

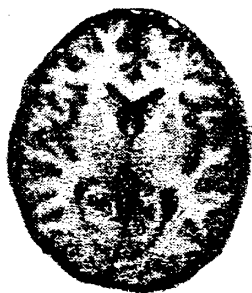
BAB II

LANDASAN TEORI

2.1 Stroke Infark

Stroke infark merupakan salah satu jenis penyakit stroke, yaitu berkurangnya aliran darah ke otak bagian tertentu dari otak yang akhirnya menyebabkan kematian sel otak.

Dari gambar 2.1 hasil MRI dapat terlihat gambar irisan otak baik pada kondisi T1 maupun kondisi T2. Pada irisan otak akan terlihat adanya tulang otak, cairan otak dan organ otak yang bentuknya berliku-liku. Pada kondisi T1 cairan otak berwarna hitam dengan otak berwarna agak keputihan. Namun bila pasien menderita infark, terdapat bagian yang berwarna hitam pada organ otaknya (bukan cairan otak). Hal ini menggambarkan adanya penyumbatan pembuluh darah pada organ otak dan biasanya organ otak sekitarnya menjadi rusak akibat bagian yang dikelilingi bagian yang terkena infark tidak mendapatkan makanan (konsumsi darah). Berkebalikan dengan kondisi T2 di mana cairan otak terlihat berwarna putih dan bagian otak berwarna keabu-abuan (agak gelap).



T1



T2

Gambar 2.1 Contoh Gambar Irisan Otak Normal



Gambar 2.2 Contoh Gambar Irisan Otak yang Terkena Infark

(Joesoef, 2004)

2.2 Magnetic Resonance Imaging (MRI)

MRI adalah suatu alat kedokteran di bidang pemeriksaan diagnostik radiologi, yang menghasilkan rekaman gambar potongan penampang tubuh organ manusia dengan menggunakan medan magnet berkekuatan antara tesla (1 tesla = 10000 Gauss) dan resonansi getaran terhadap inti atom hidrogen.

Beberapa faktor kelebihan yang dimilikinya, terutama kemampuannya membuat potongan koronal, sagital, aksial dan oblik tanpa banyak memanipulasi posisi tubuh pasien sehingga sangat sesuai untuk diagnostik jaringan lunak. Teknik penggambaran MRI relatif kompleks karena gambaran yang dihasilkan tergantung pada banyak parameter. Bila pemilihan parameter tersebut tepat, kualitas gambar MRI dapat memberikan gambaran detail tubuh manusia dengan perbedaan yang kontras, sehingga anatomi dan patologi jaringan tubuh dapat dievaluasi secara teliti.

Untuk menghasilkan gambaran MRI dengan kualitas yang optimal sebagai alat diagnostik, maka harus memperhitungkan hal-hal yang berkaitan dengan teknik penggambaran MRI, antara lain :

- a. Persiapan pasien serta teknik pemeriksaan pasien yang baik.
- b. Kontras yang sesuai dengan tujuan pemeriksaanya.
- c. Artefak pada gambar, dan cara mengatasinya.
- d. Tindakan penyelamatan terhadap keadaan darurat.

Selanjutnya MRI bila ditinjau dari tipenya terdiri dari :

- a. MRI yang memiliki kerangka terbuka (*open gantry*) dengan ruang yang luas
- b. MRI yang memiliki kerangka (*gantry*) biasa yang berlorong sempit.

Sedangkan MRI bila ditinjau dari kekuatan magnetnya terdiri dari 3 jenis, antara lain:

- a. MRI Tesla tinggi (*High Field Tesla*) memiliki kekuatan di atas 1 – 1,5 T.
- b. MRI Tesla sedang (*Medium Field Tesla*) memiliki kekuatan 0,5 – T.
- c. MRI Tesla rendah (*Low Field Tesla*) memiliki kekuatan di bawah 0,5 T.

Sebaiknya suatu rumah sakit memilih MRI yang memiliki tesla tinggi karena alat tersebut dapat digunakan untuk tehnik *Fast Scan* yaitu suatu tehnik yang memungkinkan 1 gambar irisan penampang dibuat dalam hitungan detik, sehingga kita dapat membuat banyak irisan penampang yang bervariasi dalam waktu yang sangat singkat. Dengan banyaknya variasi gambar membuat suatu lesi menjadi menjadi lebih spesifik.

Secara garis besar instrumen MRI terdiri dari:

- a. Sistem magnet yang berfungsi membentuk medan magnet. Agar dapat mengoperasikan MRI dengan baik, kita perlu mengetahui tentang : tipe

magnet, efek medan magnet, *magnet shielding*, *shimming coil* dari pesawat MRI tersebut.

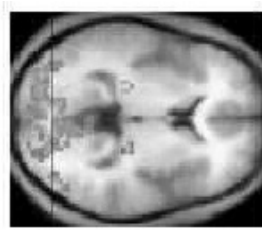
- b. Sistem pencitraan berfungsi membentuk citra yang terdiri dari 3 buah kumparan koil, yaitu :
1. Gradien koil X, untuk membuat citra potongan sagital.
 2. Gardien koil Y, untuk membuat citra potongan koronal.
 3. Gradien koil Z untuk membuat citra potongan aksial .

Bila gradien koil X, Y dan Z bekerja secara bersamaan maka akan terbentuk potongan oblik.

- c. Sistem frekuensi radio berfungsi membangkitkan dan memberikan radio frekuensi serta mendeteksi sinyal.
- d. Sistem komputer berfungsi untuk membangkitkan urutan pulsa, mengontrol semua komponen alat MRI dan menyimpan memori beberapa citra.
- e. Sistem pencetakan citra, berfungsinya untuk mencetak gambar pada film *rongent* atau untuk menyimpan citra.



Gambar (a)
potongan koronal



Gambar (b)
potongan aksial



Gambar (c)
potongan sagital

Gambar 2.3 Contoh Potongan Gambar Hasil MRI

Prinsip Dasar MRI

Dalam mempelajari mengenai MRI terdapat prinsip dasar. Struktur atom hidrogen dalam tubuh manusia saat di luar medan magnet mempunyai arah yang acak dan tidak membentuk keseimbangan. Kemudian saat diletakkan dalam alat MRI, maka atom H sejajar dengan arah medan magnet. Demikian juga arah spinning sejajar dengan arah medan magnet. Saat diberikan frekuensi radio, maka atom H mengabsorpsi energi dari frekuensi radio tersebut. Akibatnya dengan bertambahnya energi, atom H mengalami pembelokan, sedangkan besarnya pembelokan arah, dipengaruhi oleh besar dan lamanya energi radio frekuensi yang diberikan. Sewaktu radio frekuensi dihentikan maka atom H akan sejajar kembali dengan arah medan magnet. Pada saat kembali inilah, atom H akan memancarkan energi yang dimilikinya. Kemudian energi yang berupa sinyal tersebut dideteksi dengan detektor yang khusus dan diperkuat. Selanjutnya komputer mengolah dan merekonstruksi citra berdasarkan sinyal yang diperoleh dari berbagai irisan.

Selain menggunakan MRI, citra otak didapat menggunakan *Computed Tomography* (CT) Scan. Tetapi ada beberapa kelebihan MRI dibandingkan dengan pemeriksaan CT scan yaitu:

- a. MRI lebih unggul untuk mendeteksi beberapa kelainan pada jaringan lunak otak, sumsum tulang serta muskuloskeletal.
- b. Mampu memberi gambaran detail anatomi dengan lebih jelas.
- c. Mampu melakukan pemeriksaan fungsional seperti pemeriksaan difusi, perfusi dan spektroskopi yang tidak dapat dilakukan dengan CT scan.
- d. Mampu membuat gambaran potongan melintang, tegak, dan miring tanpa merubah posisi pasien.

e. MRI tidak menggunakan radiasi pengion.

(Notosiswoyo, 2004)

2.3 Pengenalan Pola

Pola adalah entitas yang terdefinisi dan dapat diidentifikasi melalui ciri-cirinya (features). Ciri-ciri tersebut digunakan untuk membedakan suatu pola dengan pola lainnya. Ciri yang bagus adalah ciri yang memiliki daya pembeda yang tinggi, sehingga pengelompokan pola berdasarkan ciri yang dimiliki dapat dilakukan dengan keakuratan yang tinggi, contohnya:

Tabel 2.1 Contoh Pola dan Ciri-Cirinya

No	Pola	Ciri
1	Huruf	Tinggi, tebal, titik sudut, lengkungan garis
2	Suara	Amplitudo, frekuensi, nada, intonasi, warna nada.
3	Tanda tangan	Panjang, kerumitan, tekanan.
4	Sidik jari	Lengkungan, jumlah garis

Ciri pada suatu pola diperoleh dari hasil pengukuran terhadap objek uji. Khusus pada pola yang terdapat di dalam citra, ciri-ciri yang dapat diperoleh berasal dari informasi:

- a. Spasial : intensitas pixel, histogram
- b. Tepi: arah, kekuatan
- c. Kontur: garis, elips, lingkaran
- d. Wilayah/bentuk: keliling, luas, pusat massa
- e. Hasil transformasi Fourier: frekuensi

2.3.1 Komponen Sistem Pengenalan Pola

Sistem pengenalan pola dasar terdiri dari

a. Sensor

Sensor digunakan untuk menangkap objek yang ciri atau fiturnya akan diekstraksi.

b. Mekanisme pre-processing

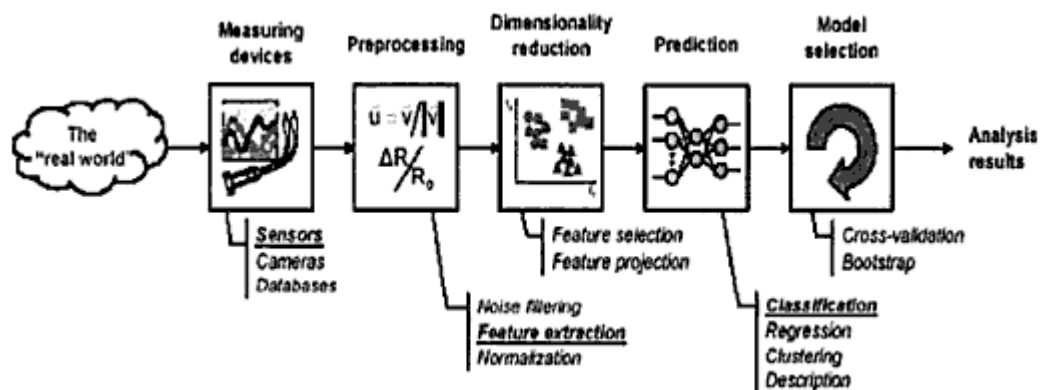
Mekanisme pengolahan objek yang ditangkap oleh sensor biasanya digunakan untuk mengurangi kompleksitas ciri yang akan dipakai untuk proses klasifikasi.

c. Mekanisme pencari fitur

Bagian ini digunakan untuk mengekstraksi ciri yang telah melalui tahapan preprocessing untuk memisahkannya dari fitur-fitur pada objek yang tidak diperlukan dalam proses klasifikasi.

d. Algoritma pemilah

Pada tahapan ini proses klasifikasi dilakukan dengan menggunakan algoritma klasifikasi tertentu. Hasil dari tahapan ini adalah klasifikasi dari objek yang ditangkap ke dalam kriteria-kriteria yang telah ditentukan.



Gambar 2.4. Komponen-komponen Sistem Pengenalan Pola

2.3.2 Fitur

Fitur adalah segala jenis aspek pembeda, kualitas atau karakteristik. Fitur bisa berwujud simbolik (misalnya warna) atau numerik (misalnya tinggi). Kombinasi dari d -buah fitur dinyatakan sebagai vektor kolom dimensi- d dan disebut vektor fitur. Ruang dimensi- d yang dibentuk oleh vektor fitur disebut ruang fitur. Objek dinyatakan sebagai sebuah titik di dalam ruang fitur. Penggambaran demikian disebut sebagai diagram hambur (*scatter plot*). Kualitas dari vektor fitur dilihat dari kemampuannya membedakan objek yang berasal dari kelas yang berbeda-beda di mana berlaku ketentuan berikut:

- a. Objek dalam kelas yang sama harus memiliki nilai vektor fitur yang sama.
- b. Objek dalam kelas yang berbeda harus memiliki nilai vektor fitur yang berlainan.

2.3.3 Pola

Pola adalah komposit atau gabungan dari fitur yang merupakan sifat dari sebuah objek. Dalam klasifikasi, pola berupa sepasang variabel (x, ω) , di mana:

- a. x adalah sekumpulan pengamatan atau fitur (vektor fitur).
- b. ω adalah konsep di balik pengamatan (label).

2.3.4 Pemilah

Tugas dari pemilah adalah menyekat ruang fitur ke dalam daerah-daerah yang dilabeli dengan kelas. Garis batas antar daerah keputusan disebut sebagai perbatasan keputusan. Pemilahan vektor fitur x meliputi penentuan daerah keputusan yang sesuai dan pengelompokkan x ke dalam kelas ini.

2.3.5 Pendekatan Pengenalan Pola

Aplikasi pengenalan pola dapat dibuat dengan beberapa pendekatan. Ada pendekatan yang menggunakan basis statistikal untuk menghasilkan pola. Pendekatan lainnya menggunakan struktur dari pola yang menyediakan informasi fundamental untuk pengenalan pola. Pendekatan yang lain adalah dengan membangun dan melatih suatu arsitektur yang secara akurat mengasosiasikan input pola tertentu dengan respon yang diharapkan. Suatu masalah bisa diselesaikan dengan salah satu atau beberapa pendekatan tersebut.

a. Pendekatan pengenalan pola statistikal (StatPR)

Pengenalan pola statistikal memiliki asumsi suatu baris statistik untuk algoritma klasifikasi. Sekelompok karakteristik pengukuran yang menunjukkan fitur diekstraksi dari data input dan digunakan untuk menentukan setiap vektor fitur ke dalam salah satu kelas. Fitur diasumsikan dihasilkan secara natural sehingga model yang bersangkutan adalah suatu state of nature atau kelas-kelas probabilitas atau fungsi kepadatan probabilitas (*probablity density function*) yang telah dikondisikan. Dengan demikian kesimpulannya adalah sebagai berikut:

1. pola dipilah berdasarkan model statistik dari fitur.
2. model statistik didefinisikan sebagai sebuah keluarga dari fungsi kerapatan peluang bersyarat kelas $\Pr(x|c_i)$ – peluang vektor fitur x jika diberikan kelas c_i .

b. Pendekatan pengenalan pola sintatik (SyntPR)

Untuk pendekatan sintatik dapat dijelaskan sebagai berikut:

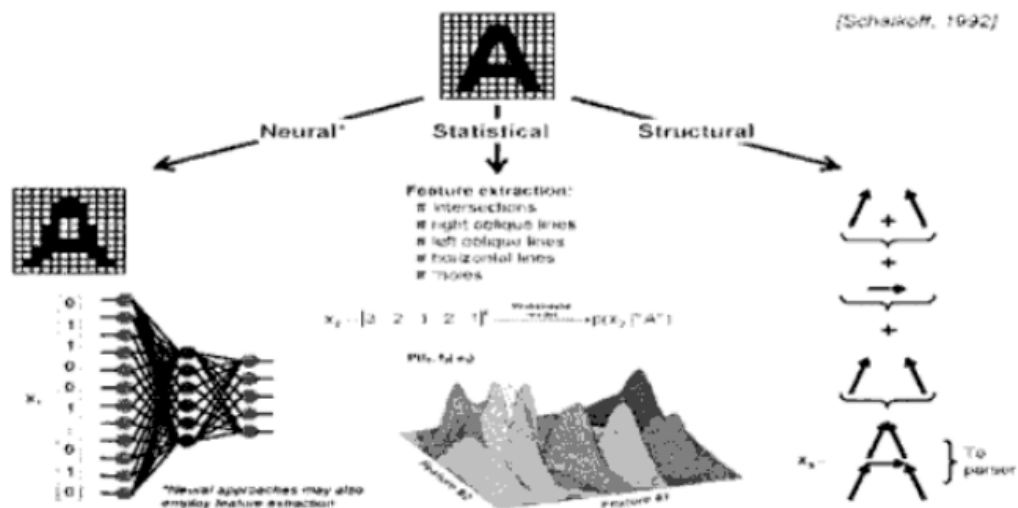
1. Pola dipilah berdasarkan kerserupaan ukuran struktural.

2. Pengetahuan direpresentasikan secara formal grammar atau deskripsi relasional (graf).
3. SyntPR dipakai tidak hanya untuk pemilahan, tetapi juga untuk dekripsi.
4. Biasanya SyntPR memformulasikan deskripsi hierarkis dari pola kompleks yang tersusun dari pola bagian yang lebih sederhana,

c. Pendekatan pengenalan pola neural (NeuroPR)

Pendekatan ini menggunakan metode jaringan syaraf tiruan untuk mengidentifikasi pola. Pendekatan ini dapat dijelaskan sebagai berikut:

1. Pemilahan dilakukan berdasarkan tanggapan suatu neuron jaringan pengolah sinyal (neuron) terhadap stimulus masukan (pola).
2. Pengetahuan disimpan dalam sambungan antar neuron dan kekuatan pempombot sinaptik.
3. NeuroPR dapat dilatih, non-algoritmik, strategi black box.
4. neuroPR sangat menarik karena dengan jumlah lapisan dan neuron secukupnya. Jaringan syaraf tiruan dapat membentuk semua jenis daerah keputusan yang rumit sekalipun.



Gambar 2.5. Ilustrasi 3 Pendekatan Pengenalan Pola

Jangkauan dari aplikasi-aplikasi sistem pengenalan pola cukup beragam, di antaranya:

- Voice recognition*, beberapa sistem rahasia menggunakan pengenalan suara sebagai kunci bagi pengguna sistemnya.
- Fingerprint identification*, sistem pengenalan sidik jari telah dipakai secara luas sebagai pengganti password atau pin untuk mengakses sistem komputer tertentu.
- Face identification*, badan penegak hukum sedang mengembangkan sistem untuk mengidentifikasi para buronan dengan melakukan scanning pada sejumlah besar data wajah para pelaku yang sudah di-databasekan berdasarkan foto pelaku kejahatan tersebut.
- Handwriting identification*, aplikasi perbankan menggunakan pengenalan tulisan untuk membuktikan pelaku transaksi adalah orang yang benar-benar berhak.

- e. *Optical Character Recognition (OCR)*. OCR saat ini digunakan secara luas pada *took retail* untuk meningkatkan produktivitas pada konter pengecekan barang.
- f. *Robot Vision*. Banyak aplikasi robotik menggunakan pengenalan pola untuk mengenali objek tertentu pada lingkungan yang unik.

(Fatta, 2009)

2.4 Pengolahan Citra

Pengolahan citra (*image processing*) merupakan suatu proses dengan masukan berupa citra dan hasilnya juga berupa citra. Pengolahan citra ini bertujuan untuk memperbaiki kualitas citra, dimana citra yang dihasilkan dapat memberikan informasi secara jelas dan informasi ciri citra tersebut sudah berupa numerik. Adapun tahap-tahap dalam pengolahan citra adalah sebagai berikut:

2.4.1 Grayscale

Proses awal yang banyak dilakukan dalam *image processing* adalah mengubah citra berwarna menjadi citra *grayscale*. Hal ini digunakan untuk menyederhanakan model citra. Di mana citra berwarna yang terdiri dari 3 layer matriks, yaitu R-layer, G-layer dan B-layer diubah menjadi 1 layer matriks *grayscale* dan hasilnya adalah citra *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan. Untuk mengubah citra berwarna yang mempunyai nilai matriks masing-masing R, G dan B menjadi *grayscale* dengan nilai gray, maka konversi dapat dilakukan dengan mengambil rata-rata dari nilai R, G, dan B sehingga dapat dituliskan menjadi :

$$Gray = \frac{R + G + B}{3} \quad (2.1)$$

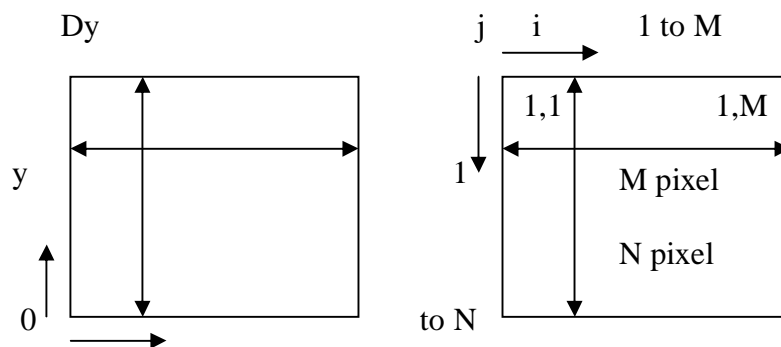
Dimana :

R : Tingkat intensivitas warna merah

G : Tingkat intensivitas warna hijau

B : Tingkat intensivitas warna biru

Secara digital suatu greyscale dapat direpresentasikan dalam bentuk array dua dimensi. Tiap elemen pada array tersebut menunjukkan intensitas (grayscale) dari image pada posisi koordinat yang bersesuaian. Apabila suatu image direpresentasikan dalam 8 bit maka berarti pada image terdapat 2^8 atau 256 level *grayscale* (biasanya bernilai 0 – 255). Dimana 0 menunjukkan level intensitas yang paling gelap dan 255 menunjukkan intensitas paling terang. Tiap elemen pada array diatas disebut sebagai *picture elemen* atau sering dikenal sebagai *pixel*. Dengan melakukan perubahan pada intensitas masing-masing *pixel* maka representasi image secara keseluruhan akan berubah. Image yang dinyatakan dengan NxM matriks mempunyai intensitas tertentu pada *pixel* tertentu. Posisi *picture elemen* (i, j) dan koordinat *pixel* (x, y) berbeda dapat dilihat pada Gambar 2.4.



Gambar 2.6 Matriks *Image*

2.4.2 Thresholding

Thresholding digunakan untuk mengatur jumlah derajat keabuan pada citra sesuai keinginan. Misalnya, jika ingin menggunakan derajat keabuan 16, maka hanya membagi derajat keabuan dengan nilai 16. Pada dasarnya, proses thresholding merupakan proses pengubahan kuantitas citra, sehingga untuk melakukan thresholding dengan derajat keabuan dapat digunakan rumus :

$$\begin{aligned} x &= b \cdot \text{int}\left(\frac{wk}{b}\right) \\ b &= \text{int}\left(\frac{256}{a}\right) \end{aligned} \quad (2.2)$$

Dimana :

wk adalah nilai derajat keabuan sebelum thresholding

x adalah nilai derajat keabuan setelah thresholding

a adalah nilai threshold

2.4.3 Histogram Equalization

Histogram equalisasi merupakan suatu proses perataan histogram, dimana distribusi nilai derajat keabuan pada suatu citra dibuat rata dan ditujukan untuk memperjelas gambar. Pada histogram equalization, histogram diratakan berdasarkan suatu fungsi linier. Nilai dari histogram equalization sebagai berikut :

$$w = \frac{c}{n} \frac{th}{n_x n_y} \quad (2.3)$$

Dimana :

w adalah nilai keabuan hasil histogram equalization

C_w adalah histogram kumulatif dari w

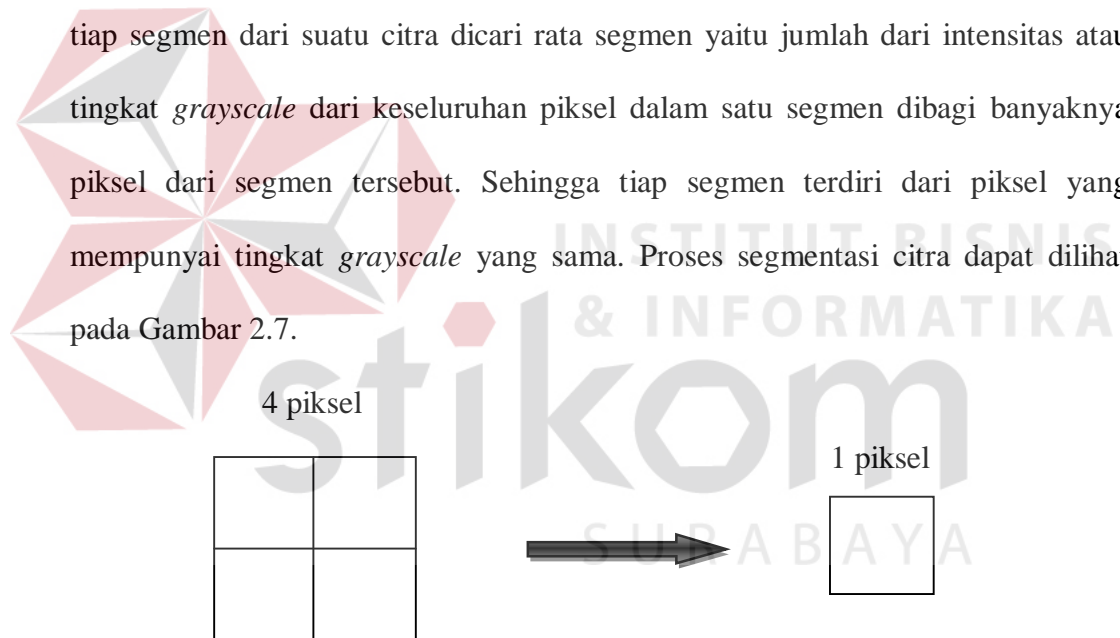
th adalah threshold derajat keabuan (256)

n_x dan n_y adalah ukuran gambar

2.4.4 Segmentasi

Secara umum segmentasi citra dapat diartikan membagi citra menjadi bagian-bagian penyusunnya atau segmen-segmen yang lebih kecil sehingga diharapkan untuk pengolahan datanya dapat menjadi lebih cepat. Hasil dari tahap segmentasi citra ini berupa data piksel yang menyusun batas dari suatu daerah atau semua titik dalam suatu daerah. Tahapan dalam proses segmentasi adalah,

tiap segmen dari suatu citra dicari rata segmen yaitu jumlah dari intensitas atau tingkat *grayscale* dari keseluruhan piksel dalam satu segmen dibagi banyaknya piksel dari segmen tersebut. Sehingga tiap segmen terdiri dari piksel yang mempunyai tingkat *grayscale* yang sama. Proses segmentasi citra dapat dilihat pada Gambar 2.7.



Gambar 2.7 Proses Segmentasi 4 piksel diwakili 1 piksel

2.4.5 Normalisasi

Normalisasi pada citra merupakan proses pada nilai intensitas tiap segmen dari citra agar bernilai 0 dan 1 dengan cara rata segmen dibagi dengan tingkat grayscale yang paling tinggi.

(Basuki, 2005)

2.5 Analisis Komponen Utama

Analisis Komponen Utama atau *Principle Component Analysis* (PCA) adalah teknik statistik untuk menyederhanakan kumpulan data banyak dimensi menjadi dimensi yang lebih rendah (*extraction feature*). PCA merupakan transformasi lineal ortogonal yang mentransformasi data ke sistem koordinat baru, sehingga keragaman terbesar dengan suatu proyeksi berada pada koordinat pertama (disebut prinsipal komponen pertama), keragaman terbesar kedua berada pada koordinat kedua dan seterusnya. Konsep penggunaan PCA meliputi perhitungan nilai-nilai simpangan baku, matriks kovarian, nilai karakteristik (*eigen value*) dan vektor karakteristik (*eigen vector*). PCA dapat menggunakan metode kovariansi atau korelasi. Jika diperlukan, data distandarisasi terlebih dahulu sehingga mendekati sebaran normal baku. Dalam hal ini digunakan metode kovariansi dengan algoritma berikut. Mengumpulkan data dalam bentuk matriks tingkat keabuan X berukuran $M \times N$. Misal x_1, x_2, \dots, x_M adalah vektor $N \times 1$:

(i) Menghitung rata-rata:
$$\bar{x} = \frac{1}{M} \sum_{i=1}^M x_i$$

(ii) Menghitung selisih *rata-rata*:
$$\Phi_i = x_i - \bar{x}$$

(iii) Menentukan matriks kovarian

Dari matriks $X = [\Phi_1 \ \Phi_2 \ \dots \ \Phi_M]$ (matriks $N \times M$),

Hitung kovarian:
$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = XX^T$$

(iv) Menentukan nilai karakteristik dan vektor karakteristik dari matriks kovarian

$$C : \lambda_1 > \lambda_2 > \dots > \lambda_N \quad \text{dan} \quad C : u_1, u_2, \dots, u_n$$

(v) Mengurutkan vektor karakteristik u dan nilai karakteristik λ dalam matriks diagonal dalam urutan menurun sesuai dengan nilai peluang kumulatif

terbesar untuk tiap vektor karakteristik sehingga diperoleh nilai-nilai karakteristik yang dominan.

2.6 Algoritma Eigen Image

2.6.1 Eigen Value

Salah satu tool penting dalam mendeskripsikan eigen value dari suatu matriks bujur sangkar adalah polinomial karakteristik: Jika λ adalah eigen value dari matriks A maka akan ekuivalen dengan persamaan linear $(A - \lambda I) v = 0$ (di mana I adalah matriks identitas) yang memiliki pemecahan non-zero v (suatu eigen vector), sehingga ekuivalen dengan determinan:

$$\det(A - \lambda I) = 0 \quad (2.4)$$

Fungsi $p(\lambda) = \det(A - \lambda I)$ adalah sebuah polinomial dalam λ karena determinan dihitung dengan sum of product. Semua eigen value dari suatu matriks A dapat dihitung dengan menyelesaikan persamaan $p_A(\lambda) = 0$. Jika A adalah matriks ukuran $n \times n$, maka p_A memiliki derajat n dan A akan memiliki paling banyak n buah eigen value.

2.6.2 Eigen Vector

Ketika eigen value λ diketahui, eigen vector bisa dicari dengan memecahkan:

$$(A - \lambda I) v = 0$$

Dalam beberapa kasus bisa dijumpai suatu matriks tanpa eigen value, misalnya A :

$$\begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}$$

di mana karakteristik polinomialnya adalah $\lambda^2 + 1$ sehingga eigen valuenya adalah bilangan kompleks i , $-i$. Eigen vector yang berasosiasi juga tidak riil. Perhatikan lagi contoh berikut ini. Jika diberikan matriks A:

$$\begin{pmatrix} 2 & 1 \\ 0 & 3 \end{pmatrix}$$

Maka polinomial karakteristiknya dapat dicari sebagai berikut:

$$\det \begin{pmatrix} (2 - \lambda) & -1 \\ 0 & (3 - \lambda) \end{pmatrix} = \lambda^2 - 5\lambda + 6 = 0$$

Ini adalah persamaan kuadrat. Akarnya adalah $\lambda = 2$ dan $\lambda = 3$.

Adapun eigen vector yang bisa diperoleh ada 2 buah. Eigen vector pertama dicari dengan mensubstitusikan $\lambda = 3$ ke dalam persamaan. Misalnya Y_0 adalah eigen vector yang berasosiasi dengan eigen value $\lambda = 3$. Set Y_0 dengan nilai

$$Y_0 = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix}$$

Substitusikan Y_0 dengan v pada persamaan:

$$(A - \lambda I)v = 0$$

Kita akan mendapatkan

$$(2 - 3) X_0 + -Y_0 = 0$$

$$0 + (3-3) Y_0 = 0$$

Kita bisa sederhanakan menjadi

$$- X_0 - Y_0 = 0$$

Atau bisa disederhanakan menjadi

$$Y_0 = - X_0$$

Sehingga eigen vector untuk eigen value = 3 adalah

$$Y_0 = \begin{pmatrix} X_0 \\ Y_0 \end{pmatrix} = \begin{pmatrix} X_0 \\ -X_0 \end{pmatrix} = X_0 \begin{pmatrix} 1 \\ -1 \end{pmatrix}$$

2.6.3 Eigen Image

Eigen image adalah kumpulan dari eigen vector yang digunakan untuk masalah *computer vision* pada pengenalan gambar, seperti pengenalan wajah manusia. Untuk menghasilkan eigen images, sekumpulan besar citra digital dari data gambar diambil pada kondisi pencahayaan yang sama dan kemudian dinormalisasi dan kemudian diolah pada resolusi yang sama (misalnya $m \times n$), dan kemudian diperlakukan sebagai vector dimensi mn di mana komponennya diambil dari nilai dari pikselnya.

2.6.4 Algoritma Eigen Image dengan PCA

Eigen image adalah salah satu pengenalan gambar yang didasarkan pada *Principal Component Analysis* (PCA) yang dikembangkan di MIT. Training image direpresentasikan dalam sebuah vektor flat (gabungan vektor) dan digabung bersama – sama menjadi sebuah matriks tunggal. Eigen image dari masing-masing citra kemudian diekstraksi dan disimpan dalam file temporary atau database. Test image yang masuk didefinisikan juga nilai eigen image-nya dibandingkan dengan eigen image dari image dalam database atau file temporary.

Berikut adalah algoritma perhitungan eigen image dengan menggunakan PCA :

1. Mengumpulkan setiap citra otak $I_1, I_2, I_3, \dots, I_M$ (training image). Setiap citra harus memiliki orientasi dan ukuran yang sama, citra otak terletak di tengah gambar dan memiliki ukuran 183×183 piksel.
2. Ubah matriks citra I_i menjadi vektor Γ_i .
3. Hitung rata-rata citra Ψ .

$$\Psi = \frac{1}{M} \sum_{i=1}^M \Gamma_i \quad (2.5)$$

4. Kurangi masing-masing vektor citra Γ_i dengan rata-rata.

$$\Phi_i = \Gamma_i - \Psi \quad (2.6)$$

5. Hitung matriks kovarians C.

$$C = \frac{1}{M} \sum_{n=1}^M \Phi_n \Phi_n^T = AA^T \quad (\text{matriks } N^2 \times N^2) \quad (2.7)$$

$$\text{where } A = [\Phi_1 \Phi_2 \dots \Phi_M] \quad (\text{matriks } N^2 \times N^2)$$

6. Menggunakan metode PCA pada perhitungan matriks kovarian C untuk mencari eigen vector dan eigen value. Matriks kovarian C ordo $N^2 \times N^2$ tidak efektif dalam perhitungan, sehingga metode PCA digunakan.

$$L = A^T A \quad L_{m,m} = \Phi_m^T \Phi_m \quad (2.8)$$

$$u_l = \sum_{k=1}^M v_{lk} \Phi_k \quad l = 1, 2, \dots, M \quad (2.9)$$

Setelah menemukan eigen vector dan eigen value dari data citra pada proses training, maka eigen value tersebut akan digunakan untuk pengenalan citra.

Berikut algoritma pengenalan citra uji coba:

1. Normalisasi data citra, dengan asumsi citra otak Γ mempunyai orientasi pada posisi tengah dan ukuran citra yang sama dengan data training.

$$\Gamma : \Phi = \Gamma - \Psi$$

2. Melakukan proyeksi ke eigen space. Citra uji ditransformasikan ke komponen eigen image. Hasil bobot disimpan ke dalam vector bobot

$$\Omega_{NEM}^T$$

$$\omega_k = u_k^T (\Gamma_{new} - \Psi) \quad k = 1, 2, \dots, M' \quad (2.10)$$

$$\Omega_{new}^T = [\omega_1 \ \omega_2 \ \dots \ \omega_{M'}] \quad (2.11)$$

3. Euclidian distance antara 2 vektor bobot $d(\Omega_i, \Omega_j)$ digunakan untuk mengukur kemiripan antara dua buah gambar, i dan j . Berikut adalah formula dari perhitungan Euclidian distance.

$$x = [a_1(x), a_2(x), \dots, a_n(x)]$$

$$d(x_i, x_j) = \sqrt{\sum_{r=1}^n (a_r(x_i) - a_r(x_j))^2} \quad (2.12)$$

Keterangan :

AA^T dapat memiliki hingga N^2 eigen value dan eigen vector

$A^T A$ dapat memiliki hingga M eigen value dan eigen vector

I matriks citra

$N \times N$ ukuran ordo dari matriks I

Γ training set

Γ_i citra otak ke- i dari training set

Γ_{new} citra uji coba(citra baru)

Ψ rata-rata matriks citra

$M = |\Gamma|$ jumlah eigen image

M' jumlah eigen image yang digunakan untuk pengenalan citra

C matriks kovarians

X^T X transpose (jika X adalah matriks)

u	eigen vector (eigen image)
λ	eigen value
ω_i	bobot ke-i
ω_i^T	vektor bobot dari citra ke-i

(Turk and Pentland, 1991)

2.7 Sistem Persamaan Linear

Sistem Persamaan Linear dengan metode iterasi digunakan untuk mencari nilai eigen vector seperti pada persamaan 2.8. Eigen vector didapat dengan mencari eigen value seperti pada persamaan 2.4. Berikut sistem persamaan linear yang terdiri dari n-persamaan dengan n variabel x_1, x_2, \dots, x_n dinyatakan dengan

$$\begin{aligned}
 a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1 \\
 a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2 \\
 &\vdots \\
 a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n
 \end{aligned} \quad (2.13)$$

Sistem (2.13) dapat diekspresikan dengan bentuk perkalian matriks.

Sistem persamaan linear dapat diselesaikan dengan metode langsung atau metode iterasi. Kedua metode tersebut mempunyai kelemahan dan keunggulan. Metode yang dipilih akan menentukan keakuratan penyelesaian sistem tersebut. Dalam kasus tertentu, yaitu sistem yang besar, metode iterasi lebih cocok digunakan. Dalam menentukan penyelesaian sistem persamaan linear, metode iterasi menggunakan algoritma secara rekursif. Algoritma tersebut dilakukan sampai diperoleh suatu nilai yang konvergen dengan toleransi yang diberikan. Ada dua metode iterasi yang sering digunakan, yaitu metode Jacobi dan metode Gauss-

Seidel. Metode Jacobi dikenalkan oleh Carl Jacobi (1804-1851) dan metode Gauss-Seidel dikenalkan oleh Johann Carl Friedrich Gauss (1777-1855) dan Philipp Ludwig von Seidel (1821-1896).

2.8 Metode Jacobi

Persamaan ke- i dalam sistem persamaan 2.13 dinyatakan sebagai berikut :

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{ii}x_i + \dots + a_{in}x_n = b_i \quad \text{di mana } i = 1, 2, 3, \dots, n \quad (2.14)$$

Persamaan (2.14) dapat diekspresikan sebagai:

$$a_{ii}x_i + \sum_{j=1, j \neq i}^n a_{ij}x_j = b_i \quad (2.15)$$

Dari persamaan (2.15) dapat diperoleh penyelesaian persamaan ke- i yaitu:

$$x_i = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j \right] \quad (2.16)$$

Dengan demikian, algoritma metode Jacobi diekspresikan sebagai :

$$x_i^{(k+1)} = \frac{1}{a_{ii}} \left[b_i - \sum_{j=1, j \neq i}^n a_{ij}x_j^{(k)} \right], \quad \text{dimana } k = 0, 1, 2, \dots \quad (2.17)$$

Untuk menyelesaikan sistem persamaan linear dengan metode Jacobi diperlukan suatu nilai pendekatan awal yaitu $x^{(0)}$. Nilai $x^{(0)}$ biasanya tidak diketahui dan dipilih $x^{(0)} = 0$.

(Nugoroho, 2011)