BAB II

LANDASAN TEORI

2.1 Ilmu Biometrik

Dalam ilmu biometrik, kata biometrik berasal dari bahasa Yunani yaitu *bios* = hidup dan *metron* = ukuran, menurut Kamus Besar Bahasa Indonesia, terbitan Balai Pustaka, 1986. "Biometri adalah pengukuran yang dilakukan dalam kepentingan proses biologi". Cara kerja dari ilmu ini adalah dengan mengunakan ciri–ciri biologi yang dapat memberikan informasi yang unik berkaitan dengan identifikasi masing–masing individu. Dalam teknologi informasi, biometrik dikenal untuk mengukur dan menganalisis karakteristik bagian tubuh pada manusia umumnya seperti sidik jari, retina, DNA, pola suara, dan pola wajah.

Menurut logikanya, ilmu biometrik ini terbagi atas dua modul : yaitu modul pendaftaran dan modul identifikasi. Untuk modul pendaftaran dimana memiliki fungsi untuk mengambil data karakteristik dari masing–masing individu yang nantinya disimpan ke dalam sistem. Dalam modul pendaftaran, karakteristik dari masing–masing individu dipindai menggunakan alat pemindai khusus biometrik untuk menghasilkan sebuah representasi digital yang belum diolah. Untuk dapat digunakan dalam proses pencocokan, representasi digital tersebut diproses lebih lanjut untuk mendapatkan representasi yang cukup untuk mewakilinya yang disebut sebagai *template*. *Template* ini kemudian disimpan dalam pusat Database dalam sistem biometrik. Untuk modul identifikasi, difungsikan untuk mengidentifikasi masing–masing individu pada titik akses.

Pada modul ini pemindai biometrik menangkap karakteristik yang akan diidentifikasi dan diubah menjadi format digital, kemudian oleh ekstraktor fitur diproses menjadi representasi yang sama dengan *template*-nya dan kemudian dicocokkan untuk mendapatkan suatu identitas.

2.1.1 Pengenalan Pola Suara

Metode ini menangkap suara dari *speaker* menurut sifat-sifat bahasa. Penggunaan utamanya adalah aplikasi keamanan berbasis telepon. Keakurasiannya dapat dipengaruhi oleh hal-hal berikut seperti suara gaduh dan pengaruh-pengaruh dari penyakit atau kelelahan pada suara. Satu masalah nyata dengan pengenalan suara adalah sistem dapat dikelabui oleh suara *tape* dari suara seseorang. Untuk alasan ini sistem suara lanjutan harus mampu memperluas atau memperpanjang proses verifikasi dengan memberikan perkataan-perkataan yang lebih sulit dan panjang, membacanya dengan keras atau meminta sebuah perkataan yang berbeda yang dibaca setiap waktu. Selain itu pengenalan pola suara ini sangat berpengaruh terhadap frekuensi suara yang akan dikenali nantinya. Contoh dari penggunaan metode LPC untuk pengenalan suara paru-paru normal menunjukkan hasil yang cukup baik dengan akurasi dapat mencapai 98%. Hal ini dimungkinkan karena rekaman dan pemotongan datanya cukup baik sehingga tidak terlalu berbeda dalam tiap kelas data, mengingat sifat LPC yang time variant. Selain LPC, Jaringan Syaraf Tiruan – Backpropogation juga sering dipergunakan dalam pengenalan pola pada suara.

2.1.2 Pengenalan Pola Retina

Sebagai suatu fitur yang sangat penting pada tubuh manusia, retina memainkan peran penting di dalam ekstrasksi fitur wajah dan analisis ekspresi perwajahan. Secara nyata dapat kita katakan bahwa retina selain merupakan fitur yang sangat penting juga dipertimbangkan sebagai fitur yang relatif stabiluntuk diangkat di dlam kajian ekstraksi citra wajah dibanding fitur–fitur wajah yang lain. Inilah alasan yang kuat mengapa retina menjadi suatu fitur yang diambil sebagai salah satu tema yang dapat dijadikan bagian pada sistem biometrik.

Mungkin dari semua itu yang paling aman dari bekerjanya sistem biometrik adalah retina, dan lapisan-lapisan pembuluh yang dilokasikan di belakang mata. Gambar retina sulit untuk ditangkap dan selama pendataan *user* harus memusatkan sebuah titik serta mempertahankannya sehingga kamera dapat melaksanakan penangkapan gambar dengan baik. Hal yang sebenarnya ditentukan adalah pola dari pembuluh-pembuluh darah. Tetapi ketika pola-pola ini unik pada tiap-tiap orang, identifikasi dapat menjadi lebih presisi. Sistem yang didasarkan pada dua bagian mata, iris, dan retina adalah dipertimbangkan untuk menawarkan tingkat keamanan terbaik.

2.1.3 Pengenalan DNA

Dalam kehidupan kita, DNA merupakan suatu unit informasi kehidupan terkecil yang dimiliki oleh semua makhluk hidup dan diturunkan secara turun temurun. Semakin dekat kekerabatan seseorang maka semakin mirip DNA yang dimilikinya. DNA yang dimiliki tiap makhluk hidup ini adalah unik, dengan kata

lain DNA seseorang tidak mungkin sama dengan DNA orang lain, kecuali dengan DNA yang dimiliki keluarga. Sampel yang diambil untuk pengenalan DNA berasal dari seluruh tubuh makhluk hidup itu sendiri, contoh rambut, kulit, kuku, air liur, darah, dan lain–lain.

Rangkaian DNA, yang mengandung informasi kehidupan unik untuk setiap makhluk hidup terdiri dari kumpulan gugus karbon dimana setiap gugus karbon dapat dianalogikan sebagai karakter, sehingga DNA itu sendiri dapat dianalogikan sebagai rangkaian karakter atau String. DNA terdiri dari empat jenis gugus karbon yaitu Adenin (A), Sitosin (S), Timin (T), dan Guanin (G). Dengan demikian maka DNA sama dengan string yang merupakan kombinasi dari 4 jenis karakter yaitu A, S, T, dan G.

2.1.4 Pengenalan Pola Sidik Jari

Dalam pengenalan pola sidik jari terdapat hal–hal yang unik yaitu guratan–guratan yang terdapat pada dasar dari setiap jari baik tangan ataupun kaki. Tiap makhluk hidup memiliki guratan–guratan jari yang berbeda, maka dari itu pengenalan pola sidik jari ini dapat dikatakan sangat unik. Pengenalan pola sidik jari ini sudah dikenal pada akhir abad ke-19. Sidik jari sangat unik dimana jika ada anak kembar sekalipun, kedua sidik jari mereka tidak mungkin sama. Dalam penelitian para sains dikemukakan bahwa jika terdapat 5 juta manusia di bumi, maka kemungkinan munculnya dua sidik jari yang sama adalah 300 tahun kemudian.

Sidik jari dikelompokan menjadi tiga model, yaitu : model *arch* dimana kerutan pada sidik jari muncul dari ujung, kemudian mulai naik di tengah, dan berakhir di ujung yang lain. Untuk model *loop* kerutan muncul dari satu sisi jari, kemudian membentuk sebuah kurva , dan menuju keluar dari sisi yang sama ketika kerutan itu muncul. Sedangkan untuk model *whorl* kerutan berbentuk sirkuler yang mengelilingi sebuah titik pusat dari jari.

2.1.5 Pengenalan Pola Wajah

Pengenalan pola wajah adalah proses yang bertujuan untuk mengenali setiap wajah yang terdeteksi oleh sebuah kamera atau *webcam*, yang nantinya akan dicocokan dengan pola wajah yang terdapat di database. Proses ini akan berjalan dengan catatan wajah yang akan dikenali harus menginputkan data pola wajahnya pada database aplikasi.

Wajah manusia merupakan objek yang dinamis yang memilik variabilitas yang tinggi. Hal ini membuat sistem pengenalan wajah yang memanfaatkan komputer merupakan sistem yang tidak sederhana. Pada dasarnya ada lima langkah yang dilakukan dalam pengenalan wajah yaitu menangkap citra wajah baik langsung maupun tidak langsung, melakukan segmentasi atau deteksi wajah dengan cara melokalisir wajah dari latar belakangnya, mengekstraksi ciri wajah dari region wajah yang sudah terdeteksi untuk mendapatkan template wajah, membandingkan template dengan template pada basisdata wajah dan menunjukkan hasilnya.

Belakangan ini banyak teknik-teknik telah dikembangkan untuk mendeteksi dan mengenali wajah dalam sebuah citra pada digital. Namun sampai saat ini kemampuan mendeteksi wajah manusia secara otomatis berbantuan komputer belum bisa menandingi kemampuan manusia untuk melakukan hal yang sama. Karakteristik pada wajah memiliki tingkatan variabilitas yang tinggi dalam ukuran, bentuk, pose, warna dan tekstur. Faktor luar juga dapat mempengaruhi kesuksesan komputer dalam mendeteksi wajah diantaranya pencahayaan, posisi kamera, dan kualitas kamera itu sendiri.

2.2 Pengolahan Citra

Pengolahan citra merupakan teknik untuk memodifikasi atau menginterpretasi gambar yang ada, sperti foto dan rangkaian gambar film. Dua macam prinsip pengolahan citra adalah, Meningkatkan kualitas gambar Memberikan persepsi dari informasi visual, seperti pada robotic. Untuk melakukan pengolahan citra, pertama membuat digitasi dari foto menjadi file image.

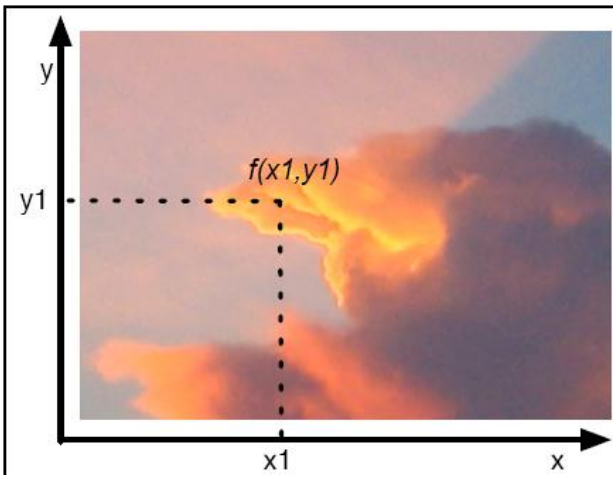
Citra / gambar *(image)* merupakan hal yang vital dan menjadi bagian integral dari kehidupan sehari-hari. Pada kepentingan tertentu, citra (gambar) digunakan sebagai alat untuk mengungkapkan pertimbangan *(reason)*, interpretasi, ilustrasi, penggambaran *(represent)*, ingatan *(memorise)*, pendidikan, komunikasi, evaluasi, navigasi, survey, hiburan, dan lain sebagainya. Tetapi kemudian konsep citra dan pengolahannya dihubungkan dengan pengubahan dan perbaikan citra (gambar) yang bertujuan antara lain :

a. memperbaiki kesalahan data sinyal gambar akibat transmisi dan selama akuisisi sinyal.
b. meningkatkan penampakan gambar sehingga dapat 'diterima' oleh sistem penglihatan manusia.

Umumnya pengolahan citra melibatkan alat bantu komputer, sehingga muncul istilah *'computer vision',* yang selanjutnya dihubungkan dengan konsep komputasi dan elektronik dari dunia mesin sejalan dengan atribut penglihatan manusia.

Aspek komputer adalah sebuah sistem yang berhubungan dengan elemen-elemen perangkat keras seperti sensor optik, arsitektur pengolahan paralel pada komputer, grafika komputer dan alat penampil *(display)* serta elemen-elemen perangkat lunak seperti manipulasi data dan perhitungan data gambar. Sedangkan aspek penglihatan *(vision)* adalah bayangan dari sistem penglihatan manusia dan melingkupi aspek fungsional dari mata, saraf optik dan otak.

Secara ideal, mesin diharapkan mempunyai kemampuan yang sama dengan kemampuan penglihatan manusia. Sehingga suatu pertimbangan (misalnya untuk membedakan obyek berdasarkan permukaan, warna, cahaya dan lain-lain) dibuat berdasarkan karakteristik sistem penglihatan manusia. Dalam kenyataannya, sistem penglihatan manusia sangat kompleks, yaitu dengan banyaknya tingkatan pengolahan pada mata dan jaringan saraf penglihatan di otak.



Gambar 2.1 *Citra Digital*

Beberapa faktor yang terdapat dalam citra atau *image*, adalah :

1**.** Intensitas cahaya

Intensitas dapat ditranslasikan menjadi suatu sinyal elektris, dan secara paling sederhana menggunakan *photosensitive cells* atau *photosensitive sesistive devices.* Salah satu dari alat – alat photosensitive dapat digunakan untukmembuat kamera primitif yang menghasilkan sederetan sinyal yang menghasilkan tingkatan–tingkatan intensitas cahaya untuk masing–masing spot pada gambar.

2. Warna

Penangkapan warna pada suatu citra meliputi penangkapan tiga citra secara simultan. Dengan sistem RGB, sebagai suatu standarisasi, intensitas masing – masing warna baik red, green, maupun blue harus diukur pada masing – masing spot. Dengan kamera yang beroperasi secara linear yang menjelajahi keseluruhan *visible spectrum,* kumpulan – kumpilanwarba yang sederhana dapat

digunakan untuk mengambil tiga citra yang masing – masing, satu untuk spektra red, green, dan blue.

Image processing merupakan suatu bagian terjadinya semua proses pengolahan citra dalam bentuk dua dimensi, yaitu dari pertama kali citra didapatkan dari sensor kemudian diproses sampai didapatkan hasil sesuai yang diinginkan. Bagian dari *image processing* yang digunakan dalam proses pengenalan objek, yaitu *image segmentation*, *thresholding, edge detection,* dan *object classification.*



Gambar 2.2 *Blok Diagram Pengenalan Bentuk*

2.2.1 *Image* Segmentasi

Bagian ini digunakan untuk memisahkan atau membedakan antara objek yang akan diproses dengan latar belakang dari objek tersebut. Metode yang digunakan yaitu segementasi berdasarkan warna. Segmentasi di sini digunakan untuk mengubah *colour image* yang diambil dari kamera menjadi citra biner. Pada akhir proses segmentasi ini akan diperoleh dua warna yaitu hitam dan putih.

2.2.2 Thresholding

Dalam aplikasi pengolahan citra seringkali berguna bila dapat dilakukan pemisahan objek yang diinginkan dengan latar belakang. *Thresholding* seringkali digunakan untuk melakukan segmentasi antara objek dengan latar dari suatu citra.

Dengan kata lain *thresholding* digunakan untuk mengatur jumlah derajat keabuan pada sebuah citra dengan cara melakukan perubahan kuantisasi pada citra.

Citra masukan yang biasanya dilakukan proses *thresholding* adalah citra hitam putih, dan citra keluaran yang dihasikan biasanya adalah citra biner yang menampilkan perbedaan antara objek dengan latar belakangnya. Penggunaan *thresholding* yang paling umum dan sederhana adalah pengkonversian citra hitam putih menjadi citra biner.Tidak semua citra dapat dipisahkan antara objek dengan latarnya dengan mengunakan *thresholding* yang sederhana. Hal ini disebabkan karena intensitas cahaya pada objek dan latar tidak terlalu berbeda, Untuk dapat memisahkan antara objek dan latar maka kedua intensitas tersebut harus jauh berbeda.

2.2.3 Deteksi Tepi

Yang dimaksud dengan tepi atau *edge* dalam hal ini adalah perubahan nilai derajat keabuan atau *brightness value* (BV) pada citra yang besar dalam jarak yang kecil. Perbedaan intensitas inilah yang menandakan adanya perbedaan objek, sehingga selanjutnya dapat diketahui objek-objek yang berbeda pada citra yang dianalisis. Tepi pada umumnya berada pada batas antara dua objek yang berbeda..

Macam-macam metode untuk proses deteksi tepi, antara lain: Metode Robert, Metode Prewitt, dan Metode Sobel.

2.3 Wajah

Setiap manusia memiliki ciri–ciri wajah yang berbeda–beda. Ciri–ciri yang paling dominan dan terlihat sangat jelas adalah warna pada wajah. Ciri–ciri ini sangat penting dalam ilmu biometrik dalam keakuratan data yang nantinya akan di proses.

2.3.1 Warna

Kulit dapat digunakan sebagai sumber informasi untuk proses pendeteksian. Informasi yang diperoleh dari kulit sangat relevan untuk proses pendeteksian antara lain untuk proses–proses pendeteksian pada wajah manusia (*face tracking*).

Secara tradisional oleh para pakar dibedakan ada tiga ras utama yaitu : Ras kulit hitam, Ras kulit putih, Ras kulit kuning. Namun setelah diteliti lebih lanjut ternyata pembagian ras manusia bisa lebih rinci lagi, seperti : *Ras Khoisan* (orang *Bushmen* atau *Hottentot* dari Afrika Selatan), *Ras Australoid* (orang *Davida*, orang Asia Tenggara "Asli", orang Papua, dan orang Australia), *Ras Negroid* (Kulit Hitam), *Ras Kaukasoid* (Kulit Putih), *Ras Mongoloid* (Kulit Kuning). Untuk umumnya sebenarnya hanya ada dua perbedaan utama yaitu orang Afrika dan Non – Afrika. Kemudian orang kulit hitam di daerah Asia Tenggara pada masa lalu mendiami seluruh India Selatan, Asia Tenggara hingga Australia ternyata setelah diteliti lebih mirip *Ras Mongoloid* dibandingkan *Ras Negroid*, meski banyak yang berambut keriting dan hitam.



Gambar 2.3 *Ras Mongoloid, RAS Kaukasoid, RAS Negroid*

Ras Mongoloid adalah ras manusia yang sebagian besar menetap di Asia Utara, Asia Timur, Asia Tenggara, Madagaskar di lepas pantai timur Afrika, beberapa bagian India Timur Laut, Eropa Utara, Amerika Utara, Amerika Selatan, dan Oseania. Ras ini biasa disebut "berkulit kuning". Namun ini tidak selalu benar. Misalkan orang Indian di Amerika dianggap berkulit merah dan orang Asia Tenggara seringkali berkulit coklat muda sampai coklat gelap.

2.4 Algoritma

Kata algoritma diserap dari kata algorithm. Pada awalnya kata algoritm atau algorism berasal dari nama seorang penulis buku Arab yang sangat terkenal, yaitu Abu Ja'far Muhammad Ibnu Musa Al-Khuwarizmi. Kata Al-Khuwarizmi oleh orang – orang barat dibaca menjadi al-gorism. Al-Khuwarizmi adalah penulis buku yang berjudul Kitab Al-Jabar Wal-Muqaabala (The Book of Restoration and Reduction), dan pada akhirnya kata algorism itu berubah menjadi algoritm.

Definisi Algoritma, "Algoritma adalah urutan langkah–langkah logis penyelesaian masalah yang disusun secara sistematis". Sedangkan menurut

Kamus Besar Bahasa Indonesia, terbitan Balai Pustaka, 1986. “Algoritma adalah urutan logis pengambilan keputusan untuk pemecahan masalah”.

2.4.1 Algoritma Camshift

Ada berbagai macam pendekatan algoritma yang dapat digunakan untuk melakukan pelacakan obyek, misalnya : pelacakan obyek dengan kontur, menggunakan teknik *Eigenspace*, menggunakan suatu set hipotesa statistik, mengkonvolusi citra dengan fitur detektor, dan masih banyak lagi lainnya.



Gambar 2.4 *Tracking Face dengan Camshift*

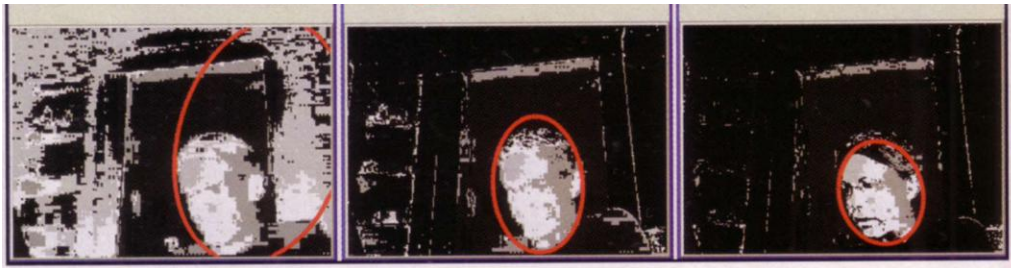
CamShift adalah singkatan dari *Continuously Adaptive Mean Shift*, yang merupakan pengembangan dari algoritma *Mean Shift* yang dilakukan secara terus menerus (berulang) untuk melakukan adaptasi atau penyesuaian terhadap distribusi probabilitas warna yang selalu berubah tiap pergantian *frame* dari *video sequence*. Langkah-langkah dari algoritma *Mean Shift* adalah sebagai berikut:

a. Ukuran *search window* yang sudah ditentukan.
b. Lokasi awal *search window* yang sudah ditentukan.
c. Hitung daerah *mean* dalam *search window.*
d. Posisikan *search window* ke tengah daerah *mean* seperti dihitung pada *step*

e. Ulangi *step* c dan d hingga konvergen (atau hingga pergeseran daerah mean kurang dari *threshold/* batas yang ditentukan).

Sedangkan langkah-langkah dari algoritma *CamShift* adalah sebagai berikut:

a. Tentukan ukuran awal *search window.*
b. Tentukan lokasi awal dari *search window.*
c. Tentukan daerah kalkulasi (calculation region) pada bagian tengah *search window* dengan ukuran lebih besar dari *search window.*
d. *Frame* citra video dikonversi ke dalam sistem warna HSV (Hue, *Saturation, Value),* dan dilakukan *color histogram lookup* dalam *calculation region* yang akan menghasilkan citra distribusi probabilitas warna kulit.
e. Lakukan algoritma *Mean Shift* seperti di atas (satu atau banyak iterasi) dengan input berupa ukuran dan lokasi *search window* serta citra distribusi probabilitas warna, simpan *zeroth moment.*
f. Set nilai x, y, z, dan *head roll* yang diperoleh dari *step* e.
g. Nilai x, y dipakai untuk set titik tengah *search window,* (1/22* *area*) untuk set ukuran *search window.*
h. Ulangi *step* c untuk setiap pergantian *frame* citra video.



Gambar 2.5 *1. Menentukan ukuran awal search window , 2 Menentukan lokasi awal dari search window, 3. menentukan daerah kalkulasi*

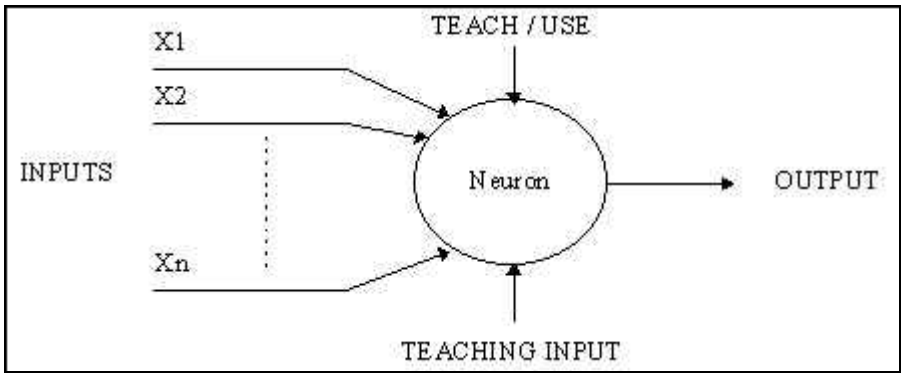
2.4.2 Jaringan Syaraf Tiruan

Jaringan Syaraf Tiruan dibuat pertama kali pada tahun 1943 oleh *neurophysiologist* Waren McCulloch dan *logician* Walter Pits, namun teknologi yang tersedia pada saat itu belum memungkinkan mereka berbuat lebih jauh.

Jaringan Syaraf Tiruan adalah paradigma pemrosesan suatu informasi yang terinspirasi oleh sistim sel syaraf biologi, sama seperti otak yang memproses suatu informasi. Elemen mendasar dari paradigma tersebut adalah struktur yang baru dari sistim pemrosesan informasi. Jaringan Syaraf Tiruan, seperti manusia, belajar dari suatu contoh. Jaringan Syaraf Tiruan dibentuk untuk memecahkan suatu masalah tertentu seperti pengenalan pola atau klasifikasi karena proses pembelajaran. Jaringan Syaraf Tiruan berkembang secara pesat pada beberapa tahun terakhir.

Jaringan Syaraf Tiruan telah dikembangkan sebelum adanya suatu komputer konvensional yang canggih dan terus berkembang walaupun pernah mengalami masa vakum selama beberapa tahun. Jaringan Syaraf Tiruan keluar dari penelitian kecerdasan buatan, terutama percobaan untuk menirukan *fault-*

tolerence dan kemampuan untuk belajar dari sistem syaraf biologi dengan model struktur *low-level* dari otak. Otak terdiri dari sekitar (10.000.000.000) sel syaraf yang saling berhubungan. Sel syaraf mempunyai cabang struktur input (*dendrites*), sebuah inti sel dan percabangan struktur output (axon). *Axon* dari sebuah sel terhubung dengan *dendrites* yang lain melalui sebuah *synapse*. Ketika sebuah sel syaraf aktif, kemudian menimbulkan suatu signal *electrochemical* pada *axon*. Signal ini melewati *synapses* menuju ke sel syaraf yang lain. Sebuah sel syaraf lain akan mendapatkan signal jika memenuhi batasan tertentu yang sering disebut dengan nilai ambang atau (*threshold*).



Gambar 2.6 *Sebuah Sel Syaraf Sederhana*

Jaringan Syaraf Tiruan mulai dilirik banyak kalangan karena mempunyai banyak kelebihan dibandingkan sistem konvensional. Jaringan Syaraf Tiruan mewakili pikiran manusia untuk mendekatkan diri dengan komputer, maksudnya Jaringan Syaraf Tiruan dirancang agar komputer dapat bekerja seperti/layaknya otak manusia. Berikut ini beberapa keunggulan dari Jaringan Syaraf Tiruan adalah :

1. *Adaptive learning*: Suatu kemampuan untuk melakukan suatu kegiatan yang didasarkan atas data yang diberikan pada saat pembelajaran atau dari pengalaman sebelumnya.

2. *Self-Organisation*: Dapat membuat organisasi sendiri atau merepresentasikan informasi yang didapat pada saat pembelajaran.
3. *Real Time Operation*: Dapat menghasilkan perhitungan parallel dan dengan device hardware yang khusus yang dibuat akan memberikan keuntungan dengan adanya kemampuan tersebut.
4. *Fault Tolerance* melalui *Redundant Information Coding*: Kerusakan pada bagian tertentu dari jaringan akan mengakibatkan penurunan kemampuan. Beberapa jaringan mempunyai kemampuan untuk menahan kerusakan besar pada jaringan.
5. Kelebihan Jaringan Syaraf Tiruan terletak pada kemampuan belajar yang dimilikinya. Dengan kemampuan tersebut *pengguna tidak perlu merumuskan kaidah atau fungsinya.* Jaringan Syaraf Tiruan akan belajar mencari sendiri kaidah atau fungsi tersebut. Dengan demikian Jaringan Syaraf Tiruan mampu digunakan untuk menyelesaikan masalah yang rumit dan atau masalah yang terdapat kaidah atau fungsi yang tidak diketahui.
6. Kemampuan Jaringan Syaraf Tiruan dalam menyelesaikan masalah yang rumit telah dibuktikan dalam berbagai macam penelitian.

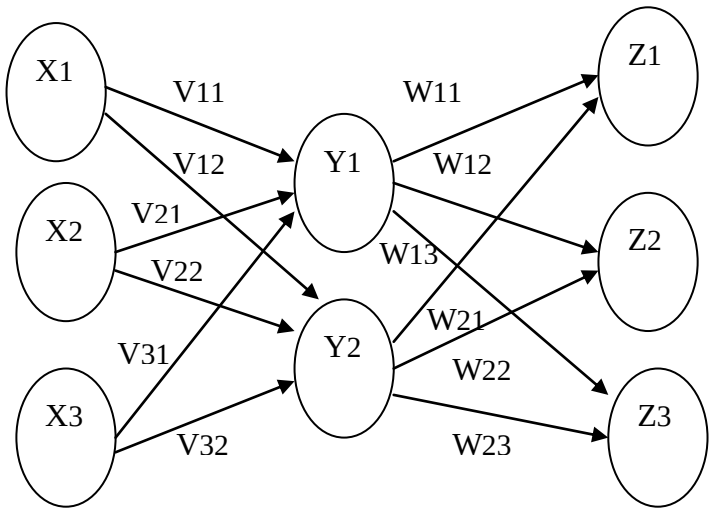
2.4.2.1 Algoritma Backpropogation

Algoritma Backpropogation adalah salah satu metode dari Jaringan Syaraf Tiruan. Jaringan Syaraf Tiruan adalah suatu sistem pemrosesan informasi yang cara kerjanya memiliki kesamaan tertentu dengan jaringan syaraf biologis

[Fausett, 1994]. Jaringan Syaraf Tiruan dikembangkan sebagai model matematis dari syaraf biologis dengan berdasarkan asumsi bahwa:

1. Pemrosesan terjadi pada elemen-elemen sederhana yang disebut neuron.
2. Sinyal dilewatkan antar neuron melalui penghubung.
3. Setiap penghubung memiliki bobot yang akan mengalikan sinyal yang lewat.
4. Setiap neuron memiliki fungsi aktivasi yangakan menentukan nilai sinyal output.

Jaringan Syaraf Tiruan dapat digolongkan menjadi berbagai jenis berdasarkan pada arsitekturnya, yaitu pola hubungan antara neuron-neuron, dan algoritma trainingnya, yaitu cara penentuan nilai bobot pada penghubung. Seperti contoh gambar dibawah ini yang menerangkan arsiterktur Jaringan Syaraf Tiruan dari algoritma Backpropogation (Fase Foward).



Gambar 2.7 *Arsitektur Jaringan Backpropogation* (Fase Forward)

Salah satu algoritma pelatihan jaringan syaraf tiruan yang banyak dimanfaatkan dalam bidang pengenalan pola adalah backpropagation. Algoritma ini umumnya digunakan pada jaringan syaraf tiruan yang berjenis multi-layer feed-forward, yang tersusun dari beberapa lapisan dan sinyal dialirkan secara searah dari input menuju output. Algoritma pelatihan backpropagation pada dasarnya terdiri dari tiga tahapan [Fausett, 1994], yaitu:

1. Input nilai data pelatihan sehingga diperoleh nilai output.

2. Propagasi balik dari nilai error yang diperoleh.

3. Penyesuaian bobot koneksi untuk Meminimalkan nilai error.

Ketiga tahapan tersebut diulangi terus-menerus sampai mendapatkan nilai error yang diinginkan. Setelah training selesai dilakukan, hanya tahap pertama yang diperlukan untuk memanfaatkan jaringan syaraf tiruan tersebut. Secara matematis [Rumelhart, 1986], ide dasar dari algoritma backpropagation ini sesungguhnya adalah penerapan dari aturan rantai (*chain rule*) untuk menghitung pengaruh masing-masing bobot terhadap fungsi error:

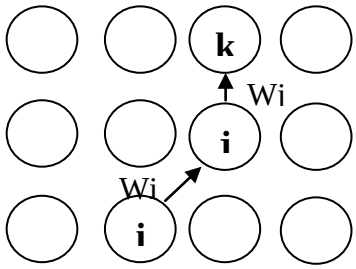
$$\frac{\partial \mathrm{E}}{\partial w_{ij}} = \frac{\partial \mathrm{E}}{\partial s_i} \frac{\partial s_i}{\partial w_{ij}}$$

dan

$$\frac{\partial s_i}{\partial w_{ij}} = \frac{\partial s_i}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} = f'_{\log}(net_i)\, s_j$$

Dimana *ij w* adalah bobot penghubung dari neuron *j* ke neuron *i*, *i s* adalah output, dan *i net* adalah jumlah hasilkali pada input dari neuron *i*. pembelajaran

jaringan syaraf tiruan dengan algoritma backpropagation memperbaiki bobot (w) antara hubungan jaringan neuron berdasarkan nilai kemelesetan (error value) pada lapisan neuron keluaran, dimana nilai kemelesetan ini tergantung pada selisih antara nilai keluaran yang dihasilkan jaringan syaraf tiruan (o) dengan nilai keluaran yang menjadi target (d). misalkan jaringan syaraf tiruan terdiri dari tiga lapis dengan algoritma backpropogation sebagai berikut :



Gambar 2.8 *Jaringan Syaraf Tiruan Tiga Lapis*

Sesuai dengan gambar diatas untuk jenis algoritma backpropogation, maka hubungan antara neuron lapisan masukan – keluaran secara matematis dapat dijabarkan sebagai berikut :

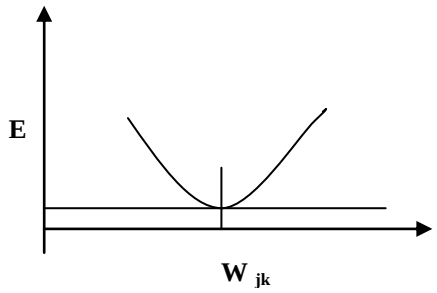
$$o_k = \frac{1}{1+e^{-net_k}}$$

$$net_k = \sum_j w_{jk} o_j$$

$$o_j = \frac{1}{1+e^{-net_j}}$$

$$net_j = \sum_j w_{ij} o_i$$

Tujuan dari jaringan syaraf tiruan adalah memperkecil nilai kemelesetan pada keluaran (nilai yang dihasilkan), hubungan antara nilai kemelesetan tersebut untuk pola p dan unit keluaran k adalah :



Gambar 2.9 *Hubungan E dengan Nilai Bobot Neuron*

Jadi untuk hubungan nilai E dengan nilai bobot neuron untuk secara sistematisnya dapat dijabarkan sebagai berikut :

$$\mathrm{E} = \frac{1}{2}\sum_{o}\left[\sum_{p}(t_{pk} - o_{pk})^2\right]$$

Nilai kemelesetan E dipergunakan untuk menambah bobot w antara neuron, jadi ada keterkaitan nisba E terhadap w. sebagai berikut :

- Hubungan neuron hidden (j) – neuron keluaran (k)

$$\frac{\partial \mathrm{E}}{\partial w_{jk}} = \frac{\partial \mathrm{E}}{\partial \mathrm{P_K}}\frac{\partial \mathrm{P_K}}{\partial net_{\mathrm{K}}}\frac{\partial net_{\mathrm{K}}}{\partial w_{jk}} = -(t_{\mathrm{K}} - o_{\mathrm{K}})o_{\mathrm{K}}(-o_{\mathrm{K}})o_j$$

$$w_{jk} \leftarrow w_{jk} + (\eta)((t_{\mathrm{K}} - o_k)o_k(-o_k)o_j)$$

Untuk tujuan penyederhanaan persamaan dan memudahkan pemahaman, persamaan diatas ditulis kembali dalam bentuk :

$$\partial_{\mathrm{K}} = (t_{dk} - o_{\mathrm{K}})o_{\mathrm{K}}(-o_k)$$

$$\Delta w_{jk} = \eta\delta_{\mathrm{K}}o_j$$

- Hubungan neuron masukan (i) – neuron hidden (j)

$$\frac{\partial \mathrm{E}}{\partial w_{ij}} = \sum_{\mathrm{K}}\frac{\partial \mathrm{E}}{\partial o_{\mathrm{K}}}\frac{\partial o_{\mathrm{K}}}{\partial net_{\mathrm{K}}}\frac{\partial net_{\mathrm{K}}}{\partial o_j}\frac{\partial o_i}{\partial net_j}\frac{\partial net_i}{\partial o_{ij}}$$

$$= \sum_{K} - (t_K - o_k) o_k (1 - o_k) w_{jk} o_j (1 - o_j) o_j$$

$$= \sum_{K} - \delta_k w_{jk} o_j (1 - o_j) o_i$$

$$= -o_i o_j (1 - o_j) \sum_{K} \partial_K w_{jK}$$

$$= -o_i \delta_j$$

$$\delta_j = o_j (1 - o_j) \sum_{K} \partial_K w_{jK}$$

$$\Delta w_{ij} = \eta \delta_K o_i$$

Dengan demikian, untuk jaringan syaraf tiruan yang terdiri dari tiga lapis dengan paradigma pembelajaran terawasi model algoritma backpropogation, bentuk persamaan dapat ditulis sebagai berikut :

$$\Delta w_{ij} = \eta \delta_j o_i$$

$$\delta_j = o_j (1 - o_j) \sum_{K} \partial_K w_{jK}$$

$$\partial_K = (t_k - o_K) o_K (1 - o_k)$$

$$\Delta w_{jk} = \eta \delta_k o_j$$

2.5 Konsep Dasar Sistem

Suatu sistem terdiri dari elemen-elemen yang saling berhubungan dengan melalui berbagai interaksi dan kerjasama untuk mencapai satu tujuan. Elemen-elemen tersebut tidak dapat dipisahkan antara satu dengan yang lainnya, dimana saling berhubungan dan saling mempengaruhi proses secara keseluruhan.

Dalam mendefinisikan suatu sistem, terdapat beberapa pendekatan yang sering digunakan. Pendekatan pertama dalam mendefinisikan sistem lebih menekankan pada prosedur. Pendefinisian sistem dengan penekanan prosedur sebagai berikut :

"*Sistem adalah suatu jaringan kerja dari prosedur-prosedur yang saling berhubungan, berkumpul bersama-sama untuk melakukan suatu kegiatan atau untuk menyelesaikan suatu sasaran tertentu*". (H. M Jogiyanto, 1993:1)

Pendekatan yang kedua yang sering dipergunakan dalam mendefinisikan sistem lebih menekankan pada elemen atau komponennya. Pendefinisian sistem dengan penekanan elemen sebagai berikut :

"*Sistem adalah kumpulan dari elemen-elemen yang berorientasi untuk mencapai suatu tujuan*". (H. M Jogiyanto, 1993:2)

Sistem dapat diartikan sebagai suatu kesatuan dari dua atau lebih komponen yang berinteraksi untuk mencapai satu tujuan. Pendekatan sistem yang menekankan pada komponen lebih mudah untuk dipelajari dan diterapkan pada berbagai bidang terutama untuk pengembangan sistem lebih lanjut.

Perancangan sistem secara umum adalah suatu tahap dimana di dalamnya terdapat identifikasi komponen-komponen sistem informasi yang akan dirancang secara rinci yang bertujuan untuk memberikan gambaran kepada pengguna atau *user* mengenai sistem yang akan dibangun. Sedangkan desain sistem secara terinci dimaksudkan untuk pembuat program komputer dan ahli teknik lainnya yang akan mengimplementasikan sistem.

Penggambaran dan perancangan model sistem secara logika dapat dibuat dalam bentuk Diagram Alir Data (DAD) atau *Data Flow Diagram* (DFD), Arus data pada Diagram Alir Data atau DFD nantinya dapat dijabarkan dengan menggunakan kamus data (*Data Dictionary*).

2.5.1 Diagram Konteks

Diagram konteks (*contex diagram*) adalah keadaan sistem secara umum dimana memiliki hubungan-hubungan dengan komponen diluar sistem atau dengan sistem yang lainnya yang digambarkan secara logika. Definisi diagram konteks adalah sebagai berikut :

"*Diagram konteks adalah diagram yang tidak detail dari sebuah sistem informasi yang menggambarkan aliran-aliran data masuk dan keluar dari sistem. Diagram ini digambarkan sebuah lingkaran yang menjelaskan tentang batasan sistem yang saling berhubungan dengan kesatuan luar (external entity) yang akan memberikan masukan dan menerima keluaran dari sistem tersebut yang dihubungkan dengan aliran yang menghubungkan sebuah sistem dengan lingkaran sistem*". (Lani Sidharta, Sistem Informasi Bisnis, 1995 : 66)

2.5.2 Diagram Aliran Data

Diagram arus data adalah suatu alat yang digunakan pada metodologi pengembangan sistem yang terstruktur. Dari diagram arus ini dapat tergambarkan sistem secara keseluruhan dan juga merupakan dari representasi grafik dari sebuah

sistem dimana menunjukan proses dan aliran data yang keluar dan masuk ke dalam sistem, akan tetapi tidek menunjukan dimana, bagaimana, dan oleh siapa proses itu dilakukan. Definisi *Data Flow Diagram* adalah sebagai berikut :

"*Data Flow Diagram adalah representasi grafik yang menggambarkan arus data dari suatu sistem. Data Flow Diagram menggambarkan komponen-komponen tersebut, asal dan tujuan dan penyimpanan data*". (Lani Sidharta, Sistem Informasi Bisnis, 1995 : 65)

Simbol-simbol yang digunakan diagram alir data atau *Data Flow Diagram* adalah sebagai berikut :

1. Kesatuan Luar (External Entity)

Kesatuan Luar (External Entity) merupakan kesatuan dilingkungan luar sistem yang dapat berupa manusia, organisasi, atau sistem lain. Lingkungan luar dan batas sistem dinotasikan dalam simbol yang sama.



Gambar 2.10 *Notasi Kesatuan Lingkungan*

2. Arus Data (*Data Flow*)

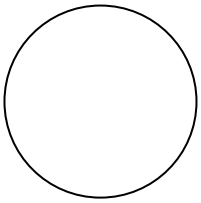
Arus Data (*Data Flow*) di dalam DFD diberi simbol suatu panah. Arus data ini mengalir diantara proses (*process*), simpanan data (*data storage*), dan kesatuan luar (*external entity*)



Gambar 2.11 *Notasi Arus Data*

3. Proses (*process*)

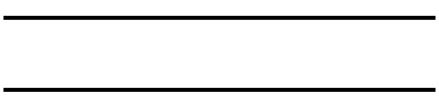
Proses (*process*) adalah kegiatan yang dilakukan oleh manusia, mesin, atau komputer yang bertujuan mengolah data yang dimasukan menjadi keluaran yang diinginkan.



Gambar 2.12 *Notasi Proses*

4. Simpanan Data (File)

Simpanan data disimbolkan dengan sepasang garis horizontal paralel. Simpanan data adalah tempat menyimpan atau untuk mengakses data yang diperlukan oleh sistem.



Gambar 2.13 *Notasi Table*

2.5.3 Kamus Data

Kamus data merupakan kumpulan dari data-data yang ada pada suatu sistem. Tujuannya adalah untuk memberikan informasi mengenai definisi struktur, pemakaian masing-masing elemen. "*Elemen adalah unit data yang terkecil*". (Tauri D. Mahyozir, Analisa dan Perancangan Sistem Pengolahan Data, 1995 : 44). Elemen-elemen yang dapat menyusun sebuah sistem ini adalah, sebagai berikut :

1. Tujuan, merupakan tujuan dari sistem tersebut.
2. Batasan, merupakan batasan-batasan yang ada untuk mencapai tujuan dari sistem.
3. Kontrol, merupakan pengawas dari pelaksanaan pencapaian tujuan sistem.
4. Input, merupakan bagian dari sistem yang memberikan data masukan kedalam sistem.
5. Proses, merupakan bagian yang memproses masukan data menjadi informasi yang sesuai dengan keinginan penerima.
6. Output, merupakan keluaran atau tujuan akhir dari sistem.
7. Umpan Balik, merupakan elemen sistem yang memiliki tugas untuk melihat kembali apakah sistem telah berjalan sesuai dengan yang diinginkan.

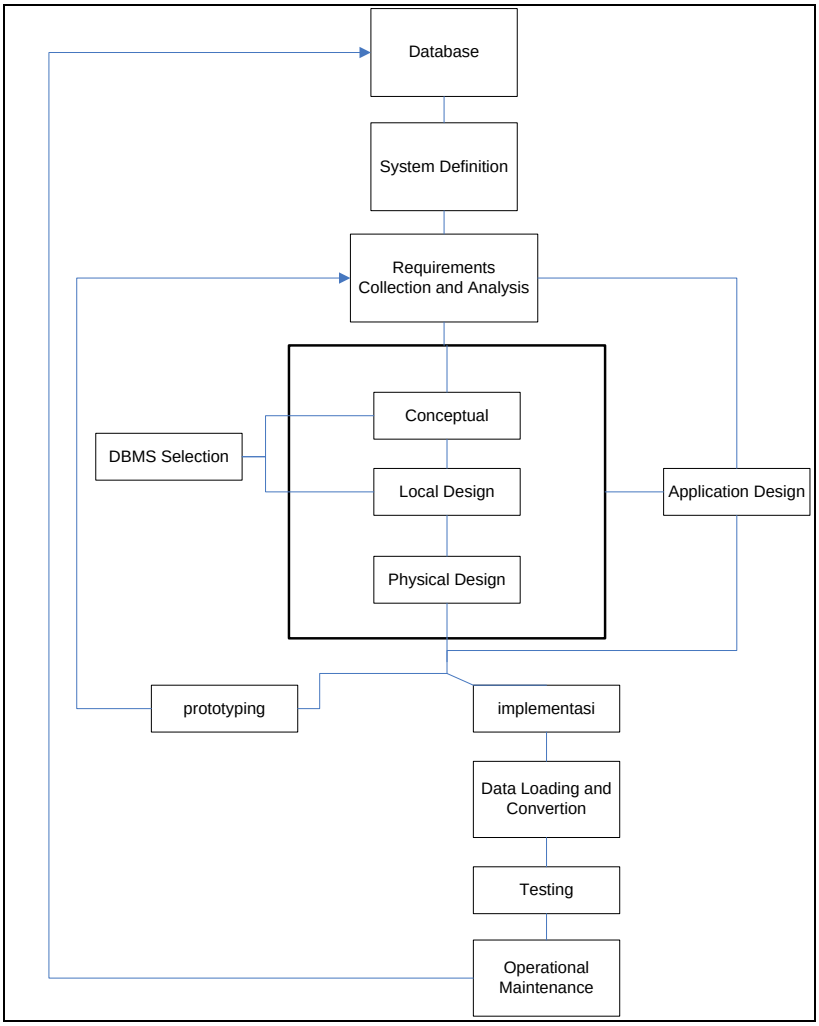
Kamus data berisi keterangan tentang arus data, alias, bentuk data, penjelasan, periode, volume, dan struktur data yang merupakan katalog fakta tentang data dan kebutuhan-kebutuhan informasi dari suatu sistem.

2.6 Konsep Dasar Database

Database adalah suatu koleksi data komputer yang terintegrasi, diorganisasikan dan disimpan dengan cara yang memudahkan pengambilan kembali. DASD (medium file master yang baik) harus digunakan. Tujuan utama dari konsep Database adalah meminimumkan pengulangan data dan mencapai independensi. Pengulangan data (data redundansi) adalah duplikasi data artinya data yang sama disimpan dalan beberapa file. Independensi data adalah kemampuan untuk membuat perubahan dalan struktur data tanpa membuat perubahan pada program yang memproses data. Independensi data dicapai dengan menempatkan spesifikasi data dalam label dan kamus yang terpidah secara fisik dari program. Program mengacu pada tabel untuk mengakses data. Perubahan pada struktur data hanya dilakukan sekali, yaitu dalam tabel. Ketika perusahaan mengadopsi konsep Database, hirarki data menjadi:

a. Database
b. file
c. catatan
d. elamen data

File-file tersendiri dapat tetap ada, mewakili komponen -komponen utama dari database namun organisasi fisik dari data tidak menghambat pemakai. Tersedia berbagai cara untuk mengintegrasikan isi dari file-file yang memiliki hubungan logis.



Gambar 2.14 *Siklus Database*

2.6.1 Bentuk Database

1. Hirarki Database

Biasa digunakan untuk jaringan komunikasi data yang berupa hirarki. Dasar hirarki database berusaha untuk menggambar realita dalam sebuah organisasi kebentuk data komputer.

2. Jaringan Database

Jaringan database dibuat karena jaringan komunikasi memiliki topologi *Mesh*, sehingga membutuhkan bentuk ini. Jaringan Database memiliki struktur file yang sama, sehingga file yang satu dapat mengetahui / mendapatkan informasi file yang lain dengan benar.

2.6.2 Struktur Database

Integrasi logis file dapat dicapai secara eksplisit atau secara implisit. Untuk Hubungan eksplisit *inverted index* dan *link field* menetapkan hubungan eksplisit antara data yang terintefrasi secara logis dalam file yang sama. Suatu pendekatan untuk menetapkan hubungan eksplisit antara catatan dari beberapa file adalah dengan menyusun catatan-catatan tersebut dalam suatu hirarki. Ini disebut struktur hirarkis. Dalam struktur seperti ini, setiap catatan pada satu tingkat dapat dihubungkan ke berbagai catatan yang setingkat lebih rendah. Catatan yang memiliki anak disebut *parent* dan anak catatan itu disebut *children*, dan untuk Hubungan implisit Pada awal 1970-an Edgar f. Codd dan C.J. Date, keduanya dari IBM tetapi bekerja secara terpisah, mengembangkan statu pendekatan untuk menetapkan hubungan antar catatan yang tidak harus dinyatakan secara eksplisit. *Link field* khusus tidak perlu disertakan dalam catatan. Pendekatan *Codd* dan *Date* dinamai struktur relasional, dan menggunakan hubungan implisit, yaitu hubungan yang dapat dinyatakan secara tidak langsung dari catatan data yang telah ada. Keuntungan utama dari struktur relasional bagi CBIS adalah fleksibelitas yang ditawarkanya dalam rancangan dan penggunaan database. Pemakai dan spesialis

informasi dibebaskan dari keharusan mengidentifikasi semua informasi yang diperlukan sebelum menciptakan database. Tipe-tipe database :

a. *Operational Database*, database menyimpan data detail yang dibutuhkan untuk mendukung operasi dari *entire organization*.
b. *Analytical Database*, menyimpan data dan information *extrated* dari operasional yang diseleksi dan luar database. Meliputi data dan informasi yang banyak dibutuhkan oleh manajer organisasi dan *user*.
c. *Data WareHouse*, Merupakan pusat data sentral yang ditampilkan dan diintegrasikan sehingga dapat digunakan oleh manajer dan *user* professional untuk macam-macam analisis bisnis, penelitian pasar dan *decision support*.
d. *Distributed Database*.
e. *End User Database*, Database terdiri dari variasi data yang dikembangkan oleh end user pada workstation.
f. *HyperMedia Database.*
g. *External Database.*

2.6.3 Komponen Database :

1. **File Database** : memiliki elemen-elemen data yang disimpan dalam salah satu format organisasi file database.

2. **DBMS** : suatu kelompok program software yang mengelola database, mengontrol akses terhadap database, menjaga pengamanan database dan melakukan tugas-tugas lain.
3. **Sistem Antar-Muka Bahasa Induk** : (*A Host Language Interface sistem*)

   Bagian dari DBMS yang berkomunikasi dengan program aplikasi, menafsirkan intruksi dan bahasa tingkat tinggi aplikasi.
4. Program Aplikasi.
5. Sebuah **sistem Antar muka Bahasa Alami** : (*A Natural Language Interface sistem*) Suatu bahasa pertanyaan (*query language*) yang memungkinkan pemakai untuk mendapatkan keterangan tentang apa saja yang tersedua pada sistem komputer.
6. **Kamus Data** : (*data dictionary*) Pusat penyimpanan infomasi data-data dari database yang memuat skema database, yang mana nama dari setiap item dalam database serta deskripsi dan definisi atribut-atributnya yang merujuk pada data standar.
7. **Terminal Pengaksesan dan pemutakhiran yang online** : Letaknya dapat berdekatan / berjauhan.
8. **Sistem keluaran / pembuat Reportase** : (*The output sistem or report Generator*) Terdiri dari laporan biasa ,dokumen dan laporan khusus.

2.6.4 Kriteria Database

1. Stuktur filenya memudahkan untuk mengcutkan suatu record dengan record lainnya.
2. Penggabungan file secara menyilang dimungkinkan, sehingga record yang sebelumnya bebas karena biasa digabung dan diproses bersama secara otomatis.
3. File program/datanya bersifat bebas, sehingga memudahkan untuk pemutakhiran dan perawatan database.
4. Memilih rumusan bersama (*common definition*) dalam kaitannya dengan definisi data, format record dan berbagai jenis deskripsi lainnya.
5. Memiliki DBMS untuk mengelola data.
6. Kamus Data
7. Memiliki memori akses langsung yang besar untuk memuat data DBMS.
8. Memiliki program dan piranti komunikasi yang canggih, yang memungkinkan pengguna untuk mengakses data secara serempak.
9. Memiliki teknik-teknik penyalinan (*back up*), penghidupan kembali (*restart*) dan perolehan kembali (*recovery*) yang canggih yang dapat merekrontuksi kembali file-file database jika ada data yang rusak/hilang.
10. Adanya *Query Language*.

2.6.5 Manajerial Database

Manajerial Database meliputi :

1. Sistem *Intelegent*, Untuk perencanaan strategis, baik dalam substansi jumlahnya maupun sifatnya bagi kegiatan manajemen puncak.
2. Masalah-masalah management khusus.
3. Model Manajemen.
4. Tugas Kunci sistem informasi.

Perangkat lunak yang menetapkan dan memelihara integrasi logis antar file, baik eksplisit maupun implisit disebut sistem manajemen Database(*database management system*) DBMS. IDS dari *General Electric* adalah contoh pertamanya dan kemudian diikuti oleh sejumlah usaha serupa dari pemasok perangkat keras dan perangkat lunak lain. Contoh DBMS yang menggunakan struktur hirarkis adalah IMS (*Infirmation Management Sistem*) dari IBM dan Sistem 2000 dari Intel.

Proses menciptakan Database mencakup tiga langkah utama, yaitu; Menentukan kebutuhan data, meliputi pendekatan berorientasi proses, pendekatan model perusahaan. Menjelaskan data, dengan cara sistem kamus data, data description language, dan Memasukan data.

Pengelola Database seorang spesialis informasi yang bertanggung jawab atas Database disebut pengelola Database atau DBA. Tugas DBEA terbagi dalam empat bidang utama; Perencanaan Database, mencakup sama dengan para

manajer untuk mendefinisikan skema perusahaan dengan para pemakai untuk mendefinisikan subskema mereka. Selain itu juga perperan penting dalam memilih DBMS. Penerapan database, terdiri dari menciptakan database yang sesuai dengan DBMS yang dipilih, serta menetapkan dan menegakkan kebijakan dan prosedur penggunaaan database. Operasi database, mencakup menawarkan program pendidikan kepada pemakai database dan menyediakan bantuan saat diperlukan. Keamanan database, meliputi pemantauan kegiatan database dengan menggunakan statistik yang disediakan DBMS. Selain itu juga memastikan bahwa database tetap aman.

2.6.6 DBMS (*Database Management System*)

DBMS: Suatu cara dalam bentuk sistem yang berguna dalam menyimpan data penggunaan cara yang tepat dapat mempercepat penyimpanan data, pemrosesan data dan pengambilan data. Empat hal penting dalam DBMS :

a. *Query language*
b. Security consideration
c. Biaya tidak langsung pemrosesan
d. Kecocokan dengan tipa aplikasi

Elemen-elemen utama dari DBMS :

a. Data *description language processor*
b. Performance statistics processor

c. Modul *backup/recovery*

d. Manajer database

Keuntungan DBMS :

a. Mengurangi pengulangan data

b. Mencapai independesi data

c. Mengintegrasikan data beberapa file

d. Mengambil data dan informasi secara cepat

e. Meningkatkan keamanan

f. Meningkatkan presentasi kesiapan data (*data availability*) yang berarti tersedia pada waktu dibutuhkan.

g. Mempercepat penyimpanan dan pengambilan data, mempermudah pemrosesan datadan Mempermudah pemrograman karena fleksibel.

h. Mengurangi penyimpanan data yang rangkap.

Kerugian DBMS :

a. Kurangnya ahli database.

b. Biaya pemrosesan data sangat tinggi.

c. Kebutuhan software dan Hardware yang bertambah.

d. Memperoleh perangkat lunak yang mahal.

e. Memperoleh konfigurasi perangkat keras yang besar.

f. Mempekerjakan dan mempertahankan staf DBA.

2.7 Aplikasi Yang Digunakan

Pada era globalisasi banyak aplikasi yang sering digunakan untuk membuat suatu perangkat lunak dalam bidang ilmu komputer, dimana sudah menjadi pengetahuan umum yang dapat dikenal oleh orang banyak. Aplikasi yang sering digunakan terdiri dari dua hal yang penting yaitu untuk pembangun perangkat lunak, dan Database. Contoh beberapa bahasa pemrograman yang diimplementasikan pada suatu program dimana sering dipergunakan diantaranya Delphi dan Visual Basic. Dan untuk Databasenya yaitu DATABASE Microsoft Access, dan Sql.

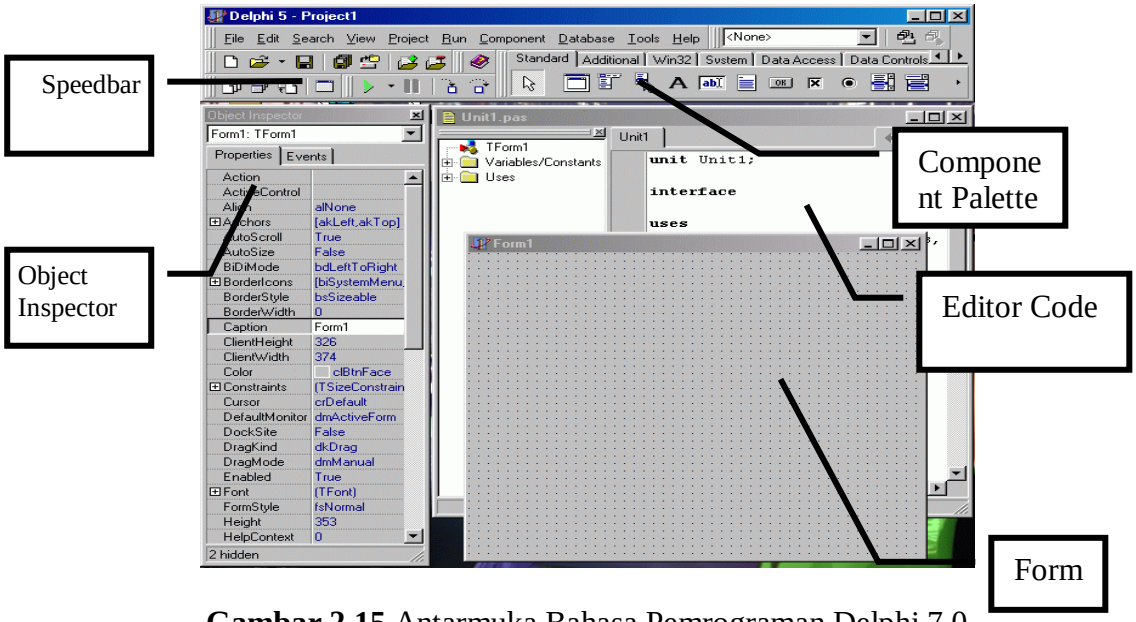
2.7.1 Sejarah Delphi

Delphi merupakan generasi lanjutan dari Turbo Pascal (yang diluncurkan tahun 1983 oleh *Borland International Incoporation*). Sesuai dengan sistem operasi saat ini, Turbo Pascal memang dirancang utntuk dijalankan pada sistem operasi DOS. Kemudian seiring dengan perkembangan zaman, *Borland International Incoporation* merilis Turbo Pascal untuk sistem operasi Windows, versi ini dijalankan pada Windows 3.X.

Pada tahun 1992 lahir suatu bahasa pemrograman baru yang diberi nama Borland Pascal Versi 7, hasil penggabungn Turbo Pascal dengan Turbo Pascal For Windows. Pada tahun 1995 muncul Borland Delphi, berselang setahun kemudian diluncurkan Borland Delphi versi 2.0, versi 3.0, dan sampai dengan kini Borland Delphi versi 8.0.

2.7.2 Mengenal Delphi

Pada saat menjalankan Delphi, akan terlihat sebuah jendela yang akan menampilkan proyek baru, dan pada proyek ini terdapat sebuah *form* yang dapat dirancang dan disusun program aplikasi. Inilah yang disebut IDE (*Integrated Development Environment*). Melalui IDE inilah pemrograman secara visual dapat merancang tampilan untuk pemakai (antarmuka pemakai) dan menuliskan programnya.



Gambar 2.15 Antarmuka Bahasa Pemrograman Delphi 7.0

Ada beberapa bagian pada IDE. Bagian–bagian IDE antara lain :

a. *Menu* dimana semua perintah yang diperlukan selama merancang danmembangun program aplikasi tersedia dalam *menu bar*, yang terletak di bagian atas jendela utama Delphi. Pada menu terdapat : *File*,

Edit, *Search*, *View*, *Project*, *Run*, *Component*, *Database*, *Tool*, dan menu *help*.

b. *Speedbar* merupakan sekumpulan tombol yang digunakan untuk mengakses beberapa perintah dalam menu. Misalnya perintah untu membuka project, membuka file, menyimpan file, dan sebagainya.

c. *Component Palette* adalah *tool* yang berupa kumpulan *tab*, dimana pada setiap *tab* atau halaman memuat berbagai tombol komponen yang digunakan sebagai elemen antarmuka program aplikasi.

d. *Object Inspector* merupakan penghubung antara tampilan aplikasi dengan kode program yang menjadikan aplikasi yang kita buat berjalan dengan baik. Ada dua hal yang terpenting dalam *Object Inspector* yaitu penetapan *property* bagi komponen–komponen dalam *form*, dan penetapan prosedur – prosedur penanganan suatu kejadian.

e. *Form* adalah jendela yang merupakan komponen dasar dari sebuah aplikasi. *Form* dapat berfungsi sebagai jendela bagi program aplikasi atau sebuah kotak dialog.

f. *Editor Code* merupakan jendela editor yang penuh dengan rencana. Editor ini dapat mengakses dan memodifikasi kode–kode program aplikasi. Pada editor ini merupakan *reserved* wordnya Delphi.

2.7.3 Struktur Program Delphi

Program aplikasi yang dibuat dengan menggunakan Delphi tersimpan sebagai sebuah *project*. *Project* ini merupakan kumpulan dari *file* unit yang menyusun aplikasi. *Project* aplikasi Delphi terdiri dari :

a. Project

Dimana tersimpan dalam ekstensi .DPR. Dalam satu *project* aplikasi hanya bisa terdapat satu *file project*. Setiap memulai *project* baru, Delphi akan membentuk sebuah *file project* dan akan meng-*update*-nya selama proses pembuatan *project* tersebut.

Untuk *project* yang bernama *project*1, dimana di dalamnya terdapat *form* bernama *form*1 (yang *source code*-nya tersimpan sebagai *file* unit1.pas), Delphi membuat *source code*-nya seperti di bawah ini.

program *project*1

uses

Forms,

Unit1 in 'Unit1.pas' {*Form*1};

{$R*.RES}

begin

Application.create(T*form*, Form1);

Application.Run(*Form*1);

end.

File ini dimulai dengan reserved word program, diikuti nama *file* (dimana pada contoh di atas *project* ini bernama ***project*1**). Setiap kali menambahkan *form* atau unit baru kedalam *project*, secara otomatis Delphi akan menambahkannya kedalam *file project*, tepatnya pada bagian **uses**.

b. Form

Form adalah titik utama dari suatu aplikasi Delphi. Fungsi dari *form* itu sendiri yaitu meng-*edi*t *properti*-nya atau menuliskan kode.form tersimpan dalam dua file, yaitu :

1. *File form* (.DFM), menyimpan image binary dari *form* yang dilakukan seperti mengubah tinggi, lebar, warna, dan *form*, dan lain–lain yang berkaitan dengan tampilan *form*.
2. *File* Unit (.PAS), menyimpan source kode dari *file* .DFM. Didalam *file* .PAS inilah dituliskan kode–kode event-handler, yang menentukan tentang bagaimana *form* serta komponen–komponen didalamnya harus bereaksi saat aplikasi dijalankan.

Bila menambahkan *form* baru dalam *project*, secara otomatis Delphi akan membuat *file* unit untuk *form* baru tersebut, dan menambahkannya kebagian *uses* dalam *file project*.

c. Unit

Unit dalam blok utama pembentuk aplikasi Delphi. *File–file* unit ini berisi kode sumber bagi elemen-elemen aplikasi Delphi yang dibuat. Ada *file* unit merupakan pasangan *file form* .DPM, dan ada pula *file* unit milik suatu *project*.

Dalam pembuatan aplikasi, dapat ditambahkan pula f*orm-form* dan *file-file* unit yang terkait. Seperti *file-file* yang ditulis dalam Borland Pascal versi sebelumnya, atau *file-file* yang tidak dibuat oleh Delphi, tetapi menggunakan *library* atau *source* lain. *File project* menggabungkan semua *file* ini bersama-sama, sehingga Delphi mengetahui apa yang harus dikompilasi dan di-*link* atau membentuk *file executable* (.EXE) atau *dynamic link library*.

2.8 Microsoft Access XP

Microsoft access XP merupakan salah satu software yang memungkinkan kita membuat Database. Dimana kita dapat mengelola seluruh data ke dalam file Database. Dengan Microsoft Access XP kita dapat merancang, membuat dan mengelola Database dengan mudah. Microsoft Access Xp merupakan pengembangan dari Access sebelumnya, dengan harapan program aplikasi Database ini lebih mudah dalam pemakaiannya dibandingkan dengan versi sebelumnya, selain itu Microsoft Access XP ini dapat diintegrasikan dengan program aplikasi Microsoft Office XP itu Sendiri ataupun dengan program aplikasi yang lain.

Komponen–komponen yang terlibat dalam pembuatan suatu Database diantaranya adalah *table*. *Table* merupakan kumpulan data yang merupakan komponen utama dari sebuah Database.