

## BAB III METODE PENELITIAN

### 3.1. Alat dan Bahan Penelitian

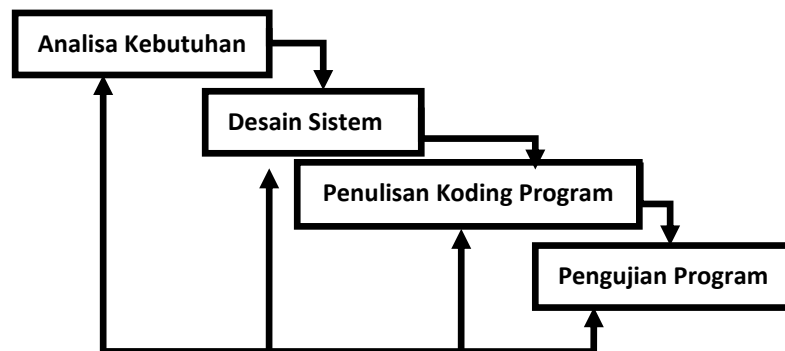
Pada penelitian ini digunakan citra pendarahan otak dengan format BMP yang diperoleh dari hasil *resize* foto dari citra *CT Scan* pendarahan otak. Perangkat keras yang digunakan yaitu *Processor Intel (R) Core (TM) i3-3220 CPU @3.00 Ghz*, *hard disk 500 GB* dan *RAM 4 GB*. Sedangkan perangkat lunak yang digunakan yaitu sistem operasi *Windows 7*, *MATLAB 7.5.0 (R2007b)*, dan *Paint*.

### 3.2. Metode Pengumpulan Data

Pada tahap observasi ini dilakukan pengumpulan data di RSUD Abdul Moeloek. Dan pengumpulan citra *CT Scan* pendarahan otak dari data rumah sakit, yang terkumpul ini akan di gunakan dalam pembuatan serta pengujian program.

### 3.3. Metode Pengembangan Perangkat Lunak

Menurut (Prasetyo, 2011), metode yang digunakan dalam perancangan perangkat aplikasi dalam penelitian ini adalah metode *waterfall*. Seperti yang dijabarkan oleh metode ini memiliki tahapan yaitu sebagai berikut:



Gambar 3.1. Model *Waterfall*

### 3.3.1. Analisis Kebutuhan

Untuk memahami sifat program yang di bangun, rekayasa perangkat lunak harus memahami domain informasi, tingkah laku untuk kerja, dan antar muka (*interface*) yang di perlukan. Kebutuhan untuk baik *system* maupun perangkat lunak di dokumentasikan agar dapat di lihat lagi oleh user.

### 3.3.2. Desain System

Desain perangkat lunak adalah proses pembuatan antar muka (*interface*) perangkat lunak sebelum melakukan koding. Sebagai mana persyaratan, desain dokumentasi dan menjadi bagian dari konfigurasi perangkat lunak.

### 3.3.3. Penulisan Koding Program

Yaitu sebuah penulisan koding ke dalam sebuah program yang akan di buat sesuai dengan apa yang di butuhkan.

### 3.3.4. Pengujian Program

Pengujian program di lakukan dengan menginputkan *file* citra pendarahan pada otak kedalam program yang telah di buat. Ini di lakukan terhadap program pre-processing dan program EM-GMM. Pengujian di lakukan berdasarkan kualitas citra yang di hasilkan dan di uji berdasarkan waktu yang di perlukan dalam proses tersebut.

## 3.4. *Expectation Maximization Gaussian Mixture Model (EM-GMM)*

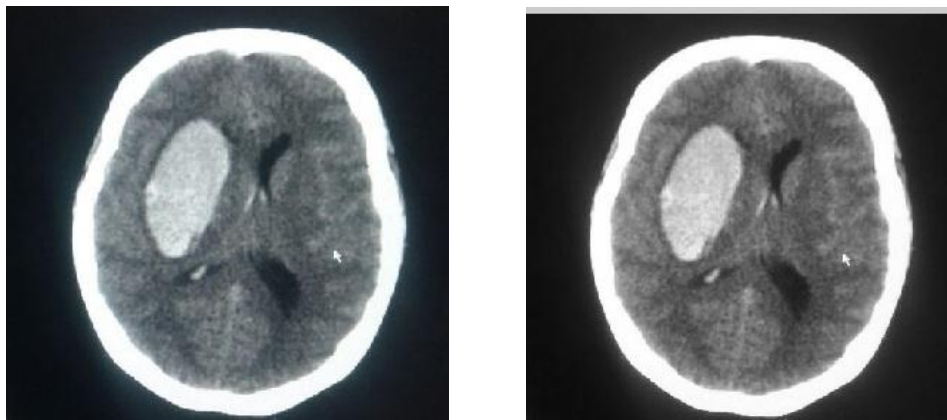
### 3.4.1. Citra Input

Gambar input berupa citra pendarahan pada otak yang di ambil di rumah sakit, citra ini berupa citra RGB dengan format BMP.

### 3.4.2. Mengubah Citra RGB menjadi *Grayscale*

Proses ini termasuk pada tahapan *prosessing*. Citra RGB di ubah warnanya menjadi abu-abu. Hal ini di maksudkan untuk mempermudah perhitungan nilai pixel. Pada citra RGB setiap pixel mempunyai tiga nilai, masing-masing untuk nilai warna merah (*red*), hijau (*green*), dan biru (*blue*). Sedangkan

pada citra keabuan, tiap piksel hanya memiliki satu nilai yang mewakili skala keabuannya. Dapat dilihat pada gambar berikut:



A. Citra Asli

B. Citra Yang Sudah Di *Grayscale*

**Gambar 3.2.** A. Citra Asli B. Citra Yang Sudah Di *Grayscale*

### 3.4.3. Meningkatkan Kecerahan dan Kontras Citra

Setelah citra diubah menjadi *grayscale* maka perlu dilakukan suatu peningkatan kontras citra. Hal ini dimaksudkan agar citra menjadi lebih baik sehingga mudah untuk dilakukan deteksi tepi. Operasi peningkatan kecerahan dan peregangan kontras dapat dilakukan sekaligus untuk kepentingan memperbaiki citra. Secara umum, gabungan kedua operasi tersebut dapat ditulis menjadi:

$$g(y,x) = f(y,x) + \beta$$

Berdasarkan rumus diatas, kontras akan naik jika  $\alpha > 1$  dan kontras akan turun jika  $\alpha < 1$ . Sedangkan nilai  $\beta$  jika bernilai negatif maka kecerahan akan naik namun jika  $\beta$  bernilai positif kecerahan akan turun.

Perhitungan :

$$g(y,x) = \alpha f(y,x) + \beta$$

$$g(y,x) = 2.5 \begin{bmatrix} 186 & 190 & 196 & 200 \\ 182 & 184 & 188 & 196 \\ 178 & 178 & 184 & 188 \\ 176 & 171 & 175 & 179 \end{bmatrix} + (-45)$$

$$g(y,x) = \begin{bmatrix} 420 & 430 & 445 & 455 \\ 410 & 415 & 425 & 445 \\ 400 & 400 & 415 & 425 \\ 395 & 382.5 & 392.5 & 402.5 \end{bmatrix}$$

#### 3.4.4. Mengubah Citra Grayscale Menjadi Citra EM-GMM.

Mengubah citra *grayscale* menjadi citra *biner* di maksudkan untuk memisahkan objek yang di amati dengan backgroundnya dengan perintah sebagai berikut:

```
t1=graythresh(g1); %(Untuk mempertajam citra biner)
a1=im2bw(g1,t1); %(Untuk menghasilkan perhitungan grayscale dan citra
biner)
l1 = +a1
m1 = [0 0 -1 0 0; 0 -1 -2 -1 0; -1 -2 16 -2 -1; 0 -1 -2 -1 0; 0 0 -1 0 0] % (rumus
EM-GMM)

e1=conv2(l1,m1); %(Untuk hasil perhitungan menggunakan EM-GMM)
s1=imfill(e1,'holes'); %(Untuk memperjelas citra biner(memfilter))
n1=imclearborder(s1,18); %(Menghilangkan border yang menempel pada
citra
seD = strel ('diamond',1);
d1 = imerode (n1, seD);
```

Selanjutnya pengambilan sebuah citra yang akan dicari besarnya nilai citra dengan menggunakan perintah `imshow (BW)`. Untuk citra *CT Scan* otak dengan pendarahan otak dibawah 30 CC mempunyai angka piksel 1681. Dapat di lihat pada gambar di bawah ini:



**Gambar 3.3.** Proses Pengambilan EM-GMM

#### 3.4.5. Melakukan Segmentasi Citra

Segmentasi yang paling sederhana dilakukan dengan metode ambang intensitas (*thresholding*). Nilai yang lebih kecil daripada nilai ambang diberlakukan sebagai area pertama dan yang lebih besar daripada atau sama dengan nilai ambang dikelompokkan sebagai area yang kedua. Dalam hal ini, salah satu area tersebut berkedudukan sebagai latar belakang. Cara seperti itulah yang disebut dengan metode pengambangan dwi-aras (*bi-level thresholding*) atau terkadang dinamakan pengambangan intensitas. Secara matematis dirumuskan sebagai berikut :

$$b(x,y)=f(x)= \begin{cases} 1, & \text{untuk } (y, x) \geq T \\ 0, & \text{untuk } (y, x) < T \end{cases}$$

T adalah ambang intensitas.



**Gambar 3.4.** Citra Hasil Segmentasi

Contoh perhitungan:

Citra *grayscale* 256 warna dengan ukuran 4 x 4 piksel, yang telah dideteksi.

Akan dilakukan operasi *threshold* dengan fungsi berikut :

$$f_0(x,y) = \begin{cases} 0, & \text{untuk } (x,y) < 184 \\ 1, & \text{untuk } (x,y) \geq 184 \end{cases}$$

186	190	196	200
182	184	188	196
178	178	184	188
176	171	175	179

Maka

1	1	1	1
0	1	1	1
0	0	1	1
0	0	0	0

perhitungan digital yang dilakukan adalah sebagai berikut : untuk setiap nilai intensitas citra asli yang nilainya  $< 184$  diubah menjadi 0 sedangkan setiap nilai intensitas citra asli yang nilainya  $\geq 184$  diubah menjadi 1.

### 3.5. Proses Penentuan Hasil Pendarahan Otak Pada Citra Hasil EM-GMM

Setelah objek disegmentasi, objek dihitung jumlah piksel putihnya dan kemudian dibandingkan jumlah piksel putihnya dengan piksel citra acuan. Citra acuan adalah citra yang digunakan untuk membedakan pendarahan otak apakah pendarahannya diatas 30 CC atau dibawah 30 CC.

Untuk menghitung luas area dapat dirumuskan sebagai berikut :

$$A = \text{jumlah piksel di baris ke-1} + \text{baris ke-2} + \dots + \text{baris ke-m}$$

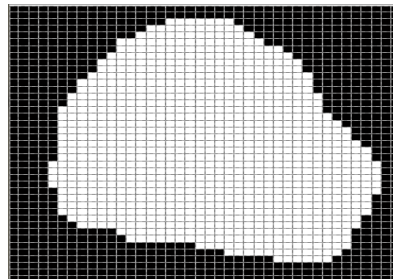
Dimana m adalah banyaknya baris dalam suatu citra.

Rumus lain menurut pendapat Sutoyo (2009) perhitungan luas objek dapat dirumuskan sebagai berikut :

$$A = \sum_{x=1}^m \sum_{y=1}^n O(x,y)$$

Mengisyaratkan bahwa nilai A diperoleh dengan cara membaca piksel-piksel milik objek. Bila ditemukan piksel milik objek nilai A bertambah 1. Bila bukan piksel milik objek maka nilai A tidak berubah.

Perhitungan :

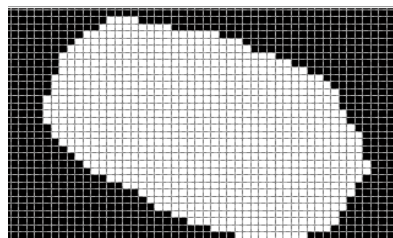


**Gambar 3.5.** Citra Acuan Yang Telah Disegmentasi

$A = \text{jumlah piksel di baris ke-1} + \text{baris ke-2} + \dots + \text{baris ke-27}$

$A = 7+10+12+15+17+19+19+21+20+\dots+ 26$

$= 1681 \text{ piksel}$



**Gambar 3.6.** Citra Yang Akan Diolah

$A = \text{jumlah piksel di baris ke-1} + \text{baris ke-2} + \dots + \text{baris ke-32}$

$A = 4+9+17+21+23+25+29+31+33+\dots+ 9 \Rightarrow A = 2681 \text{ piksel}$

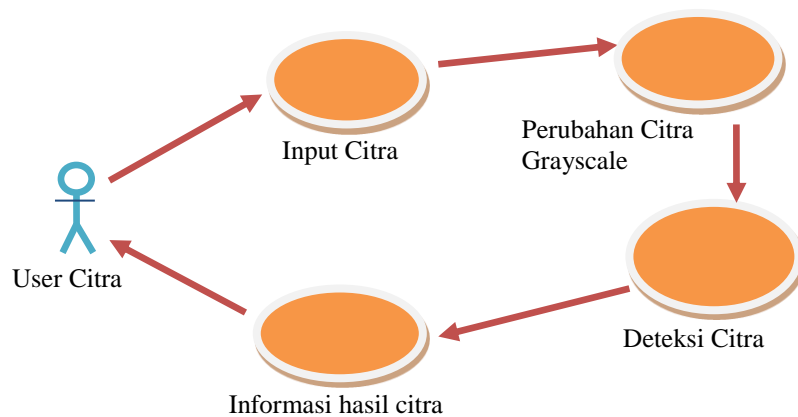
Dari citra yang telah di proses, dapat diambil kesimpulan mengenai keadaan pendarahan pada otak.

### 3.6. Perancangan

Tahap perancangan pada aplikasi yang akan dibangun adalah rancangan *use case diagram*, *activity diagram*, *sequence diagram*, dan rancangan antarmuka.

#### 3.6.1. Use Case Diagram

Gambar 3.7. dibawah ini merupakan *use case diagram* dari aplikasi untuk mengidentifikasi pendarahan otak pada citra *CT Scan* Pendarahan Otak.



**Gambar 3.7.** Use Case Diagram Aplikasi Untuk Mengidentifikasi Pendarahan Otak

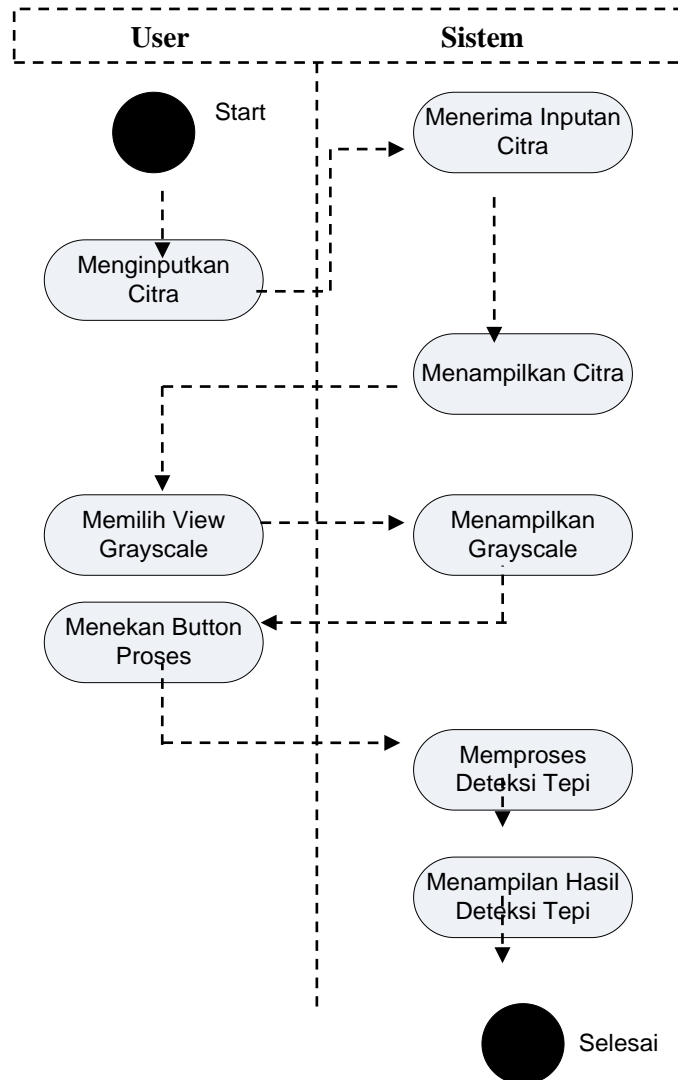
Penjelasan Gambar 3.7. *use case* adalah sebagai berikut:

Aktor dari aplikasi untuk mengidentifikasi pendarahan otak. Tugas aktor sistem ini adalah menginputkan citra kedalam sistem kemudian sistem akan memproses citra inputan. Setelah proses selesai, aktor sistem ini akan mendapatkan informasi mengenai citra yang dihasilkan yang telah di proses.



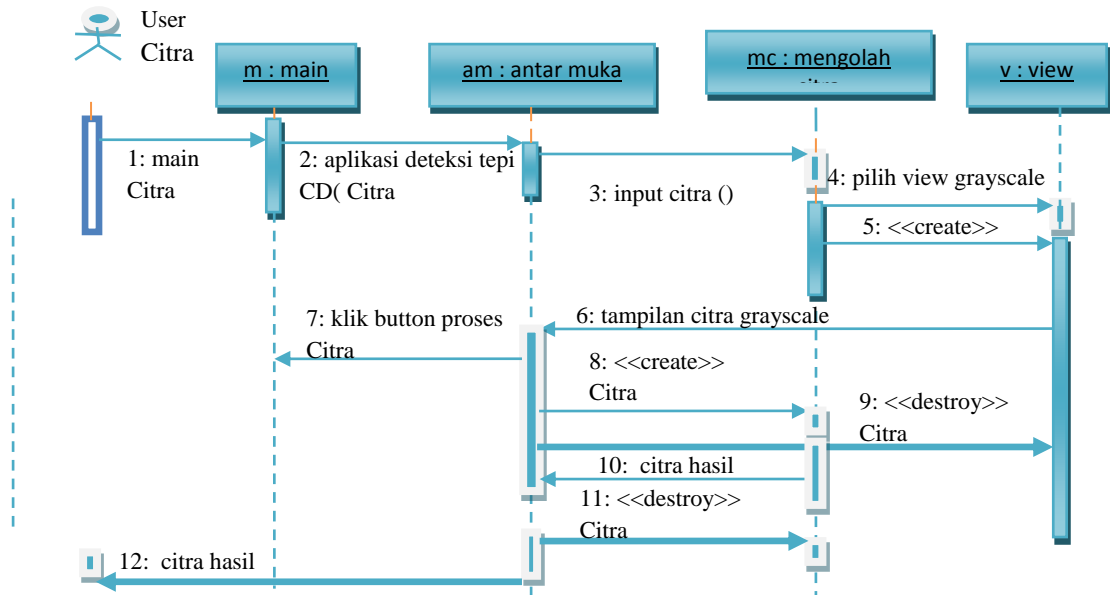
### 3.6.2. Activity Diagram

Gambar 3.3 di bawah ini merupakan *activity diagram* dari aplikasi untuk mengidentifikasi pendarahan otak pada citra *CT Scan* pendarahan otak.



**Gambar 3.8.** *Activity Diagram* EM-GMM Untuk Mengidentifikasi Pendarahan Otak

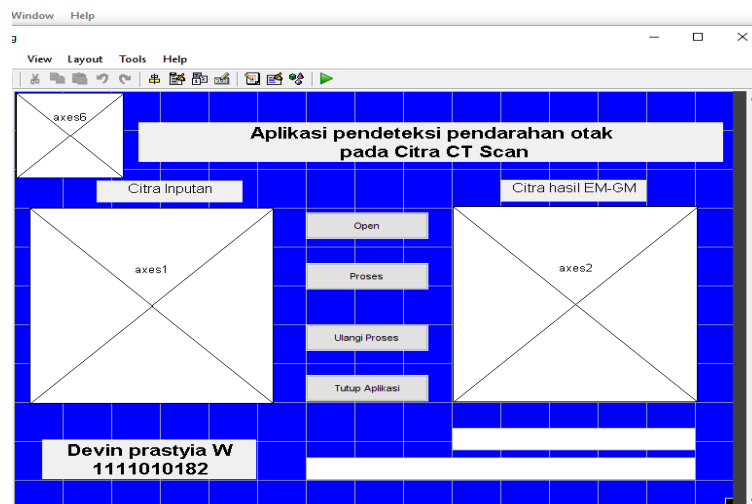
### 3.6.3. Sequence Diagram



Gambar 3.9. Sequence Diagram EM-GMM

### 3.7. Rancangan Tampilan Antar Muka

Gambar berikut ini adalah merupakan rancangan tampilan antar muka program untuk mengidentifikasi pendarahan otak pada citra *CT Scan* pendarahan otak menggunakan metode EM-GMM. Bisa dilihat pada gambar 3.10. berikut :



Gambar 3.10. Rancangan Antar Muka Aplikasi