fuzzy

Operasi himpunan *fuzzy* diperlukan untuk proses inferensi atau penalaran, dalam hal ini yang dioperasikan adalah derajat keanggotaan sebagai hasil dari operasi dua buah himpunan *fuzzy* disebut sebagai *fire strength* atau -predikat. Berikut beberapa operasi dasar yang paling sering digunakan untuk mengombinasikan dan memodifikasi himpunan *fuzzy*.

1. Operasi Gabungan (*Union*)

Operasi ini sering disebut operator OR dari himpunan fuzzy A dan B dinyatakan sebagai $A \cup B$. Dalam sistem logika *fuzzy*, Operasi gabungan disebut sebagai *Max*. Operasi *Max* ditulis dengan persamaan berikut.

$$\mu_{A \cup B}(x) = \max\{\mu_A(x), \mu_B(x)\} \text{ untuk setiap } x \in X \dots\dots\dots(2.6)$$

Derajat keanggotaan setiap unsur himpunan *fuzzy* $A \cup B$ adalah derajat keanggotaannya pada himpunan *fuzzy* A atau B yang memiliki nilai terbesar.

2. Operasi Irisan (*Intersection*)

Operasi irisan juga disebut operator AND dari himpunan fuzzy A dan B dinyatakan sebagai $A \cap B$. Dalam sistem logika fuzzy, operasi irisan disebut sebagai *Min*, operasi *Min* ditulis dengan persamaan berikut.

$$\mu_{A \cap B}(x) = \min\{\mu_A(x), \mu_B(x)\} \text{ untuk setiap } x \in X \dots\dots\dots(2.7)$$

Derajat keanggotaan setiap unsur himpunan *fuzzy* $A \cap B$ adalah derajat keanggotaannya pada himpunan *fuzzy* A atau B yang memiliki nilai terkecil.

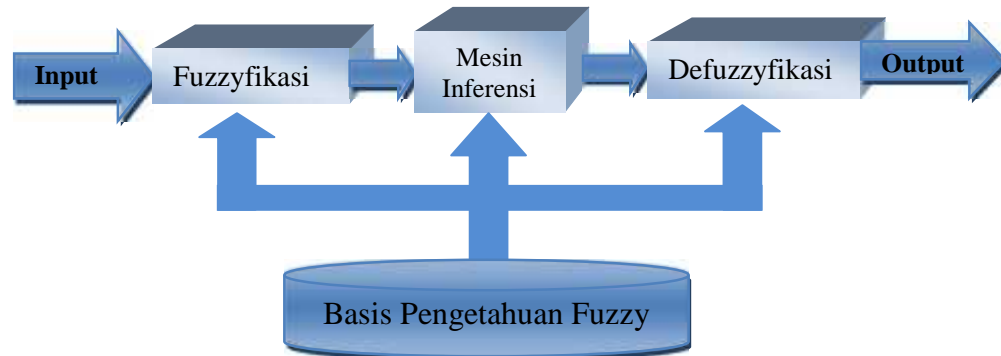
3. Operator Komplemen (*Complement*)

Bila himpunan *fuzzy* A pada himpunan universal X mempunyai fungsi keanggotaan $\mu_A(x)$ maka komplemen dari himpunan *fuzzy* A (disebut NOT) adalah himpunan *fuzzy* A^c dengan fungsi keanggotaan untuk setiap x elemen X.

$$\mu_{A^c}(x) = 1 - \mu_A(x) \dots\dots\dots(2.8)$$

2.4 Cara Kerja Logika Fuzzy

Berikut ini struktur elemen dasar *system inferensi fuzzy* :



Gambar 2.7 Struktur Sistem Inferensi Fuzzy

Keterangan :

- a. Basis pengetahuan *fuzzy* merupakan kumpulan rule-rule *fuzzy* dalam bentuk pernyataan IF...THEN.
- b. Fuzzyfikasi merupakan proses untuk mengubah *input* sistem yang mempunyai nilai tegas menjadi linguistic menggunakan fungsi keanggotaan yang di simpan dalam basis pengetahuan *fuzzy*.
- c. Mesin inferensi merupakan proses untuk mengubah *input fuzzy* menjadi *output fuzzy* dengan cara mengikuti aturan-aturan (IF-THEN Rules) yang telah ditetapkan pada basis pengetahuan *fuzzy*.
- c. Defuzzyfikasi adalah proses mengubah *output fuzzy* yang diperoleh dari mesin inferensi menjadi nilai tegas menggunakan fungsi keanggotaan yang sesuai dengan saat dilakukan fuzzyfikasi.

Cara kerja logika *fuzzy* meliputi beberapa tahapan berikut :

1. Fuzzyfikasi.
2. Pembentukan basis pengetahuan Fuzzy (Rule dalam bentuk IF..THEN).
3. Mesin inferensi (Fungsi implikasi *Max-Min* atau *Dot-Product*)
4. Defuzzyfikasi

Ada beberapa cara melakukan defuzzyfikasi, diantaranya metode berikut :

- a) Metode rata-rata (Average)

$$z^* = \frac{\mu_i z_i}{\sum \mu_i}$$

b) Metode titik tengah

$$z^* = \frac{\int \mu(z)zdz}{\int \mu(z)zdz}$$

2.5 Fuzzy Inference System Tsukamoto

Pada metode Tsukamoto, setiap aturan dipresentasikan menggunakan himpunan-himpunan *fuzzy* dengan fungsi keanggotaan yang monoton Untuk menentukan nilai *output crisp* hasil yang tegas (Z) dicari dengan cara mengubah input berupa himpunan *fuzzy* yang diperoleh dari komposisi aturan-aturan *fuzzy* menjadi suatu bilangan pada domain himpunan *fuzzy* tersebut. Cara ini disebut dengan metode *defuzzifikasi* (penegasan). Metode *defuzzifikasi* yang digunakan dalam metode Tsukamoto adalah metode *defuzzifikasi* rata-rata terpusat *Center Average Defuzzifier*.

Secara umum bentuk model fuzzy Tsukamoto adalah :

IF (X IS A) and (Y IS B) Then (Z IS C)

Dimana A, B, dan C adalah himpunan fuzzy.

Misalkan diketahui 2 rule berikut ;

IF (x is A_1) AND (y is B_1) THEN (z is C_1)

IF (x is A_2) AND (y is B_2) THEN (z is C_2)

Dalam inferensinya, metode Tsukamoto menggunakan tahapan berikut :

1. Fuzzyfikasi
2. Pembentukan basis pengetahuan fuzzy (Rule dalam bentuk IF...THEN).
3. Mesin inferensi

Menggunakan fungsi implikasi Min untuk mendapatkan nilai α -predikat tiap-tiap rule ($\alpha_1, \alpha_2, \alpha_3, \dots, \alpha_n$).

Kemudian masing-masing nilai α -predikat ini digunakan untuk menghitung keluaran hasil inferensi secara tegas (crisp) masing-masing rule ($z_1, z_2, z_3, \dots, z_n$).

4. Defuzzyfikasi

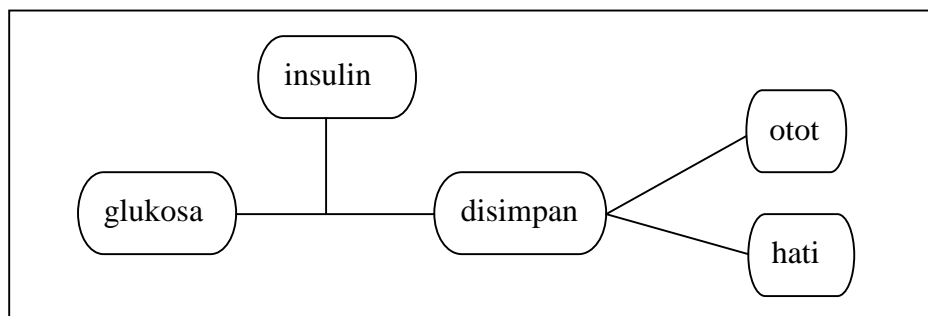
Menggunakan metode rata-rata (Average)
$$z = \frac{\sum a_i z_i}{\sum a_i}$$

2.6 Diabetes Melitus

Ulfa Nurahmi, S.Kep.,Ns (2012) Diabetes mellitus merupakan gangguan pankreas yang memiliki kadar gula darah yang tinggi.

Epie Suryono (2013) Pankreas merupakan sebuah kelenjar yang didalamnya terdapat sekumpulan sel yang disebut pulau langerhans. Pankreas menghasilkan dua hormon yang keduanya berfungsi untuk mengatur jumlah gula dalam darah, yakni hormon insulin dan glukagon.

Insulin berfungsi mengatur glukosa yang masuk kedalam tubuh agar tetap seimbang. Sebagian glukosa diubah menjadi glikogen kemudian disimpan dihati dan otot. Berikut ini gambar bagan kerja insulin :



Gambar 2.8 Bagan Kerja Insulin

Ulfa Nurahmi, S.Kep.,Ns (2012) Macam - macam gejala khas yang biasanya dirasakan oleh penderita diabetes yaitu :

1. Sering kali buang air kecil dengan volume yang banyak, yaitu lebih sering dari pada biasanya, khusus nya pada malam hari (poliuri).
2. Sering kali merasa haus dan ingin minum sebanyak-banyaknya (*Polidipsi*). Dengan banyaknya urine yang keluar, badan akan kekurangan air atau dehidrasi. Untuk mengatasi hal tersebut tubuh akan menimbulkan rasa haus sehingga Penderita diabetes ingin selalu minum terutama yang dingin, manis, segar, dalam jumlah banyak.
3. Nafsu makan meingkat (*Polifagi*) dan merasa kurang tenaga

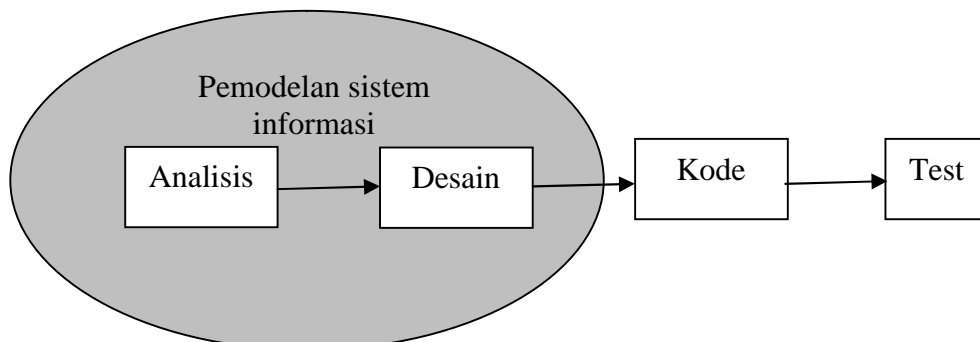
Diabetes terjadi karena insulin pada tubuh seseorang bermasalah, pemasukan gula kedalam sel-sel tubuh kurang sehingga energi yang di bentuk menjadi kurang. Inilah yang menyebabkan seseorang kekurangan tenaga. Salain itu, sel juga miskin gula sehingga otak juga berfikir bahwa kurang energi karena kurang makan, maka tubuh berusaha meningkatkan asupan makanan dengan menimbulkan rasa lapar.

4. Berat Badan turun dan menjadi kurus atau Berat badan bertambah dan menjadi gemuk.

Dalam sistem pembuangan urine, pada penderita diabetes tidak terkendali dapat kehilangan sebanyak 500 gram glukosa dalam urine per 24 jam atau setara dengan 2000 kalori perhari hilang dari tubuh, dan ini akan banyak mengurangi berat badan. namun jika ketika tubuh kekurangan kalori diiringi dengan konsumsi makanan dan minuman yang berlebihan sementara pankreas sedang dalam gangguan maka akan mengakibatkan badan menjadi gemuk karna glukosa yang seharusnya diolah menjadi energi, tidak terproses dan berubah menjadi lemak.

2.7 Model Sekuensial Linier

Roger S. Presman, Ph. D (2002) Menyatakan bahwa sekuensial linier untuk rekayasa perangkat lunak yang sering disebut juga dengan “ siklus kehidupan klasik “ atau model air terjun, model sekuensial linier adalah paradigma rekayasa perangkat lunak yang paling luas dipakai dan paling tua.



Gambar 2.9 Model sekuensial linier

Model sekuensial linier mengusulkan sebuah pendekatan kepada perkembangan perangkat lunak yang sistematis dan sekuensial yang mulai pada tingkat dan kemajuan sistem pada seluruh analisis, desain, kode, pengujian, dan pemeliharaan, dimodelkan setelah siklus rekayasa konvensional.

Model sekuensial linier melingkupi aktivitas-aktivitas berikut ini:

a. Rekayasa dan pemodelan sistem informasi

Karena perangkat lunak selalu menjadi bagian dari sebuah sistem yang lebih besar, kerja dimulai dengan membangun syarat dari semua elemen sistem dan mengalokasikan beberapa subset dari kebutuhan ke perangkat lunak tersebut, pandangan sistem ini penting ketika perangkat lunak harus berhubungan dengan elemen-elemen yang lain, seperti perangkat lunak, manusia dan database. Rekayasa dan analisis sistem menyangkut pengumpulan kebutuhan pada tingkat puncak. Rekayasa informasi mencakup juga pengumpulan kebutuhan pada tingkat bisnis strategis dan tingkat area bisnis.

b. Analisis kebutuhan perangkat lunak

Proses pengumpulan kebutuhan diintensifkan dan difokuskan, khususnya pada perangkat lunak (analisis) harus memahami domain informasi, tingkahlaku, unjuk kerja, dan antar muka(interface) yang diperlukan.

c. Desain

Desain perangkat lunak sebenarnya adalah proses multi langkah yang berfokus pada tempat atribut sebuah program yang berbeda, struktur data, arsitektur perangkat lunak, representasi interface, dan detail (algoritma) procedural. Proses desain menerjemahkan syarat kebutuhan kedalam sebuah representasi perangkat lunak yang dapat diperkirakan demi kualitas sebelum dimulai pemunculan kode. Sebagaimana persyaratan desain didokumentasikan menjadi bagian dari konfigurasi perangkat lunak.

d. Generasi kode

Desain harus diterjemahkan kedalam bentuk mesin yang bisa dibaca. Langkah pembuatan kode melakukan tugas ini. Jika desain dilakukan dengan cara yang lengkap, pembuatan kode dapat diselesaikan secara mekanis.

e. Pengujian

Perangkat lunak akan mengalami perubahan setelah disampaikan kepada pelanggan (pengecualian yang mungkin adalah perangkat lunak yang di lekatkan). Perubahan akan terjadi karena kesalahan-kesalahan dan memastikan bahwa input yang dibatasi akan memberikan hasil actual yang sesuai dengan hasil.

f. Pemeliharaan

ditentukan, karena perangkat lunak harus disesuaikan untuk mengakomodasi perubahan-perubahan didalam lingkungan eksternalnya (contohnya perubahan yang dibutuhkan sebagai akibat dari perangkat peripheral atau system operasi yang baru), atau karena pelanggan membutuhkan perkembangan fungsional atau unjuk kerja. Pemeliharaan perangkat lunak mengaplikasikan lagi setiap fase program sebelumnya dan tidak membuat yang baru lagi.

2.8 World Wide Web

Sutarman (2003) *World Wide Web* atau *WWW* adalah jaringan beribu-ribu computer yang dikategorikan menjadi dua : *Client* dan *Server* dengan menggunakan *Software* khusus membentuk sebuah jaringan yang disebut jaringan *client-server*. Dalam cara kerja dari *WWW* ada dua hal yang terpenting yaitu *software web server* dan *software web browser*.

Server menyimpan/menyediakan informasi dan memproses permintaan dari *client*, apabila ada *client* yang meminta informasi maka *server* mengirimkannya. Informasi yang diakses dapat berupa teks, gambar, suara. *Server* juga mengirimkan perintah-perintah ke *client* tentang bagaimana cara menampilkan semua informasi tersebut.

HTTP (HyperText Transfer Protocol) adalah suatu protokol yang menentukan aturan dan perlu diikuti oleh *web browser* dalam meminta atau mengambil suatu dokumen dan oleh *web server* dalam menyediakan dokumen yang diminta *web browser*. Protokol ini merupakan protokol standar yang digunakan untuk mengakses dokumen html.

sutarman (2003) Protokol transfer adalah suatu protokol yang digunakan untuk pengiriman informasi di internet. HTTP adalah protokol standar untuk suatu dokumen *web*. Selain HTTP di internet juga dikenal beberapa protokol lain diantaranya:

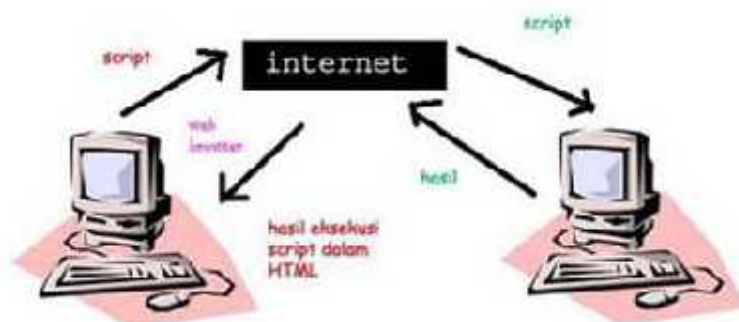
1. FTP (*File Transfer Protocol*), protokol ini dirancang untuk memungkinkan pemakaian dalam hal transfer *file* dalam format text atau binary dalam suatu *server* komputer di internet.
2. Gopher, protocol ini dirancng untuk mengakses *server* gopher yang menyediakan informasi dengan menggunakan suatu system menu atau melalui hubungan telnet.
3. News NNTP (*Network News Transfer Protokol*) adalah protocol yang digunakan untuk mendistribusikan berita diUSENet. USENet adalah suatu sisem yang dirancang sebagai forum diskusi dengan berdasarkan pada topic-topik yang disebut *newsgroup*.

Web adalah fasilitas *hypertext* untuk menampilkan data berupa teks, gambar, suara, animasi dan data multimedia lainnya.

Pada pemrograman *website* dapat dikategorikan menjadi 2 kategori :

1. *Server – side Programming*

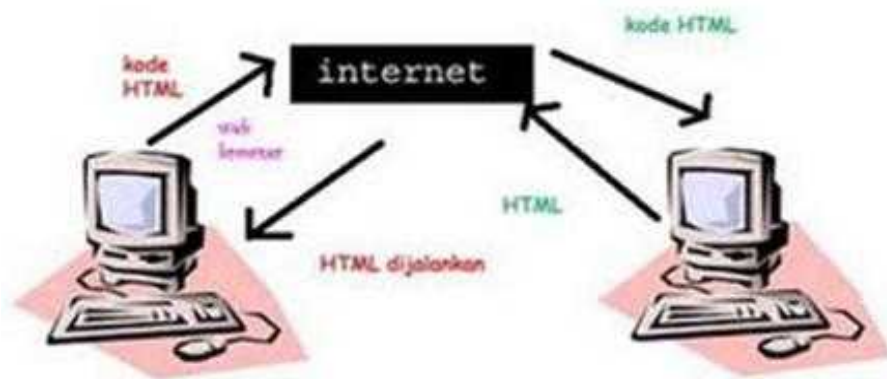
Pada *server-side programming* perintah-perintah program dijalankan di *web server* kemudian hasilnya dikirimkan ke browser dalam bentuk HTML. Pada gambar 2.9 ditunjukkan pemrograman pada sisi.



Gambar 2.10 Server-Side Programming

2. Client – side Programming

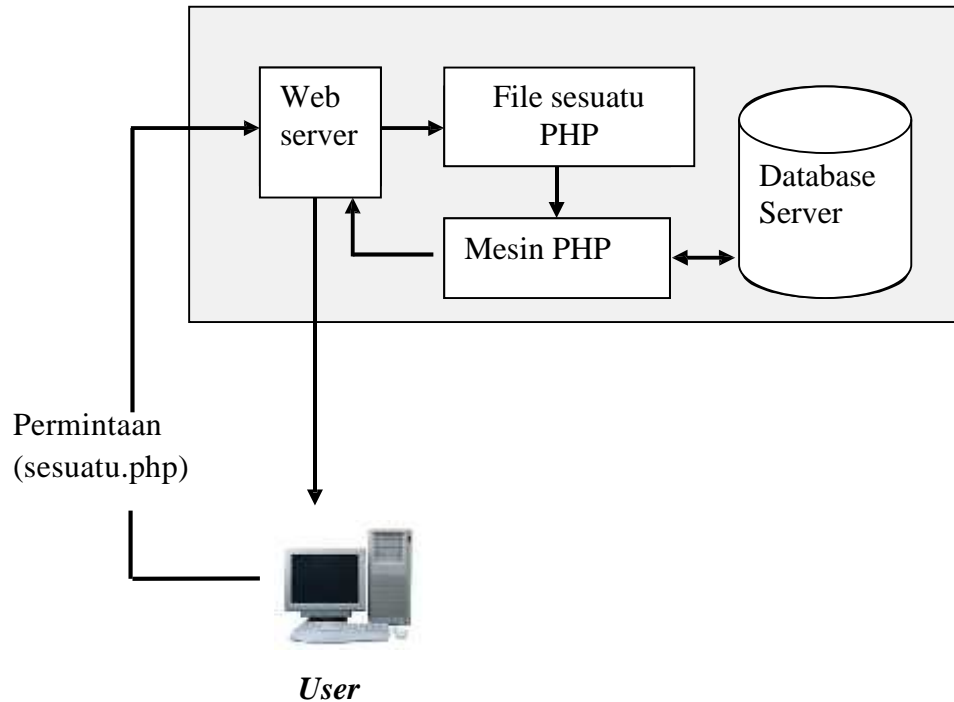
Sedangkan *client-side programming*, perintah-perintah program dijalankan di *web browser* sehingga ketika *client* meminta dokumen yang mengandung *script* maka *script* tersebut akan di download dari *server*-nya kemudian dijalankan di browser yang bersangkutan. Pada gambar 2.11 ditunjukkan ilustrasi pemrograman di sisi *client*.



Gambar 2.11 Client – Side Programming

2.9 PHP (Hypertext PreProcessor)

Php merupakan bahasa web serverside yang bersifat open source. Bahasa Php menyatu dengan script HTML yang sepenuhnya dijalankan pada server. File yang hanya berisi kode HTML yang dirancang tidak mendukung pembuatan aplikasi yang melibatkan database karena HTML dirancang untuk menyajikan informasi yang bersifat statis (tampilan yang isinya tetap hingga web master atau penanggung jawab web melakukan perubahan isi). Oleh karena itu, muncul pemikiran untuk membuat suatu perantara yang memungkinkan aplikasi bisa menghasilkan sesuatu yang bersifat dinamis dan berinteraksi dengan database. Akhirnya lahirlah sebagai perantara seperti PHP, ASP, dan JSP. Gambar 2.12 memperlihatkan skema yang memungkinkan aplikasi berinteraksi dengan database menggunakan PHP.



Gambar 2.12 Pemanggilan Aplikasi Web Bertipe PHP

Pada gambar 2.12, setelah *web server* menemukan *file* yang diminta *user* (*sesuatu.php*), *file* tersebut diserahkan ke mesin PHP untuk diproses. Bila PHP mendeteksi adanya interaksi dengan *database* maka PHP akan melakukan permintaan pada *database server* dan hasil dari *database server* diproses lebih lanjut. Setelah semua isi *file* diproses, maka hasilnya berupa diserahkan ke *web server* yang selanjutnya *web server* mengirimkan kode HTML kepada *user*.

Kode sumber PHP (*sesuatu.php*) tidak akan diketahui oleh *user* karena *user* hanya menerima kode hasil pemrosesannya. Dengan cara seperti ini, kerahasiaan kode sumber bisa terjaga. Tidak perlu ada kekhawatiran bahwa pemakai bisa melihat *password* yang digunakan untuk mengakses *database*.

