BAB II
LANDASAN TEORI

2.1. Pendarahan Otak

Pendarahan otak merupakan tipe stroke yang disebabkan oleh pecahnya arteri di dalam otak sehingga mengakibatkan terjadinya pendarahan lokal pada jaringan di sekelilingnya. Pendarahan yang dapat terjadi di dalam otak biasanya terjadi antara otak dan selaput-selaput yang menutupinya, antara lapisan-lapisan dari penutup otak, maupun antara tengkorak dan penutup dari otak. Awal mula pendarahan otak biasanya disebabkan dengan terjadinya cedera pada kepala, meliputi cedera kepala ringan (CKR), cedera kepala sedang (CKS), dan cedera kepala berat (CKB). Selain itu, tekanan darah tinggi yang terjadi selama bertahun-tahun dapat menyebabkan dinding pembuluh darah melemah. Jika tidak diobati, tekanan darah tinggi bisa menjadi penyebab utama terjadinya pendarahan otak. *Aneurisma* atau makin lemahnya dinding pembuluh darah yang mengalami pembengkakan. Jika sudah parah, pembuluh darah akan meledak dan membanjiri otak dengan darah sehingga menimbulkan stroke.

Adanya pendarahan pada otak. Kelainan pembuluh darah atau *malaformasi arteriovenosa*, yaitu lemahnya pembuluh darah di dalam dan di sekitar otak yang didapat seseorang sejak lahir. Penyakit hati yang biasanya berhubungan dengan peningkatan terjadinya pendarahan internal. Kelainan darah atau kelainan pendarahan, seperti *hemofilia* dan anemia sel sabit yang berkontribusi terhadap terjadinya penurunan kadar trombosit darah. Gejala gejala yang sering dialami oleh penderita pendarahan otak adalah sakit kepala hebat yang datang secara mendadak. Mendadak mengalami kesemutan atau kelumpuhan di wajah, lengan, atau kaki. Mata mengalami kesulitan melihat, baik pada salah satu atau kedua-duanya. Sulit menelan makanan. Susah mengendalikan koordinasi tubuh dan hilang keseimbangan.

Sering merasa mual, muntah-muntah, hilang kesadaran, lesu, mengantuk, dan tidak sadar akan keadaan di sekitarnya. Mengalami masalah terkait kemampuan bahasa, baik ketika menulis, bicara, membaca, atau memahami sesuatu. Mengalami kebingungan atau mengigau. Penderita pendarahan otak perlu penanganan medis secara serius (Taylor, 2013). Tidak ada klasifikasi dalam pendarahan otak, namun jika pendarahannya lebih dari 30 CC perlu segera ditangani dengan operasi.

2.2. Citra

Citra *(image)* merupakan gambar pada bidang dwimatra (dua dimensi). Citra merupakan fungsi *continue* dari intensitas cahaya pada bidang dua dimensi, maksudnya sumber cahaya menerangi objek, objek memantulkan kembali sebagian dari berkas cahaya tersebut kemudian pantulan cahaya ini ditangkap oleh alat-alat optik, misalnya mata manusia, kamera, pemindai *(scanner)*, dan sebagainya sehingga bayangan objek yang disebut citra tersebut terekam (Munir, 2004). Citra sebagai keluaran dari suatu sistem perekaman data dapat bersifat:

a. Optik berupa foto
b. Analog berupa sinyal video seperti gambar pada monitor televisi
c. Digital yang dapat langsung disimpan pada suatu pita magnetik

2.2.1. Citra Analog

Menurut (Positron, 2012), citra analog adalah citra yang bersifat kontinu, seperti gambar pada monitor televisi, foto sinar-X, foto yang tercetak dikertas foto, lukisan, pemandangan alam, hasil *CT Scan*, gambar-gambar yang terekam pada pita kaset, dan lain sebagainya. Citra analog tidak dapat direpresentasikan dalam komputer sehingga tidak bisa diproses di komputer secara langsung.

2.2.2. Citra Digital

Menurut (Sutoyo, 2009), citra digital merupakan citra yang di dapat di olah oleh komputer. Citra digital dapat di definisikan sebagai fungsi f (x, y)

berukuran m baris dan kolom dengan *x* dan *y* adalah koordinat spesial dan amplitudo f di titik koordinat (x,y) merupakan hasil kali dari jumlah cahaya yang mengenai objek *(ilumination)* dengan derajat kemampuan objek tersebut memantulkan cahaya *(replection).* Sebuah citra mempunyai elemen terkecil yang di sebut *picture elements (pixel).* Pixel ini berisi angka-angka yang menunjukan itensitas warna-warna tertentu di dalam citra. Berdasarkan nilai pixelnya, citra digital di kelompokan dalam tiga jenis citra, yaitu sebagai berikut :

1. Citra Warna (*True Color*)

Menurut (Gusa, 2013), citra warna sering disebut juga citra RGB atau citra *true color* karena dapat mempresentasikan warna objek menyerupai warna aslinya dengan mengkombinasikan ketiga warna dasar yaitu merah/*red* (R), hijau/*green* (G), dan biru/*blue* (B). Tiap piksel memiliki tiga nilai kanal yang mewakili tiap komponen dasar citra.

2. Citra *Grayscale* (skala keabuan)

Menurut (Gusa, 2013), citra *grayscale* merupakan citra digital yang hanya memiliki satu nilai kanal pada setiap pixelnya. Dengan kata lain nilai bagian red, green, blue. Nilai tersebut digunakan untuk menunjukkan tingkat intensitas. Warna yang dimiliki citra *grayscale* adalah warna keabuan dengan berbagai tingkatan dari hitam hingga putih. Citra *grayscale* dapat diperoleh dari citra RGB nilai intensitas citra.

$$\text{Nilai Keabuan} = 0.2989 * \text{R} + 0.5870 * \text{G} + 0.1140 * \text{B}$$

Dimana,

R : Nilai intensitas warna merah

G : Nilai intensitas warna hijau

B : Nilai intensitas warna biru.

3. Citra Biner

Menurut (Gusa, 2013), Citra biner adalah citra yang hanya memiliki dua kemungkinan nilai piksel yaitu hitam (0) dan putih (1). Citra biner juga

disebut sebagai citra bw (*Black* and *White*) atau citra monokrom. Citra biner sering muncul sebagai hasil dari proses pengembangan *(thresholding).* Secara umum proses pengembangan citra Grayscale untuk menghasilkan citra biner sebagai berikut.

$$g\ x,y\ = \begin{matrix} 1\ jika\ f\ \ x,y\ \ \geq T \\ 0\ jika\ f\ \ \ x,y\ \ < T \end{matrix}$$

Dengan $g\ (x,y)$ adalah citra biner dari citra grayscale $f\ (x,y)$ dan T menyatakan nilai ambang *(thesholding).* Kualitas citra biner yang di hasilkan tergantung pada nilai T yang di gunakan.

2.2.3. Format Citra

Format citra didalam file itu ada beberapa jenis diantaranya *jpg, gif* dan *bmp*. *Jpg* dan *gif* umumnya di kompresi sedangkan *bmp* tidak, oleh karena itu kualitas gambar dari *bmp* lebih bagus dikarenakan tidak ada informasi yang hilang maka format berkas *bitmap* akan lebih mudah digunakan karena data asli lebih banyak dipertahankan (Munir, 2004). Terjemahan bebas dari *bitmap (BMP)* adalah pemetaan *bit*. Artinya, nilai intensitas pixel didalam citra dipetakan kesejumlah *bit* tertentu. Peta *bit* yang umum adalah 8, artinya setiap *pixel* panjangnya 8 *bit*. 8 *bit* ini merepresentasikan nilai intensitas *pixel*. Dengan demikian ada sebanyak 2 8 = 256 derajat keabuan, yang dimulai dari 0 sampai 255 (Munir, 2004).

Citra dalam format *BMP* ada tiga macam, yakni: citra biner, citra berwarna, dan citra hitam-putih *(grayscale).* Citra *biner* hanya mempunyai dua nilai keabuan, 0 dan 1. Oleh karena itu, 1 bit sudah cukup untuk merepresentasikan nilai piksel. Citra berwarna adalah citra yang lebih umum. Warna yang terlihat pada citra *bitmap* merupakan kombinasi dari tiga warna dasar, yaitu merah, hijau dan biru. Setiap pixel disusun oleh tiga komponen warna: R (*red*), G (*green*), dan B (*blue*). Pada citra hitam-putih (*grayscale*), nilai R = G = B untuk menyatakan bahwa citra hitam-putih (*grayscale*) hanya mempunyai satu nilai kanal warna setiap pikselnya, nilai tersebut digunakan

untuk menunjukkan tingkat intensitas. Warna yang dimiliki adalah warna dari hitam, keabuan dan putih. Tingkatan keabuan disini merupakan warna abu yang memiliki tingkatan dari warna hitam hingga mendekati warna putih. Nilai 0 mewakili keadaan gelap atau warna hitam, sedangkan nilai maksimum mewakili intensitas sangat terang atau warna putih. Citra *grayscale* umumnya adalah citra 8 *bit* (Munir, 2004). *CT Scan* merupakan salah satu satu contoh citra *grayscale* di bidang biomedis yang banyak memiliki informasi bagi dunia kedokteran untuk mendiagnosis dan mengevaluasi data kesehatan dari pasiennya. Pada penelitian kasus ini akan menggunakan citra *CT Scan* pendarahan otak yang akan diubah ke format berkas *bitmap (*BMP*).*

2.2.4. Citra *CT Scan*

Citra *CT Scan* diperoleh dari suatu alat *diagnostic* yaitu *CT* (*computer tomography*) *Scanner*. *CT (computer tomography) Scanner* adalah alat *diagnostic* dengan teknik radiografi yang menghasilkan gambar potongan tubuh secara melintang berdasarkan penyerapan sinar-x pada irisan tubuh yang ditampilkan pada layar monitor tv hitam putih. *Computer Tomography (CT)* biasa juga disebut *Computed axial tomography (CAT), computer-assisted tomography*, atau *(body section roentgenography)* yang merupakan suatu proses yang menggunakan digital processing untuk menghasilkan suatu gambaran internal tiga dimensi suatu obyek dari satu rangkaian sinar x yang menghasilkan gambar dua dimensi. Kata *"tomography"* diperoleh dari bahasa Yunani yaitu *tomos* (irisan) dan *graphia* (gambarkan) (Ramadhani, 2006).

CT Scanner dapat menghasilkan gambar-gambar yang sangat akurat dari objek-objek di dalam tubuh seperti tulang, organ, dan pembuluh darah. Gambar gambar ini sangat berguna dalam mendiagnosa berbagai penyakit, seperti kanker, penyakit jantung dan stroke. Citra yang dihasilkan *CT Scanner* jauh lebih detail dibanding citra yang diperoleh *x-ray* biasa. Prinsip dasar *CT scanner* mirip dengan perangkat radiografi yang sudah lebih umum dikenal. Kedua perangkat ini sama-sama memanfaatkan intensitas radiasi terusan setelah melewati suatu obyek untuk membentuk citra atau gambar. Perbedaan

antara keduanya adalah pada teknik yang digunakan untuk memperoleh citra dan pada citra yang dihasilkan. Tidak seperti citra yang dihasilkan dari teknik radiografi, informasi citra yang ditampilkan oleh *CT scan* tidak *overlap* (tumpang tindih) sehingga dapat memperoleh citra yang dapat diamati tidak hanya pada bidang tegak lurus berkas sinar (seperti pada foto *rontgen*), citra *CT scan* dapat menampilkan informasi tampang lintang obyek yang diinspeksi. Oleh karena itu, citra ini dapat memberikan sebaran kerapatan struktur internal obyek sehingga citra yang dihasilkan oleh *CT scan* lebih mudah dianalisis daripada citra yang dihasilkan oleh teknik radiografi konvensional.

CT Scan memiliki beberapa kelebihan dibanding *x-ray* biasa. Citra yang diperoleh *CT Scan* beresolusi lebih tinggi, sinar rontgen dalam *CT Scan* dapat difokuskan pada satu organ atau objek saja, dan citra perolehan *CT Scan* menunjukkan posisi suatu objek relatif terhadap objek-objek di sekitarnya sehingga dokter dapat mengetahui posisi objek itu secara tepat dan akurat. Kelebihan-kelebihan tersebut telah membuat *CT Scan* menjadi proses radiografis medis yang paling sering direkomendasikan oleh dokter dan, dalam banyak kasus, telah menggantikan proses *x-ray* biasa secara total.

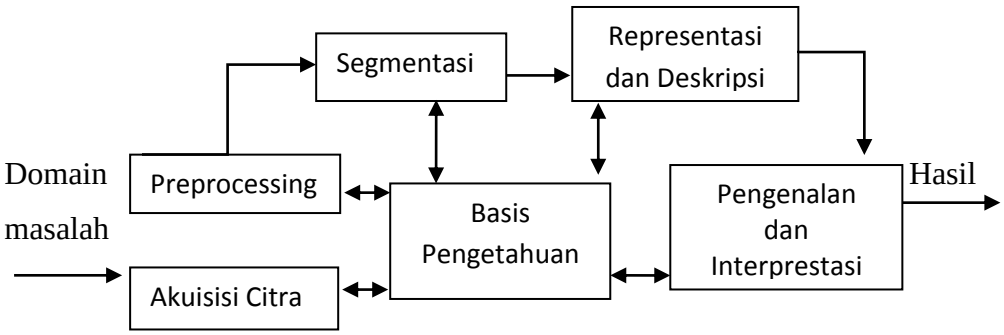
2.3. Pengolahan Citra

Menurut (Indrawati, 2013), Pengolahan citra adalah pemrosesan citra, khususnya dengan menggunakan komputer, menjadi citra yang kualitasnya lebih baik. Tujuan utama pengolahan citra adalah agar citra yang mengalami gangguan mudah diinterpretasi oleh manusia maupun mesin (komputer). Pengolahan citra atau *images prosessing* mencakup dua aspek proses pengubahan sebuah citra sebagai berikut:

1. Meningkatkan kualitas informasi dari sebuah citra (gambar) yang di gunakan untuk kepentingan interpretasi manusia.

2. Mengubah citra dari sebuah gambar yang di gunakan untuk mempermudah proses persepsi mesin *autonomous* agar lebih mudah dalam mengambil keputusan (Sutoyo,2009).

Selanjutnya Menurut (Sutoyo, 2009), secara umum, tahap-tahap pengolahan citra dapat di lihat pada gambar berikut:



Gambar 2.1. Tahapan Pengolahan Citra Digital

2.3.1. Akuisisi Citra

Tujuan akuisi citra adalah untuk menentukan data yang di perlukan dan memilih metode perekaman citra digital.

2.3.2. Prosessing

Tahapan ini di perlukan untuk menjamin kelancaran pada proses berikutnya yang di lakukan pada tingkatan ini adalah :

a. Peningkatan kualitas citra *(Kontras, Brightness)*
b. Menghilangkan noise
c. Perbaikan citra *(images restoration)*
d. Transpormasi *(images Transpormation)*
e. Menentukan bagian citra yang akan di observasi.

2.3.3. Representasi dan Deskripsi

Dalam hal ini representasi merupakan suatu proses untuk mempresentasikan suatu wilayah sebagai suatu daftar titik-titik koordinat dalam kurva yang tertutup, dengan deskripsi luasan atau perimeternya. Setelah suatu wilayah

dapat di representasi, proses selanjutnya adalah melakukan deskripsi citra dengan cara seleksi ciri dan exstraksi ciri *(feature extraction and selection).* Seleksi ciri bertujuan untuk memilih informasi kuantitatif dari ciri yang ada, yang dapat membedakan kelas-kelas objek secara baik,sedangkan exstraksi ciri bertujuan untuk mengukur kuantitatif ciri setiap pixel, misalnya rata-rata, standar deviasi, koofisien variasi, signaltonoise ration (SNR).

2.3.4. Pengenalan dan Interprestasi

Tahapan pengenalan bertujuan untuk memberikan label pada sebuah objek yang informasiknya di sediakan oleh deskriptor sedangkan tahan interprestasi bertujuan untuk memberikan arti atau makna kepada kelompok objek-objek yang di kenali.

2.3.5. Basis Pengetahuan

Basis pengetahuan sebagai basis data pengetahuan berguna untuk memandu oprasi dari masing-masing modul proses yang mengkontrol intraksi antara modul-modul tersebut. Selain itu, basis pengetahuan juga di gunakan sebagai reprensi pada proses *tamplate matching* atau pada pengenalan pola.

2.4. Operasi Morfologi Citra

Menurut (Prasetyo, 2011), Kata morfologi umumnya menyatakan cabang ilmu biologi yang mempelajari bentuk dan struktur tubuh hewan dan tumbuh-tumbuhan. Istilah yang sama juga di gunakan dalam pengolahan citra digital. Dalam konteks *mathematical morphology* oprasi morfologi di gunakan sebagai tool untuk pengextrakan komponen citra yang berguna dalam representasi dan deskripsi bentuk daerah, seperti *Boundaries, skeletons, dan convex hull.* Teknik morfologi juga di gunakan untuk pre atau post processing, seperti morfologi *filtering*, *thinning* dan *pruning*.

2.4.1. *Structuring Element*

Menurut *(*Anwarningsih, 2010), Pemrosesan citra secara morfologi di lakukan dengan menerapkan sebuah *Structuring element (strel)* terhadap sebuah citra.

Structuring element memegang peranan penting dalam pengolahan citra dengan operasi morfologi. *Structuring element* dapat diibaratkan dengan penyaring pada pemroses citra biasa (bukan secara morfologi). Dalam operasi morfologi, pemilihan *structuring element* sangan mempengaruhi hasil pemrosesan citra. Penggunaan dua buah *structuring element* yang berbeda akan menghasilkan hasil yang berbeda meski objek atau citra yang di analisa sama. Ada beberapa *stucturing element* yang bisa di gunakan, yaitu *rectangle, square, disk, liniar, dan diamont.* Setiap bentuk strel tersebuat memiliki kelebihan dan kekurangan masing-masing. *Structuring element* bentuk *rectangle dan square* dapat di gunakan untuk mendeteksi tepi sebuah objek. *Structuring element disk* dapat di gunakan oprasi dilasi atau rotasi. Sedangkan *Strcturing element* berbentuk *linier/line* hanya dapat mendeteksi *single border.*

2.4.2. Dilasi

Dilasi adalah oprasi morfologi yang akan menambahkan piksel pada batas antar objek dalam suatu citra digital. Secara rinci dilasi merupakan suatu proses menambahkan pixel pada batasan dari objek dalam suatu citra sehingga nantinya apabila di lakukan operasi ini maka citra hasilnya memiliki ukurannya lebih besar dibandingkan citra aslinya.

2.4.3. Erosi

Erosi merupakan kebalikan dari dilasi. Proses ini akan membuat ukuran sebuah citra menjadi lebih kecil. Berbeda dengan dilasi, apabila erosi di lakukan maka yang di kerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari piksel yang di tambah atu di hilangkan bergantung pada ukuran dan bentuk dari *Strcturing element* yang di gunakan.

2.4.4. *Opening*

Opening merupakan kombinasi proses di mana suatu citra digital di kenai oprasi erosi di lanjutkan dengan dilasi. Oprasi *opening* pada citra mempunyai

efek memperhalus batas-batas objek, memisahakan objek-objek yang sebelumnya bergandengan, dan menghilangkan objek-objek yang lebih kecil dari pada ukuran *strcturing element.*

2.4.5. Pengubahan Histogram Derajat Keabuan

Ada dua cara untuk melakukan pengubahan skala derajat keabuan, pertama dengan teknik penyamaan histogram berdasarkan citra aslinya *(histogram equalization* atau *histogram linearization)*, kedua dengan teknik pengubahan histogram yang sengaja dibuat *(direct histogram spesification).*

2.5. Segmentasi Citra

Menurut Munir (2006), segmentasi citra merupakan operasi yang bertujuan untuk memecahkan suatu citra kedalam beberapa segmen dengan suatu kriteria tertentu. Menurut Putra segmentasi citra merupakan teknik untuk membagi suatu citra menjadi beberapa daerah *(region)* dimana setiap daerah memiliki kemiripan atribut. Segmentasi meliputi beberapa teknik segmentasi yaitu pengambangan *(thresholding),* penandaan komponen terhubung, segmentasi berbasis *cluster,* dan transformasi *hough*. Teknik yang digunakan dalam kasus-kasus penelitian ini adalah teknik segmentasi berbasis *cluster*. Segmentasi ini menggunakan data multi dimensi untuk mengelompokkan piksel citra kedalam beberapa *cluster*. Pada umumnya piksel di *cluster* berdasarkan kedekatan jarak antar piksel dan iterasi. Ada banyak metode dalam cluster ini diantaranya *K-Means, C-Mean*s dan banyak lagi metode lainnya tetapi penulis menggunakan metode *expectation maximization*. Metode ini merupakan metode *cluster* yang berdasarkan iterasi melalui perhitungan probabilitasnya.

2.6. Segmentasi Dengan Metode *Expectation Maximization (*EM*)*

Metode algoritma *Expectation Maximization (*EM*)* ini merupakan metode untuk memperoleh pendugaan (ekspektasi) yang memberikan hasil yang baik dengan memaksimumkan fungsi kemungkinan. EM termasuk algoritma *clustering* yang berbasiskan model menggunakan perhitungan probabilitas. Metode iteratif

tersebut akan menghasilkan *Maximum Likelihood (*ML*),* yang menghasilkan parameter baru, yaitu bobot *mixture, mean*, dan *kovarian* atau standar *deviasi* (Handayani, 2012). EM terdiri dari dua tahap yaitu *Expectation* dan *Maximization*, secara umum dijelaskan seperti dibawah ini (Piater, 2002):

1. **E-step** untuk menghitung *expected values* (nilai dugaan) dari parameter berupa *mean*, standar *deviasi* serta probabilitas.
2. **M-step** untuk menghitung kembali parameter yang sama dengan memaksimalkan nilai *mean* (rata-rata), standar *deviasi* beserta probabilitas yang baru. Perbedaan yang digunakan untuk mengestimasi ulang parameter dilakukan secara berulang-ulang hingga mencapai local maksimum.

Kedua tahap tersebut dilakukan berulang-ulang sampai *hipotesa* dari *converge* (nilai yang terpusat) mencapai nilai yang *stationer*. Untuk model algoritma EM, setiap *cluster* memiliki *distribution probability* (kemungkinan penyebaran) yang sama dan untuk setiap kejadian data digunakan parameter nilai *estimate* (perkiraan) pada setiap *distribution* (penyebaran). Tahapan-tahapan dari algoritma EM diperlihatkan di bawah. Kita dapat menginisialisasikan pendugaan untuk *mean* dan standar *deviasi*, yang mana *mean* dan standar *deviasi* bertindak sebagai *cluster*, sampling *probability* sebagai *cluster* 1 (*P*t+1), dan probability itu sendiri sebagai cluster 2 (*Pt+1-P*). Algoritma EM mencari hipotesis *Maximum Likehood* berdasarkan iterasi. Secara singkat langkah umum algoritma *EM* menurut Mustapha (2009) dan dimodifikasi pada tahap awal berdasarkan Handayani (2008) seperti dibawah ini:

Step 1 : Menentukan *k cluster* (jumlah *cluster*).

Step 2 : Langkah inisialisasi, langkah insialisasi yaitu menginisialisasikan σ2 sebagai standar *deviasi*, *P* adalah probabilitas, μ adalah *mean* (rata-rata). Dimana nilai probabilitas berdasarkan nilai *mean*, *P0=(*μ01, μ02, . . , μ0k).

P0k = μ0k ..... (2.1)

Selain itu *mean* dan standar *deviasi* juga bertindak sebagai k *cluster* pada masing-masing *cluster*.

Dimana:
k = nomor urut dari *Gaussian*

Kemudian menghitung nilai yang telah diinisialkan berupa *mean* (rata-rata), standar *deviasi* dan probabilitas terhadap objek, berdasarkan rumus dibawah ini (Santoso, 2006):

Mean : = ΣΣ ..... (2.2)

Dimana:
μ = *mean (*rata-rata)
x = nilai data dari *pixel* gambar

Standar *Deviasi* : $\sigma = \sqrt{\frac{\sum f(x-\mu)^2}{n-1}}$ ..... (2.3)

Dimana :
σ = standar *deviasi*
σ2 = *variansi*
n = banyak data dari piksel gambar

Step 3 : Dilakukan tahap ekspektasi, yaitu berdasarkan nilai mean dan standard deviasi yang sudah didapat akan dihitung probability untuk setiap objek terhadap k *cluster* (*mean* dan standar *deviasi*) dengan menggunakan *probability density function (pdf)* dari *plot* GM. Dengan rumus probabilitas dari *Gaussian Mixture Model (GMM)* sebagai berikut:

$$P_t = f(x|\mu, \sigma^2) = \left(\frac{1}{(\sigma\sqrt{2\pi})e^{\frac{-(x-\mu)^2}{2\sigma^2}}} \right) \quad \text{.....(2.4)}$$

Dimana:

P = probabilitas

Step 4 : Dilakukan tahap *maximization*, yaitu berdasarkan nilai probability setiap objek pada step 3, akan di hitung kembali *mean*, standard *deviasi* dan probabilitas baru, dengan ketentuan A dan B sebagai *k cluster* yang mengandung nilai variabel *x1, x2, x3, . . .,xn* dimana *P*A = *P*B = 0.5, dengan rumus sebagai berikut:

[0.5*P* (*x1*|A) + 0.5 (*x1*|B)] [0.5*P* (*x2*|A) + 0.5 (x2|B)]...[0.5*P* (*xn*|A) + 0.5 (*xn*|B)] ……..... (2.5)

Dimana:

P= Probabilitas setiap objek pada k *cluster*

x1, x2, x3, xn = nilai *estimasi* (perkiraan) hingga *xn* iterasi.

Step 5 : Jika selisih antara pobabilitas lama dengan probabilitas baru yang didapat lebih besar dari nilai yang ditoleransi (1) maka dilakukan kembali step 3, namun jika tidak maka iterasi berhenti dan akan didapat hasil *cluster*.

$||P\,t{+}1 - Pt|| < 1$……..... (2.6)

Algoritma *EM-segmentation* secara garis besar dijelaskan oleh Kokkinos (2008), seperti dibawah ini:

```
System Pseudocode
[O1; : : : ;OK] = DETECT OBJECTS(I)
for i = 1 to K do
Ai = INITIALIZE(Oi)
repeat
SEGi = E STEP(AOi; B; I)
AOi = M STEP(SEGi; I)
until CONVERGENCE
VERIFY(Oi; SEGi;AOi)
end for
```

Gambar 2.2. *Algoritma System Pseudocode EM-Segmentation*

2.7. Metode *Gaussian Mixture Model* (GMM)

Gaussian Mixture Model (GMM) merupakan suatu paramater dari *probability density function (pdf)* atau fungsi kepadatan dari kemungkinan yang mewakili bobot dari jumlah kepadatan komponen *Gaussian* (bobot *means*, *weight*, *covariance*, *Likelihood* dan parameter lain yang terkait). GMM biasanya digunakan untuk suatu model parameter dari *probability distribution* (penyebaran kemungkinan) untuk ukuran yang berkesinambungan dan memiliki karakteristik pada suatu sistem *biometrik*. Sama halnya dengan bidang vokal yang berhubungan dengan fitur *spektral* pada sebuah sistem pengeras suara. Parameter dari GMM merupakan perkiraan dari data percobaan yang menggunakan algoritma iterasi EM *(Expectation-Maximization)* (Reynolds, 2004). Suatu *Gaussian Mixture Model* merupakan bobot jumlah dari kepadatan komponen *Gaussian* (parameter), seperti pada rumus umum *GMM* berikut ini (Zarpak dan Rahman, 2008):

$$G(x|\lambda) = \sum_{i=1}^{M} w_i \; p(x|\mu_i, \Sigma_i) \quad \ldots\ldots (2.7)$$

Keterangan : $\lambda(w_i, \mu_i, \Sigma_i)$ $\quad i = 1, \ldots, M,$

λ = parameter Gaussian

w_i = bobot *mixture/ weight*

Σ_i = matrik kovarian

Matrik *kovarian* berupa bilangan berpangkat dan batas dari diagonal (σ=standar *deviasi*).

Dimana: $\sum_{i=1}^{M} w_i = 1,$

Maka masing-masing komponen dari fungsi kepadatannya:

$$P = f(x|\mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} \exp \frac{-(x-\mu)^2}{2\sigma^2} \quad \ldots\ldots (2.8)$$

Kapasitas *GMM* mewakili *class* yang luas dari penyebaran *sample*. Salah satu sifat dari *GMM* yang sangat kuat yaitu kemampuan untuk memperkirakan atau menaksirkan bentuk secara khusus. *GMM* mewakili penyebaran fitur berdasarkan

posisi atau mean *vector (M)*, bentuk elip atau lonjong atau matrik *kovarian (V)*, kuantitas vektor atau model kedekatan ketetanggaan berdasarkan penyebaran dengan ciri khas dan karakteristik tertentu. Kegunaan *GMM* selain untuk mewakili penyebaran fitur pada sistem biometrik juga jadi motivasi bagi dugaan intuitif bahwa kepadatan komponen individual bisa memodelkan beberapa setelan dasar dari kelas yang disembunyikan *(hidden class).*

2.8. *Resizing* atau *Scaling* (Penyekalaan)

Menurut Hearn dan Baker (1986) *Resizing* atau *Scaling* merupakan proses pengolahan citra yang bertujuan untuk mengubah ukuran (dimensi) dari suatu citra. Perubahan citra yang terjadi akibat proses penyekalaan akan meyebabkan citra berubah menjadi citra baru yang ukurannya sesuai dengan skala perubahan yang ditetapkan, umumnya dikenal dengan sebutan faktor skala. Berikut ini adalah rumus *resizing* atau *scaling* menurut Hearn dan Baker (1986):

$$x'y'1 = [x\ y\ 1]\begin{bmatrix} Sx & 0 & 0 \\ 0 & Sy & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad \text{.....(2.9)}$$

Rumus yang dipakai:

$$x' = Sx\ x$$
$$y' = Sy\ y$$

Nilai Skala: >1, memperbesar citra asli

 <1, memperkecil citra asli

Keterangan:

x’ : koordinat x setelah penyekalaan

y’ : koordinat y setelah penyekalaan

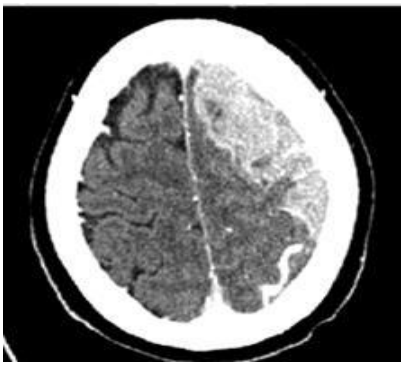
x : koordinat x sebelum penyekalaan

y : koordinat y sebelum penyekalaan

Sx : faktor skala sumbu x (horizontal)

Sy : faktor skala sumbu y (vertikal)

Contoh operasi penyekalaan dengan nilai (Sx = 0.5, Sy = 2), hasil adalah sebagai berikut:



Gambar 2.3. Citra Asli

Gambar 2.4. Citra Penyekalan

Perubahan hasil penyekalaan:

1. Perubahan lebar citra, terjadi ketika citra hanya dikalikan oleh faktor skala dengan arah sumbu x saja.
2. Perubahan tinggi citra, terjadi ketika citra hanya dikalikan oleh faktor skala dengan arah sumbu y saja.
3. Perubahan lebar dan tinggi citra, terjadi ketika citra dikalikan oleh faktor skala, baik kearah sumbu x maupun kearah sumbu y. Citra akan terskala dengan sempurna (perubahan tinggi dan lebar yang terjadi sebanding dengan tinggi dan lebar dari citra asalnya), jika faktor skala arah sumbu x bernilai sama dengan faktor skala arah sumbu y.

2.9. UML (*Unified Markup Languange*)

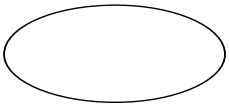
Menurut (Shalahuddin, 2014), menyatakan UML adalah bahasa spesifikasi standar yang dipergunakan untuk mendokumentasikan, menspesifikasikan dan membangun perangkat lunak. UML merupakan metodelogi dalam mengembangkan sistem berorientasi object dan juga merupakan alat untuk mendukung pengembangan sistem. UML saat ini banyak dipergunakan dalam dunia industri yang merupakan standar bahasa pemodelan umum pada dunia industri perangkat lunak dan pengembangan software. Alat bantu yang

dipergunakan dalam perancangan berorientasi objek berbasis UML adalah sebagai berikut :

2.9.1. *Use Case Diagram*

Menurut (Stiyanto, *2013), use case diagram* merupakan pemodelan untuk kelakuan (*behavior*) sistem yang dibuat. *Usecase* mendeskripsikan sebuah interaksi antara satu atau lebih actor dengan sistem yang akan dibuat. Dapat dikatakan use case digunakan untuk mengetahui fungsi apa saja yang dibuat dan siapa saja yang berhak mengunakan fungsi-fungsi tersebut. Berikut simbol yang digunakan pada *use case diagram*:

Tabel 2.1. Simbol *Use Case Diagram*

Gambar	Keterangan
	*Use Case* menggambarkan fungsionalitas yang disediakan *system* sebagai unit-unit yang bertukar pesan antar unit dengan *actor*.
	Aktor adalah orang lain atau *system* yang mengaktifkan fungsi dari target *system*.
	Ososiasi antara *actor* dan *use case* yang menggunakan panah terbuka untuk mengindikasikan bila actor berinteraksi secara pasif dengan system
	Ososiasi antara *actor* dan *use case* yang mengunakan garis untuk mengindikasikan siapa atau apa yang meminta interaksi langsung.
	Include merupakan pemangilan use case oleh *use case* lainya, contohnya pemanggilan sebuah fungsi program.

<<extend>>	Perluasan dari *use case* lain jika kondisi atau syarat terpenuhi.

2.9.2. *Activity Diagram*

Activity diagram dapat juga digunakan untuk memodelkan *action* yang akan dilakukan saat sebuah operasi dieksekusi, dan memodelkan hasil dari action tersebut**.** Simbol- simbol yang digunakan dalam *activity diagram* yaitu:

Tabel 2.2 Simbol *Activity Diagram*

Gambar	Keterangan
Activity	Mengambarkan suatu proses/kegiatan sistem
State	Menunjukan eksekusi dari suatu aksi
	Start Point merupakan awak dari suatu aktivitas
	End point, akhir aktivitas
	Fork/Rake Node Satu aliran atau beberapa aliran yang pada tahap tertentu berubah menjadi satu atau beberapa aliran.
swimlane	Swinlane, pembagian activity diagram untuk menunjukan siapa melakukan apa

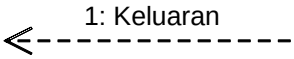
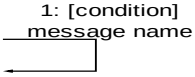
2.9.3. *Class Diagram*

Class diagram merupakan hubungan antara kelas dan penjelasan detail tiap-tiap kelas di dalam model desain dari suatu sitem, juga memperlihatkan aturan-aturan dan tanggung jawab entitas yang menentukan perilaku sistem.

2.9.4. *Sequence Diagram*

Sequence diagram menggambarkan kelakuan objek pada *use case* dengan mendeksripsikan waktu hidup objek dan pesan yang dikirimkan dan diterima antar objek. Simbol-simbol yang digunakan dalam *sequence diagram*, yaitu :

Tabel 2.3. Simbol *Sequence Diagram*

Gambar	Keterangan
	Orang, proses atau system lain yang berinteraksi dengan sistem
	mengirimkan data informasi ke objek lainnya, arah panah mengarah pada objek yang dikirimi.
	*Activation*, *activation* mewakili sebuah eksekusi operasi dari objek.
	*Message return*, simbol membalas pesan
	*Recursive*, mengambarkan pengiriman pesan untuk dirinya sendiri.
<<create>>	Menyatakan suatu objek membuat objek yang lain, arah panah mengarah objek yang di buat.
Garis Hidup / Lifeline	Menyatakan kehidupan suatu objek

2.10. Matrix Laboratori (MATLAB)

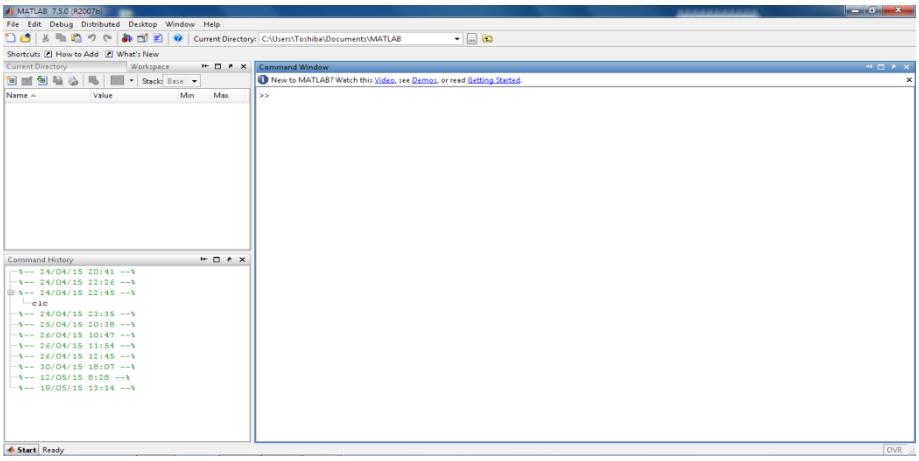
MATLAB adalah sebuah lingkungan komputasi numerical dan bahasa pemrograman komputer generasi keempat. Dikembangkan oleh The Math Works, MATLAB memungkinkan manipulasi matriks, pemplotan fungsi dan data, implementasi algoritma, pembuatan antar muka pengguna, dan pengantar muka dengan program dalam bahasa lainnya. Tipe data dasar MATLAB adalah matriks

(array). MATLAB tidak membutuhkan dimensi, oleh karena itu pengguna memori dapat dihemat. Pada MATLAB semua data di anggap sebagai matrik. Untuk satu bilangan akan dianggap sebagai matrik 1x1. untuk pengembangan algoritma,

MATLAB menyediakan antar muka *command line*, sebuah inter preter untuk menangani bahasa pemrograman MATLAB, fungsi manipulasi string dan bilangan, 2D dan 3D *plotting function* dan mampu membuat tampilan GUI (*Grapical User Interface*). Pemerograman MATLAB menginterprestasikan perintah yang mempersingkat waktu pemrograman. Lingkungan kerja MATLAB terdiri dari 3 bagian. Bagian bagian tersebut adalah Matlab Dekstop, MATLAB Editor dan *help sistem*. Untuk lebih jelasnya dapat di lihat pada penjelasan sebagai berikut:

a. MATLAB Desktop

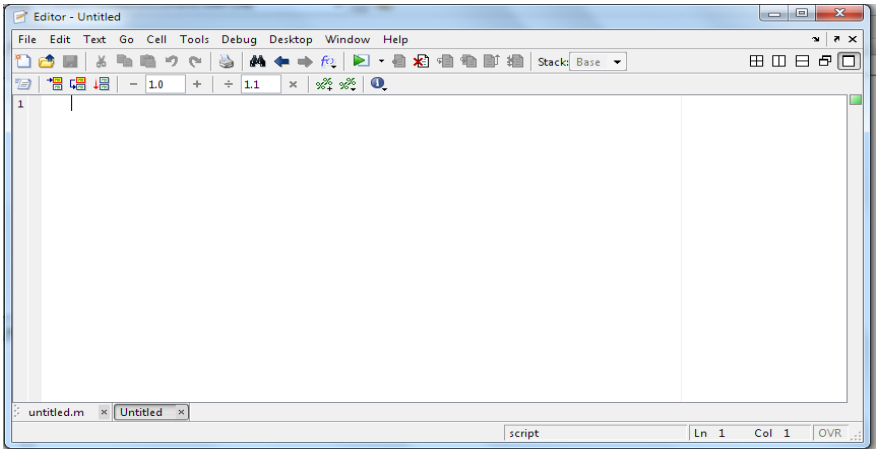
Menurut (Murvin, 2007.), Pada umumnya MATLAB Dekstop terdiri dari lima *sub window*, diantaranya *command window*, *worspace browser*, *current distory window*, *command history window*, dan *figure window* yang di gunakan untuk menampilkan grafik.



Gambar 2.5. Jendela MATLAB Desktop

b. MATLAB Editor

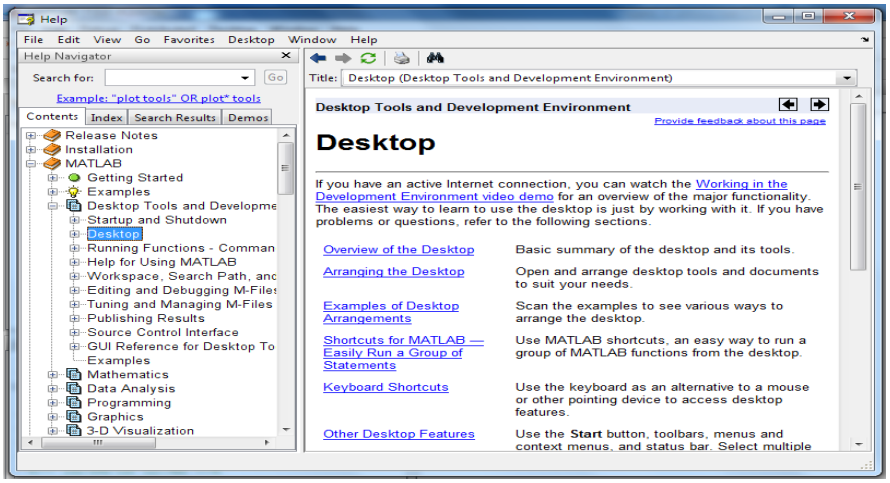
MATLAB Editor digunakan untuk membuat dan mengedit M-file. MATLAB Editor juga di gunakan untuk menyimpan, melihat file dan melakukan *fload* grafik.



Gambar 2.6. MATLAB Editor

c. *Help System*

Help System mempunyai *browser* khusus yang menangani bantuan. Dokumen *help* di presentasikan dalam sebuah dokumen *HTML* yang terorganisir.



Gambar 2.7. MATLAB *Help System*