



TUGAS AKHIR - KS184822

**SEGMENTASI CITRA MRI TUMOR OTAK  
MENGUNAKAN *GAUSSIAN MIXTURE MODEL*  
DENGAN PENDEKATAN ALGORITMA *REVERSIBLE  
JUMP MARKOV CHAIN MONTE CARLO***

**YUSUF PUJI HERMANTO  
NRP 062115 4000 0095**

**Dosen Pembimbing  
Prof. Drs. Nur Iriawan, M.Kom., Ph.D.**

**PROGRAM STUDI SARJANA  
DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2019**





**TUGAS AKHIR - KS184822**

**SEGMENTASI CITRA MRI TUMOR OTAK  
MENGUNAKAN *GAUSSIAN MIXTURE MODEL*  
DENGAN PENDEKATAN ALGORITMA *REVERSIBLE  
JUMP MARKOV CHAIN MONTE CARLO***

**YUSUF PUJI HERMANTO  
NRP 062115 4000 0095**

**Dosen Pembimbing  
Prof. Drs. Nur Iriawan, M.Kom., Ph.D.**

**PROGRAM STUDI SARJANA  
DEPARTEMEN STATISTIKA  
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2019**

*“Halaman ini sengaja dikosongkan”*



**FINAL PROJECT - KS184822**

**MRI-BASED BRAIN TUMOR SEGMENTATION USING  
GAUSSIAN MIXTURE MODEL WITH REVERSIBLE  
JUMP MARCOV CHAIN MONTE CARLO ALGORITHM**

**YUSUF PUJI HERMANTO  
SN 062115 4000 0095**

**Supervisor  
Prof. Drs. Nur Iriawan, M.Kom., Ph.D.**

**UNDERGRADUATE PROGRAMME  
DEPARTMENT OF STATISTICS  
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2019**

*“Halaman ini sengaja dikosongkan”*

## LEMBAR PENGESAHAN

# SEGMENTASI CITRA MRI TUMOR OTAK MENGUNAKAN *GAUSSIAN MIXTURE MODEL* DENGAN PENDEKATAN ALGORITMA *REVERSIBLE* *JUMP MARKOV CHAIN MONTE CARLO*

### TUGAS AKHIR

Diajukan untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Statistika  
pada  
Program Studi Sarjana Departemen Statistika  
Fakultas Matematika, Komputasi, dan Sains Data  
Institut Teknologi Sepuluh Nopember

Oleh :

**Yusuf Puji Hermanto**  
NRP. 062115 4000 0095

Disetujui oleh Pembimbing:

**Prof. Drs. Nur Iriawan, M.Kom., Ph.D.**

NIP. 19621015 198803 1 002



Mengetahui,  
Kepala Departemen Statistika



**Dr. Suhartono**

NIP. 19710929 199512 1 001

SURABAYA, JULI 2019





# SEGMENTASI CITRA MRI TUMOR OTAK MENGGUNAKAN GAUSSIAN MIXTURE MODEL DENGAN PENDEKATAN ALGORITMA REVERSIBLE JUMP MARKOV CHAIN MONTE CARLO

**Nama** : Yusuf Puji Hermanto  
**NRP** : 062115 4000 0095  
**Departemen** : Statistika  
**Dosen Pembimbing** : Prof. Drs. Nur Iriawan, MIKom., Ph.D.

## Abstrak

*Tumor otak merupakan penyakit mematikan yang menduduki peringkat ke-15 di Indonesia menurut Badan Kesehatan Dunia (WHO) pada tahun 2018. Dalam dunia medis, tumor otak dapat dideteksi melalui Magnetic Resonance Imaging (MRI). Permasalahan utama adalah bagaimana memisahkan area tumor otak sebagai Wilayah yang diminati (ROI) dengan bagian sehat lainnya (Non-ROI) di MRI. Dalam statistika komputasi, metode yang digunakan dalam segmentasi gambar adalah analisis cluster. Clustering Berbasis Model dengan Gaussian Mixture Model (GMM) sering digunakan untuk menemukan kluster tempat tumor tumbuh. Algoritma EM dan Bayesian digabungkan dengan Markov Chain Monte Carlo (MCMC) dapat digunakan untuk mengoptimalkan GMM. Namun, baik EM dan Bayesian MCMC diasumsikan bahwa jumlah cluster adalah tetap. Oleh karena itu, untuk memilih jumlah cluster yang optimal, kita harus menggunakan kriteria pemilihan cluster tertentu. Proses ini membuat segmentasi menjadi cukup rumit dan tidak otomatis. Penelitian ini mencoba menggunakan GMM menggunakan Algoritma Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) untuk mensegmentasi tumor otak dan membandingkannya dengan GMM-EM. Penggunaan RJMCMC diharapkan dapat mempercepat proses perhitungan. Selain itu, segmentasi citra MRI dapat menjadi lebih adaptif. Hasilnya menunjukkan bahwa dari Correct Classification Ratio (CCR), metode GMM-RJMCMC dapat memberikan hasil segmentasi yang lebih baik dibandingkan dengan GMM-EM.*

**Kata Kunci:** *Bayesian, Gaussian, Mixture Model, RJMCMC, Segmentasi MRI, Tumor Otak*

*“Halaman ini sengaja dikosongkan”*

# **MRI-BASED BRAIN TUMOR SEGMENTATION USING GAUSSIAN MIXTURE MODEL WITH APPROACH REVERSIBLE JUMP MARKOV CHAIN MONTE CARLO ALGORITHM**

**Name : Yusuf Puji Hermanto**  
**Student Number : 062114 4000 0095**  
**Department : Statistics**  
**Supervisor : Prof. Drs. Nur Iriawan, MIKom., Ph.D.**

## **Abstract**

*A brain tumor is the 15<sup>th</sup> deadly disease in Indonesia according to the WHO in 2018. In medical treatment, brain tumors can be detected through Magnetic Resonance Imaging (MRI). The main problem is how to separate the brain tumor area as the Region of interest (ROI) with the other healthy part (Non-ROI) in the MRI. In the computational statistics, a method used in image segmentation is cluster analysis. Model Based Clustering with Gaussian Mixture Model (GMM) is often used to find the cluster where the tumor is placed. The EM Algorithm and Bayesian coupled with Markov chain Monte Carlo (MCMC) could be used to optimize the GMM. However, both EM and Bayesian MCMC are assumed that the number of clusters is fixed. Therefore, to select the optimum number of the clusters, we have to use the certain cluster selection criteria. This process makes the segmentation quite complicated and is not automatic. This study tries to employ the GMM using Reversible Jump Markov Chain Monte Carlo Algorithm (GMM-RJMCMC) to segment the MRI-based brain tumor and compare it with the GMM-EM. The use of RJMCMC is expected to accelerate the calculation process. Moreover, the MRI image segmentation could become more adaptive. The result shows that from the Correct Classification Ratio (CCR), the GMM-RJMCMC could provide better segmentation results compared to the GMM-EM.*

**Keywords: Bayesian, Brain Tumor, Gaussian, Mixture Model, MRI Segmentation, RJMCMC**

*“Halaman ini sengaja dikosongkan”*

## KATA PENGANTAR

Puji syukur kehadiran Allah SWT atas limpahan rahmat dan hidayahNya, penulis dapat menyelesaikan Tugas Akhir yang berjudul  
**“SEGMENTASI CITRA MRI TUMOR OTAK  
MENGUNAKAN *GAUSSIAN MIXTURE MODEL* DENGAN  
PENDEKATAN ALGORITMA *REVERSIBLE JUMP  
MARKOV CHAIN MONTE CARLO*”**

dengan baik. Penulis menyadari bahwa penyusunan Tugas Akhir ini tidak terlepas dari bantuan dan dukungan dari berbagai pihak. Oleh karena itu, penulis mengucapkan terima kasih kepada:

1. Aba, Ibu, kakak, dan seluruh keluarga dirumah atas segala doa, kasih sayang, nasihat, dan dukungan yang diberikan kepada penulis.
2. Bapak Dr. Suhartono selaku Kepala Departemen Statistika yang telah memberikan banyak fasilitas untuk kelancaran penyelesaian Tugas Akhir ini.
3. Bapak Prof. Drs. Nur Iriawan, MIKom., Ph.D. selaku dosen pembimbing yang telah memberikan nasihat, kritik, saran dan waktu yang diberikan kepada penulis dalam menyelesaikan Tugas Akhir ini.
4. Ibu Dr. Dra. Kartika Fithriarsari, M.Si. dan Ibu Dra. Wiwiek Setya Winahju, M.S. selaku dosen penguji atas saran dan kritiknya yang sangat membangun.
5. Ibu Ni Luh Putu Satyaning P.Paramitha, S.Si., M.Sc. selaku dosen wali atas nasihat dan saran yang telah diberikan.
6. Ibu Anindya Apriliyanti Pravitasari, S.Si., M.Si., atas bantuan, nasihat, kritik, saran, serta waktu yang diberikan kepada penulis selama penyelesaian Tugas Akhir ini.
7. dr. Widiana Ferriastuti, Sp. Rad (K) atas bantuan, saran, dan waktu yang diberikan kepada penulis dalam pemberian rekomendasi dan validasi dari sudut pandang medis untuk Tugas Akhir ini.
8. Teman-teman terdekat M Bagus Rahmatullah, Arrafi Dwiargatra, Angga Mahendra, Barep Adji Widhi P, Nurun Nahdliyah, dan Fachrunisah yang senantiasa memberikan waktu dan dukungan yang diberikan dalam menyelesaikan Tugas Akhir ini.

9. Teman-teman seperjuangan Tugas Akhir Segmentasi Citra, Mbak Indah Nurmalasari, Taufik Azmi, dan Nur Indah Asrori yang selalu memberikan waktunya untuk diskusi selama penyelesaian Tugas Akhir ini.
10. Teman-teman seperjuangan PW 120 serta Vivacious 2015  $\Sigma 25$ , yang telah memberikan rasa kekeluargaan selama 4 tahun penulis menempuh studi di ITS.
11. Semua pihak yang terlibat dan mendukung dalam penyelesaian Tugas Akhir ini yang tidak dapat penulis sebutkan satu-persatu.

Besar harapan penulis untuk mendapatkan kritik dan saran yang membangun sehingga Tugas Akhir ini dapat bermanfaat bagi semua pihak yang terkait.

Surabaya, Juli 2019

**Penulis**

## DAFTAR ISI

	Halaman
<b>HALAMAN JUDUL</b> .....	i
<b>TITLE PAGE</b> .....	iii
<b>LEMBAR PENGESAHAN</b> .....	<b>Error! Bookmark not defined.</b>
<b>ABSTRAK</b> .....	vii
<b>ABSTRACT</b> .....	ix
<b>KATA PENGANTAR</b> .....	xi
<b>DAFTAR ISI</b> .....	xiii
<b>DAFTAR GAMBAR</b> .....	xv
<b>DAFTAR TABEL</b> .....	xvii
<b>DAFTAR LAMPIRAN</b> .....	xix
<b>DAFTAR NOTASI</b> .....	xxi
<b>BAB I PENDAHULUAN</b> .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian .....	4
1.4 Manfaat Penelitian .....	5
1.5 Batasan Masalah .....	5
<b>BAB II TINJAUAN PUSTAKA</b> .....	7
2.1 Tumor Otak .....	7
2.2 Segmentasi Citra .....	7
2.2.1 Intensitas <i>Greyscale</i> .....	9
2.2.2 Histogram Citra.....	10
2.3 <i>Preprocessing</i> .....	11
2.3.1 <i>Edge Detection</i> .....	12
2.3.2 <i>Thresholding</i> .....	13
2.3.3 Operasi Morfologi.....	14
2.4 <i>Model Based Clustering</i> .....	16
2.5 <i>Gaussian Mixture Model (GMM)</i> .....	18
2.6 Optimasi Model.....	19
2.6.1 <i>Expectation Maximization (EM)</i> .....	20
2.6.2 <i>Reversible Jump Markov Chain Monte Carlo (RJMCMC)</i> .....	23
2.7 <i>Correct Classification Ratio (CCR)</i> .....	25

<b>BAB III METODOLOGI PENELITIAN</b> .....	27
3.1 Sumber Data.....	27
3.2 Variabel Penelitian.....	27
3.3 Struktur Data.....	27
3.4 Langkah Analisis Data.....	28
3.5 Diagram Alir.....	29
<b>BAB IV ANALISIS DAN PEMBAHASAN</b> .....	31
4.1 Pemilihan Data dan <i>Preprocessing</i> .....	31
4.1.1 <i>Preprocessing</i> .....	31
4.1.2 Histogram Citra MRI Tumor Otak.....	37
4.2 Segmentasi Citra dengan Metode <i>Gaussian Mixture Model</i> dengan pendekatan <i>Revesible Jump Markov Chain Monte Carlo</i> (GMM-RJMCMC).....	39
4.3 Segmentasi Citra dengan Metode <i>Gaussian Mixture Model</i> dengan algoritman <i>Expectation Maximization</i> (GMM-EM).....	42
4.4 Perbandingan Metode <i>Gaussian Mixture Model</i> (GMM) dengan EM dan RJMCMC.....	45
<b>BAB V KESIMPULAN DAN SARAN</b> .....	49
5.1 Kesimpulan.....	49
5.2 Saran.....	50
<b>DAFTAR PUSTAKA</b> .....	51
<b>LAMPIRAN</b> .....	54
<b>BIODATA PENULIS</b> .....	77



## DAFTAR GAMBAR

	Halaman
<b>Gambar 2.1</b>	Perubahan Nilai <i>Pixel</i> Citra RGB ke dalam Citra Abu-Abu ..... 10
<b>Gambar 2.2</b>	Histogram Citra..... 11
<b>Gambar 2.3</b>	Citra Hasil <i>Thresholding</i> ..... 13
<b>Gambar 2.4</b>	<i>Ground Truth</i> (a) Segmentasi MRI Tumor Otak (b) ROI Tumor Otak ..... 25
<b>Gambar 2.5</b>	<i>Confusion Matrix</i> dan Perhitungan CCR ..... 13
<b>Gambar 3.1</b>	Diagram Alir Penelitian ..... 30
<b>Gambar 4.1</b>	Citra MRI <i>Sequence ax T1 memp+C slice</i> ke-18 ..... 32
<b>Gambar 4.2</b>	Citra Tahapan <i>Thresholding</i> (a) Citra Masukan (b) Citra Setelah <i>Thresholding</i> ..... 32
<b>Gambar 4.3</b>	Citra Tahapan <i>Head Contour</i> (a) Citra Hasil <i>Thresholding</i> (b) Citra Setelah Penentuan <i>Head Contour</i> ..... 33
<b>Gambar 4.4</b>	Citra Tahapan <i>Head Masking</i> (a) Citra Hasil <i>Head Contour</i> (b) Citra Setelah dilakukan <i>Head Masking</i> .. 33
<b>Gambar 4.5</b>	Citra Tahapan Substraksi (a) Citra Head Mask (b) Citra Threshold (c) Citra Hasil Substraksi..... 34
<b>Gambar 4.6</b>	Citra Tahapan <i>Morphological Opening</i> (a) Citra Hasil Substraksi (b) Citra Hasil <i>Opening</i> ..... 34
<b>Gambar 4.7</b>	Citra Tahapan <i>Edge Detection</i> (a) Citra Hasil <i>Opening</i> (b) Citra Hasil <i>Edge Detection</i> ..... 35
<b>Gambar 4.8</b>	Citra Tahapan <i>Blocking</i> (a) Citra Hasil Penggabungan (b) Citra Hasil <i>Blocking</i> ..... 35
<b>Gambar 4.9</b>	Citra Tahapan Substraksi Area Otak (a) Citra Hasil <i>Blocking</i> (b) Citra Hasil Penggabungan (c) Citra Hasil Substraksi Area Otak..... 36
<b>Gambar 4.10</b>	Citra Tahapan <i>Morphological Opening</i> (a) Citra Hasil Substraksi (b) Citra Hasil <i>Opening</i> ..... 36
<b>Gambar 4.11</b>	Citra Akhir <i>Preprocessing</i> (a) Citra Hasil <i>Preprocessing</i> (b) Citra Area Otak ..... 37
<b>Gambar 4.12</b>	Citra Hasil <i>Preprocessing</i> (a) Citra Awal (b) Citra Setelah <i>Preprocessing</i> ..... 37
<b>Gambar 4.13</b>	Histogram Citra Sebelum <i>Preprocessing</i> ..... 38

<b>Gambar 4.14</b>	Histogram Citra Setelah <i>Preprocessing</i> .....	38
<b>Gambar 4.15</b>	Citra Hasil Segmentasi dengan GMM (a) Citra <i>Preprocessing</i> (b) Citra Hasil Segmentasi GMM-RJMCMC.....	39
<b>Gambar 4.16</b>	Histogram Citra Hasil Segmentasi menggunakan GMM-RJMCMC.....	40
<b>Gambar 4.17</b>	Citra Hasil Segmentasi GMM-RJMCMC (a) Area ROI (b) Area Non-ROI.....	41
<b>Gambar 4.18</b>	<i>Subjective Analysis</i> GMM-RJMCMC ROI Citra MRI.....	42
<b>Gambar 4.19</b>	Citra Hasil Segmentasi dengan GMM (a) Citra <i>Preprocessing</i> (b) Citra Hasil Segmentasi GMM-EM.....	43
<b>Gambar 4.20</b>	Histogram Hasil Segmentasi Citra menggunakan GMM-EM.....	43
<b>Gambar 4.21</b>	Citra Hasil Segmentasi GMM-EM (a) Area ROI (b) Area Non-ROI.....	45
<b>Gambar 4.22</b>	<i>Subjective Analysis</i> GMM-EM ROI Citra MRI.....	45

## DAFTAR TABEL

	Halaman
<b>Tabel 2.1</b> Tabulasi Silang Ketepatan Klasifikasi.....	26
<b>Tabel 3.1</b> Struktur Data .....	27
<b>Tabel 4.1</b> Estimasi Parameter Model Segmentasi GMM- RJMCMC 3 Cluster.....	40
<b>Tabel 4.2</b> Estimasi Parameter Model Segmentasi GMM-EM 3 Cluster .....	44
<b>Tabel 4.3</b> <i>Confusion Matrix</i> untuk Metode GMM EM .....	46
<b>Tabel 4.4</b> Confusion Matrix untuk Metode GMM- RJMCMC .....	47
<b>Tabel 4.5</b> Perbandingan Metode GMM-EM & GMM- RJMCMC...47	

*“Halaman ini sengaja dikosongkan”*

## DAFTAR LAMPIRAN

	Halaman
<b>Lampiran 1</b>	<i>Syntax Preprocessing – Thresholding</i> ..... 55
<b>Lampiran 2</b>	<i>Syntax Preprocessing – Head Contour dan Head Masking</i> ..... 56
<b>Lampiran 3</b>	<i>Syntax Preprocessing – Penentuan Area Otak</i> ..... 56
<b>Lampiran 4</b>	<i>Syntax Preprocessing Utama</i> ..... 58
<b>Lampiran 5</b>	<i>Syntax Segmentasi Metode Gaussian Mixture Model (GMM) dengan Algoritma Expectation-Maximization (EM)</i> ..... 59
<b>Lampiran 6</b>	<i>Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC)</i> ..... 63
<b>Lampiran 7</b>	<i>Syntax Perhitungan AUC dengan Software R</i> ..... 63
<b>Lampiran 8</b>	<i>Output Perhitungan AUC dengan Software R</i> ..... 64
<b>Lampiran 9</b>	Gambar Nilai Numerik Matriks Grayscale Citra MRI Data Inputan ..... 65
<b>Lampiran 10</b>	Surat Pernyataan Pengambilan Data Tugas Akhir ... 66

*“Halaman ini sengaja dikosongkan”*

## DAFTAR NOTASI

### *Preprocessing*

$p$	Indeks baris
$q$	Indeks kolom
$f(p,q)$	Intensitas <i>greyscale pixel</i> baris ke- $p$ kolom ke- $q$
$g(p,q)$	Citra biner dari citra <i>grayscale</i> $f(p,q)$
$l$	Intensitas <i>greyscale</i> bernilai 0-255
$h_l$	Proporsi intensitas <i>greyscale</i> $l$
$n_l$	Jumlah <i>pixel</i> yang memiliki intensitas <i>greyscale</i> $l$
$N$	Jumlah seluruh <i>pixel</i> di dalam citra
$y_i$	Nilai intensitas <i>greyscale pixel</i> ke- $i$
$G_x$	Nilai <i>operator sobel</i> garis horizontal
$G_y$	Nilai <i>operator sobel</i> garis vertikal
$T$	Nilai <i>threshold</i>
$G$	Himpunan <i>pixel</i> yang nilainya lebih dari sama dengan <i>threshold</i>
$L$	Himpunan <i>pixel</i> yang nilainya kurang dari <i>threshold</i>
$G_i$	Nilai intensitas $G$ ke- $i$
$L_i$	Nilai intensitas $L$ ke- $i$

### **GMM**

$i$	Indeks data
$j$	Indeks komponen/ <i>cluster</i>
$K$	Banyaknya komponen penyusun distribusi <i>mixture</i>
$f(y \theta, \pi)$	Fungsi densitas distribusi <i>mixture</i>
$f_j(y \theta_j)$	Fungsi densitas distribusi <i>mixture</i> komponen ke- $j$
$\theta_j$	Vektor parameter distribusi <i>mixture</i> ke- $j$
$\pi$	Vektor parameter proporsi
$\mu$	Vektor parameter <i>mean</i>
$\sigma$	Vektor parameter varians
$\pi_j$	Parameter proporsi komponen ke- $j$
$\mu_j$	Parameter <i>mean</i> komponen ke- $j$

$\sigma_j$	Parameter varians komponen ke- $j$
$\pi_{ij}$	Parameter proporsi data ke- $i$ komponen ke- $j$
$\Sigma_j$	Parameter kovarians komponen ke- $j$
$z_{ij}$	Variabel laten data ke- $i$ komponen ke- $j$
$Z$	<i>Normalizing constant</i>

***Correct Classification Ratio (CCR)***

$GT$	<i>Ground truth</i>
$GT_j$	<i>Ground truth cluster ke-<math>j</math></i>
$Seg_j$	Hasil segmentasi <i>cluster ke-<math>j</math></i>



# **BAB I**

## **PENDAHULUAN**

Bab ini berisikan tentang latar belakang sebagai awal mula penelitian ini dilakukan. Selanjutnya dilakukan perumusan masalah serta penetapan tujuan penelitian. Selain itu, pada bab ini juga membahas mengenai manfaat dan batasan masalah dari penelitian yang dilakukan.

### **1.1 Latar Belakang**

Otak merupakan salah satu organ vital yang dimiliki manusia. Menurut Danim tahun 2010, “Otak merupakan bagian dari struktur dasar yang menjadi modalitas seseorang dalam mencapai totalitas potensi yang dimilikinya”. Sehingga bisa dikatakan otak memiliki peran penting bagi kelangsungan hidup manusia. Namun, seperti halnya organ lain yang ada pada manusia otak juga bisa terserang berbagai macam penyakit. Salah satu penyakit yang sering terjadi pada otak adalah tumor otak.

Tumor otak merupakan pertumbuhan jaringan yang tidak normal dalam otak. Tumor otak dapat bersifat primer ataupun metastasis dari tumor pada organ lainnya. Tumor otak primer tumbuh dan cenderung bertahan di otak sedangkan tumor otak metastatis dimulai sebagai kanker di bagian tubuh dan menyebar ke otak (ABTA, 2015). Tumor otak yang menimbulkan kerusakan pada jaringan otak secara langsung akan menyebabkan gangguan fungsional dari sistem saraf pusat, berupa gangguan motorik, sensorik, panca indera, bahkan kemampuan kognitif (Wahjoepramono, 2006). Ketika tumor otak terjadi, pertumbuhan sel yang berlebihan menimbulkan penekanan dan kerusakan pada sel-sel lain di otak dan mengganggu fungsi otak bagian tersebut. Kemudian, tumor tersebut akan menekan ke jaringan sekitar otak yang sehat, tulang tengkorak, dan area sekitar syaraf (Cook & Freedman, 2012).

Jumlah kasus tumor otak di Indonesia terus meningkat setiap tahunnya, yakni pada tahun 2018 menembus angka 25000 pasien. Salah satu rumah sakit yang menjadi rujukan pasien dengan kasus tumor otak di Indonesia adalah Rumah Sakit (RS) X. RS X mencatat bahwa jumlah kasus penyakit tumor otak setiap tahunnya meningkat dan sebagian besar diderita oleh kaum perempuan. Pada tahun 2010 jumlah pasien yang menjalani operasi sebanyak 116 orang. Kemudian pada tahun

2011 sebanyak 130 orang, pada tahun 2012 sebanyak 130, tahun 2013 sebanyak 130 orang dan pada tahun 2014 sebanyak 172 orang. Dan sampai bulan Juni 2018, terdapat lebih dari 650 kasus tumor otak yang memerlukan operasi. (JatimTribun, 2018).

Berbagai upaya telah dilakukan guna membantu permasalahan tumor otak. Salah satunya dengan pendeteksian tumor otak secara akurat dengan alat medis yakni MRI. *Magnetic Resonance Imaging* atau disebut MRI merupakan salah satu peralatan medis yang digunakan untuk mendeteksi adanya tumor otak. MRI merupakan suatu jenis pemeriksaan radiologi yang menggunakan gelombang elektromagnet untuk memperoleh informasi mengenai berbagai kelainan pada tubuh. Informasi yang dihasilkan dari pemeriksaan MRI digunakan untuk mendukung diagnosis penyakit serta memantau perjalanan penyakit dan hasil terapi. MRI untuk tujuan medis beroperasi dengan kekuatan medan magnet 1,5 Tesla dan 3 Tesla (Santosa Hospital Bandung Kopo, 2018). Kualitas gambar hasil *scan* MRI 3 Tesla lebih baik dibandingkan dengan 1.5 Tesla, hal ini akan memperjelas kelainan-kelainan pada pasien tumor otak. Namun, pemeriksaan dengan menggunakan MRI 3 Tesla memerlukan biaya yang relatif lebih tinggi jika dibandingkan dengan MRI 1.5 Tesla (Canadian Agency for Drugs and Technologies in Health, 2011). Oleh karena itu diperlukan suatu upaya guna meningkatkan hasil kualitas dari citra MRI 1.5 Tesla sehingga biaya yang dikeluarkan tetap terjangkau.

Salah satu pendekatan komputasi untuk meningkatkan kualitas citra MRI tumor otak adalah dengan menggunakan segmentasi citra. Proses pemisahan wilayah suatu citra berdasarkan kriteria keserupaan tertentu antara derajat keabuan suatu piksel dengan derajat keabuan piksel-piksel tetangganya (Wijaya & Prayudi, 2010). Secara digital suatu derajat keabuan atau *grayscale* dapat direpresentasikan dalam bentuk *array* dua dimensi atau matriks. Tiap elemen pada *array* tersebut menunjukkan intensitas *grayscale* dari pada posisi koordinat yang bersesuaian. Apabila suatu citra direpresentasikan dalam 8 bit, pada citra terdapat  $2^8$  atau 256 level *grayscale*, biasanya bernilai antara 0-255 (Gonzales & Woods, 2008). Nilai 0 menunjukkan level intensitas yang paling gelap sedangkan 255 menunjukkan intensitas paling terang.

Metode segmentasi citra yang sering digunakan yaitu *clustering*, dimana *pixel* dalam citra akan dikelompokkan berdasarkan intensitas warna (derajat keabuan/ *grayscale*) yang sama. *Model Based Clustering* merupakan metode pengelompokkan yang mengoptimalkan kemiripan antara objek berdasarkan pada distribusi probabilitas data. *Model based clustering* mengasumsikan bahwa data yang dibangkitkan oleh campuran dari distribusi probabilitas dengan masing-masing komponen dapat mewakili *cluster* yang berbeda sehingga objek didalam suatu *dataset* dapat dikelompokkan menjadi kelompok data yang sebelumnya tidak terdefinisi. Beberapa *cluster* bisa memiliki distribusi yang sama atau berbeda-beda sehingga campuran dari distribusi probabilitas membentuk suatu *mixture model* (Shi, 2005). Salah satu *mixture model* yang umum digunakan dalam segmentasi citra adalah *Gaussian Mixture Model* (GMM) atau sering disebut *standard GMM*. Metode ini merupakan perangkat pemodelan statistik yang *robust* dan fleksibel untuk data multivariat, serta mudah diimplementasikan dengan jumlah parameter yang dibutuhkan sedikit (Nguyen dan Wu, 2013). Dalam pengestimasi parameter model pada GMM, metode yang sering digunakan adalah algoritma Expectation Maximization (EM). Selain itu bisa dengan menggunakan pendekatan Bayesian *Markov Chain Monte Carlo* (MCMC) seperti algoritma *Gibbs Sampling* dan *Metropolis Hasting*. Namun EM maupun MCMC ini memiliki kelemahan, yakni jumlah komponen *mixture* diasumsikan bersifat tertentu, sehingga perlu dilakukan pemilihan klaster optimum dengan menggunakan kriteria pemilihan klaster tertentu yang membuat proses segmentasi ini menjadi cukup rumit dan tidak otomatis.

Untuk mengatasi permasalahan dimensi vektor parameter model yang bersifat tidak tetap Green (1995) telah mengembangkan algoritma *metropolis hasting* yang disebut dengan *Reversible Jump Markov Chain Monte Carlo* (RJCMC). Penggunaan RJCMC pada pemodelan *mixture* dapat digunakan untuk menentukan jumlah komponen *mixture* yang tidak tetap (Astuti, 2017). Oleh karena itu pada penelitian ini akan dilakukan segmentasi citra MRI otak dengan metode *Gaussian Mixture Model* (GMM) dengan menggunakan algoritma *Reversible Jump Markov Chain Monte Carlo* (RJCMC). Sehingga diharapkan dapat mempercepat proses perhitungan komputasi serta hasil segmen-

tasi citra MRI menjadi lebih jelas agar dapat mempermudah pendeteksian tumor otak dan tindakan medis selanjutnya.

## 1.2 Rumusan Masalah

Pendeteksian tumor otak menggunakan mesin MRI 3 Tesla menghasilkan gambar yang lebih bagus daripada mesin MRI 1,5 Tesla. Namun, harga untuk hasil *scan* mesin MRI 3 Tesla terbilang mahal daripada mesin MRI 1,5 Tesla. Oleh karena itu diperlukan suatu pendekatan untuk meningkatkan kualitas dari citra MRI tumor otak hasil MRI 1.5 Tesla, sehingga dengan biaya yang lebih murah dari MRI 3 Tesla mampu menghasilkan kualitas gambar yang lebih baik dari MRI 1.5 Tesla. Pendekatan komputasi untuk meningkatkan kualitas citra MRI tumor otak dapat dilakukan dengan menggunakan seg-mentasi citra. Segmentasi citra merupakan proses pemisahan wilayah suatu citra berdasarkan kriteria keserupaan tertentu antara derajat keabuan suatu piksel dengan derajat keabuan piksel-piksel tetangganya. *Gaussian mixture model* (GMM) merupakan model *mixture* yang paling sering digunakan dalam *model based clustering* khususnya pada segmentasi citra. Dalam pengestimasi parameter model pada GMM, metode yang sering digunakan adalah algoritma Expectation Maximization (EM). Selain itu bisa dengan menggunakan pendekatan Bayesian *Markov Chain Monte Carlo* (MCMC) seperti algoritma *Gibbs Sampling* dan *Metropolis Hasting*. Namun EM maupun MCMC ini memiliki kelemahan, yakni jumlah komponen *mixture* diasumsikan bersifat tertentu, sehingga perlu dilakukan pemilihan kluster optimum dengan menggunakan kriteria pemilihan kluster tertentu yang membuat proses segmentasi ini menjadi cukup rumit dan tidak otomatis. Oleh karena itu pada penelitian ini akan dilakukan segmentasi citra MRI tumor otak menggunakan *Gaussian Mixture Model* (GMM) dengan estimasi parameter menggunakan algoritma *Reversible Jump Markov Chain Monte Carlo* (RJCMCMC).

## 1.3 Tujuan Penelitian

Berdasarkan permasalahan yang telah diuraikan maka tujuan dari penelitian ini adalah mendapatkan hasil segmentasi citra MRI tumor otak menggunakan *Gaussian Mixture Model* (GMM) dengan estimasi parameter menggunakan algoritma *Reversible Jump Markov Chain Monte Carlo* (RJCMCMC).

#### **1.4 Manfaat Penelitian**

Penelitian ini diharapkan dapat memberikan kontribusi dalam dunia medis, khususnya dalam kasus tumor otak. Yakni dapat memberikan hasil citra MRI yang lebih baik dari pada citra MRI yang telah dihasilkan sebelumnya. Sehingga akan mempermudah pendeteksian tumor otak dan tindakan medis selanjutnya. Serta bagi bidang keilmuan statistika khususnya bidang komputasi diharapkan dapat memberikan sumbangsi ilmu pengetahuan dan dapat menjadi acuan bagi peneliti-peneliti yang lainnya.

#### **1.5 Batasan Masalah**

Adapun batasan masalah yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Data yang digunakan adalah citra MRI otak untuk pasien kasus tumor otak, yang merupakan data sekunder dari RS X.
2. Mesin MRI yang digunakan adalah MRI yang berkekuatan medan magnet 1,5 Tesla.

*“Halaman ini sengaja dikosongkan”*

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisikan tentang landasan teori yang digunakan untuk membantu dalam menyelesaikan permasalahan pada penelitian ini.

#### **2.1 Tumor Otak**

Tumor otak merupakan pertumbuhan jaringan yang tidak normal dalam otak. Tumor otak dapat bersifat primer ataupun metastasis dari tumor pada organ lainnya. Tumor otak primer tumbuh dan cenderung bertahan di otak sedangkan tumor otak metastatis dimulai sebagai kanker di bagian tubuh dan menyebar ke otak (American Brain Tumor Association, 2015). Tumor otak yang menimbulkan kerusakan pada jaringan otak secara langsung akan menyebabkan gangguan fungsional dari sistem saraf pusat, berupa gangguan motorik, sensorik, serta panca indera, bahkan kemampuan kognitif (Wahjoepramono, 2006). Pada saat tumor otak terjadi, pertumbuhan sel yang berlebihan menimbulkan penekanan dan kerusakan pada sel-sel lain di otak dan mengganggu fungsi otak bagian tersebut. Kemudian, tumor tersebut akan menekan jaringan sekitar otak, tengkorak, jaringan otak yang sehat dan area sekitar syaraf (Cook & Freedman, 2012).

*Magnetic Resonance Imaging* atau disebut MRI merupakan salah satu peralatan medis yang digunakan untuk mendeteksi adanya tumor otak. Informasi yang dihasilkan dari pemeriksaan MRI digunakan untuk mendukung diagnosis penyakit serta memantau perjalanan penyakit dan hasil terapi. MRI untuk tujuan medis beroperasi dengan kekuatan medan magnet 1,5 Tesla dan 3 Tesla (Santosa Hospital Bandung Kopo, 2018). Kualitas gambar hasil *scan* MRI 3 Tesla lebih baik dibandingkan dengan 1.5 Tesla, hal ini akan memperjelas kelainan-kelainan pada pasien tumor otak. Namun, pemeriksaan dengan menggunakan MRI 3 Tesla memerlukan biaya yang relative tinggi dibanding 1.5 Tesla (Canadian Agency for Drugs & Technologies in Health, 2011).

#### **2.2 Segmentasi Citra**

Citra (*image*) yang merupakan istilah lain untuk gambar merupakan salah satu komponen multimedia yang memegang peranan sangat penting sebagai bentuk informasi visual. Secara harafiah, citra (*image*) adalah gambar pada bidang dua dimensi. Ditinjau dari sudut

pandang matematis, citra merupakan fungsi menerus (*continue*) dari intensitas cahaya pada bidang dua dimensi (Munir, 2004). Dengan adanya teknologi digital yang semakin berkembang maka pemakaian gambar atau citra digital dapat dimanfaatkan untuk mengetahui sesuatu yang terjadi pada jaringan tubuh. Dengan data citra hasil MRI dapat dilakukan analisis terhadap jaringan tubuh. Namun banyak terdapat kesulitan dalam mengolah gambar/citra digital hasil MRI. Salah satu kesulitan yang ada adalah dalam melakukan manipulasi pada satu objek saja tanpa menyentuh objek lain, hal ini disebabkan karena sulitnya memisahkan objek satu dengan objek yang lain. Oleh karena itu, untuk memisahkan satu objek dengan yang lain perlu dilakukan suatu metode yaitu segmentasi citra (*image segmentation*).

Segmentasi citra berfungsi untuk memisahkan antara objek yang satu dengan objek yang lain pada gambar. Pada citra MRI segmentasi citra dilakukan untuk memisahkan *Region of Interest* (ROI) berupa segmen yang dianggap penting secara medis, dengan segmen-segmen lainnya (*Non-Region of Interest*). Segmentasi merupakan langkah awal dan menjadi kunci pembahasan dalam pengenalan objek (*object recognition*). Kualitas hasil akhirnya sebagian besar tergantung pada kualitas segmentasi itu sendiri. Selain itu, segmentasi citra merupakan proses intermediasi yang penting sebelum melangkah ke hasil akhir dari suatu sistem klasifikasi atau sistem pengenalan objek. Dengan proses segmentasi masing-masing objek pada citra MRI dapat diambil secara terpisah sehingga dapat digunakan sebagai masukan proses yang lain. Segmentasi citra (*image segmentation*) merupakan suatu teknik pengelompokan untuk citra (Adipranata, 2005).

Metode segmentasi berbasis *cluster* merupakan salah satu bagian penting dari pemrosesan citra yang bertujuan untuk pengelompokan data atau penggolongan objek berdasarkan informasi yang diperoleh dalam kelas-kelas, menjadi beberapa region yang homogen berdasarkan kriteria kemiripan tertentu, sehingga data dalam suatu *cluster* memiliki tingkat persamaan yang tinggi satu dengan lainnya, tetapi sangat berbeda dengan data *cluster* lain (Xu & Wunsch, 2008). Dengan kata lain, hal ini merupakan suatu proses pembagian citra ke dalam beberapa wilayah (*region*) yang mempunyai kesamaan fitur seperti



tingkat keabuan (*graylevel*), tekstur (*texture*), warna (*color*), gerakan (*motion*), dan lain sebagainya.

### 2.2.1 Intensitas Greyscale

Secara digital suatu *grayscale* (keabuan) dapat direpresentasikan dalam bentuk *array* dua dimensi atau matriks. Tiap elemen pada *array* tersebut menunjukkan intensitas *grayscale* dari citra pada posisi koordinat yang bersesuaian dan disebut sebagai *picture element* atau *pixel*. Dengan melakukan perubahan pada intensitas masing-masing *pixel* maka representasi citra tersebut secara keseluruhan akan berubah. Misalkan suatu citra yang direpresentasikan dalam 8 bit, maka pada citra terdapat  $2^8$  atau 256 level *grayscale*, biasanya bernilai 0-255 (Gonzalez & Wood, 2008). Dalam hal ini nilai intensitas 0 menunjukkan level intensitas yang paling gelap (hitam) dan 255 menunjukkan level intensitas yang paling terang (putih).

Citra digital yang berukuran  $P \times Q$  dinyatakan dengan matriks yang berukuran  $P$  baris dan  $Q$  kolom. Representasi citra dalam *matrix* dapat ditunjukkan pada persamaan (2.1), dimana setiap  $\mathbf{f}(p,q)$  merupakan intensitas keabuan (*grayscale*) dari citra tiap-tiap lokasi *pixel*-nya.

$$\mathbf{f}(p,q) = \begin{bmatrix} f(0,0) & f(0,1) & \dots & f(0,Q-1) \\ f(1,0) & \dots & \dots & f(1,Q-1) \\ \dots & \dots & \dots & \dots \\ f(P-1,0) & f(P-1,1) & \dots & f(P-1,Q-1) \end{bmatrix} \quad (2.1)$$

Indeks baris ( $p$ ) untuk  $p = 0, 1, 2, \dots, P$  dan indeks kolom ( $q$ ) untuk  $q = 0, 1, 2, \dots, Q$  menyatakan suatu koordinat titik pada citra, sedangkan  $f(p,q)$  merupakan intensitas *grayscale* pada titik ( $p,q$ ) (Gonzalez & Wood, 2008).

Suatu citra memiliki komposisi *Red* (R), *Green* (G), dan *Blue* (B) pada tiap *pixel*-nya. Proses perubahan citra berwarna ke dalam citra abu-abu dapat dilakukan dengan pembobotan. Persamaan metode pembobotan ini dapat dirumuskan pada persamaan berikut (Shi, 2005).

$$\text{intensitas grayscale} = 0,299R + 0,578G + 0,114B \quad (2.2)$$

Berdasarkan persamaan (2.2) dapat dilakukan perubahan nilai komposisi RGB ke dalam citra abu-abu, seperti diilustrasikan pada Gambar 2.1.

R=200	R=50	R=140	
G=0	G=50	G=0	
B=100	B=80	B=100	
R=200	R=80	R=20	→
G=200	G=80	G=0	
B=20	B=95	B=10	
R=10	R=200	R=0	
G=0	G=0	G=0	
B=20	B=100	B=0	

71	52	53
198	80	7
5	71	0

**Gambar 2.1** Perubahan Nilai *Pixel* Citra RGB ke dalam Citra Abu-Abu

### 2.2.2 Histogram Citra

Informasi penting mengenai isi citra digital dapat diketahui dengan membuat histogram citra. Histogram citra adalah grafik yang menggambarkan penyebaran nilai-nilai intensitas *pixel* dari suatu citra atau bagian tertentu di dalam citra. Dari sebuah histogram dapat diketahui frekuensi kemunculan nisbi (*relative*) dari intensitas pada citra tersebut. Histogram juga dapat menunjukkan banyak hal tentang kecerahan (*brightness*) dan kontras (*contrast*) dari citra (Munir, 2004).

Misalkan citra digital memiliki  $l$  intensitas *greyscale*, yaitu dari nilai 0 sampai 255 (misalnya pada citra dengan kuantisasi intensitas *greyscale* 8-bit). Secara matematis histogram citra dihitung dengan rumus:

$$h_l = \frac{n_l}{N} ; l = 0, 1, \dots, 255 \quad (2.3)$$

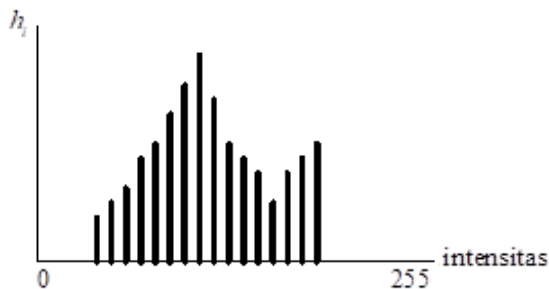
dimana,

$h_l$  = proporsi intensitas derajat keabuan  $l$

$n_l$  = jumlah *pixel* yang memiliki derajat keabuan  $l$

$N$  = jumlah seluruh *pixel* di dalam citra

Plot  $h_i$  versus intensitas dinamakan histogram. Gambar 2.2 adalah contoh sebuah histogram citra. Secara grafis histogram ditampilkan dengan diagram batang (Munir, 2004).



Gambar 2.2 Histogram Citra

Dalam penelitian ini, analisis *cluster* yang digunakan adalah pengelompokan (*clustering*) berbasis pola data atau *Model Based Clustering*. Sehingga bentuk histogram citra akan sangat membantu untuk mengenali pola *mixture*-nya. Setelah pola *mixture* didapatkan, akan diperoleh dugaan awal banyaknya mode yang digunakan dalam menduga model.

### 2.3 Preprocessing

Citra MRI otak yang digunakan terdiri dari banyak atribut dan label film, yang mencakup nama pasien, usia dan tanda untuk identifikasi MRI. Pengupasan tengkorak (*skull stripping*) dan penghapusan atribut film ini diperlukan untuk menghilangkan tengkorak dan daerah latar belakang dari MRI untuk menganalisis jaringan otak dengan baik. Proses ini dilakukan dengan menggunakan *edge detection* dengan *operator sobel* dan *thresholding*. Nilai intensitas *pixel* dianalisis dengan menggunakan nilai *threshold* atribut film. Jika nilai intensitas *pixel* lebih besar dari nilai *threshold* maka intensitas *pixel* dibuat menjadi nol dan dihapus dari MRI. Nilai intensitas atribut film yang tinggi dikeluarkan dari citra MRI otak. Nilai *threshold* yang terpisah ditetapkan untuk label dan daerah tengkorak sehingga bagian otak MRI yang tidak diinginkan dapat dilepas. (Lee, et al., 2003).

Peningkatan kualitas gambar (*image enhancement*) merupakan langkah *preprocessing* mendasar yang digunakan dalam analisis citra.

Proses ini dilakukan untuk meningkatkan interpretabilitas gambar dengan meningkatkan kontras. Biasanya, citra MRI otak tidak memiliki kontras yang baik karena keterbatasan algoritma rekonstruksi gambar atau karena sifat fisiologi itu sendiri. Operasi morfologi akan digunakan dalam *image enhancement*.

### 2.3.1 Edge Detection

*Edge detection* atau deteksi tepi merupakan salah satu tahap dari *preprocessing* segmentasi citra untuk melepaskan bagian tengkorak dengan area otak (*skull stripping*). *Operator Sobel* merupakan algoritma yang digunakan untuk mendeteksi tepian gambar. Masing-masing *pixel* pada citra yang telah diinput akan dikonvolusi menggunakan *matrix* berukuran 3 x 3. Sebagai ilustrasi, tiap-tiap *pixel* representasi citra pada persamaan (2.4) akan dikalikan dengan *operator sobel*, baik secara vertikal menggunakan persamaan (2.5), dan secara horizontal menggunakan persamaan (2.6) (Prasetyo, 2011).

$$\begin{pmatrix} Z_1 & Z_2 & Z_3 \\ Z_4 & Z_5 & Z_6 \\ Z_7 & Z_8 & Z_9 \end{pmatrix} \quad (2.4)$$

$$\begin{pmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{pmatrix} \quad (2.5)$$

$$\begin{pmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{pmatrix} \quad (2.6)$$

Persamaan Gradien *Sobel* untuk horizontal dan vertikal masing-masing ditunjukkan oleh persamaan (2.7) dan persamaan (2.8).

$$G_x = (Z_7 + 2Z_8 + Z_9) - (Z_1 + 2Z_2 + Z_3) \quad (2.7)$$

$$G_y = (Z_3 + 2Z_6 + Z_9) - (Z_1 + 2Z_4 + Z_7) \quad (2.8)$$

Kombinasi kedua *kernel* tersebut digunakan untuk menghitung jarak absolut sebuah *gradient* namun bisa juga diaplikasikan secara terpisah untuk menghitung masing-masing proses vertikal dan horizontal. Untuk menghitung jarak *gradient*, digunakan persamaan (2.9) (Prasetyo, 2011).

$$|G| = \sqrt{G_x^2 + G_y^2} \quad (2.9)$$

dimana,

$G_x$  = nilai operator *sobel* garis horizontal

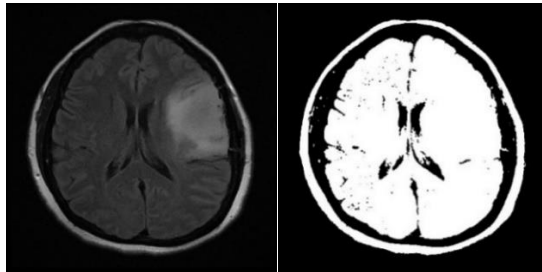
$G_y$  = nilai operator *sobel* garis vertikal

### 2.3.2 Thresholding

Secara umum proses *thresholding* terhadap citra *grayscale* bertujuan menghasilkan citra biner, secara matematis dapat ditulis didalam persamaan (2.10).

$$g(p,q) = \begin{cases} 1, & \text{jika } f(p,q) \geq T \\ 0, & \text{jika } f(p,q) < T \end{cases} \quad (2.10)$$

dimana  $g(p,q)$  merupakan citra biner dari citra *grayscale*  $f(p,q)$  pada persamaan (2.1) dan  $T$  menyatakan nilai *threshold* (Prasetyo, 2011). Ilustrasi hasil proses *thresholding* dapat dilihat pada Gambar 2.3 (Sianipar, 2017).



**Gambar 2.3** Citra Hasil *Thresholding*

Berdasarkan proses *thresholding* pada Gambar 2.3, citra awal yang memiliki intensitas abu-abu diubah menjadi citra biner berdasarkan tingkat *thresholding*-nya. Salah satu metode *thresholding* yang paling sederhana adalah *thresholding* dengan Metode Ridler-Calvard. Metode Ridler-Calvard pertama kali dipublikasikan pada Agustus 1978, dan metode ini akan bekerja dengan baik jika histogram citra bukan merupakan *bi-modal* (Sonka, Hlavac, & Boyle, 1993).

Langkah-langkah *Thresholding* dengan Metode Ridler-Calvard

1. Tentukan Nilai *threshold*

Nilai *threshold* didapat berdasarkan persamaan (2.11).

$$T = \frac{\sum_{i=1}^N y_i}{N} \quad (2.11)$$

dimana,

$T$  = nilai *threshold* pertama

$y_i$  = nilai intensitas *pixel*

$N$  = jumlah data, nilai  $N$  berupa bilangan bulat

## 2. Kelompokkan data citra masukan

Setelah nilai *threshold* didapat, nilai intensitas *pixel* yang lebih dari nilai *threshold* akan dikelompokkan dalam variabel tersendiri. Begitu juga nilai intensitas *pixel* kurang dari sama dengan nilai *threshold*. Pengelompokan data didasarkan pada persamaan (2.12) dan (2.13).

$$y_i \in G, \text{ jika } y_i \geq T \quad (2.12)$$

$$y_i \in L, \text{ jika } y_i < T \quad (2.13)$$

dimana:

$y_i$  = nilai intensitas *pixel*

$G$  = himpunan *pixel* yang nilainya lebih dari sama dengan *threshold*

$L$  = himpunan *pixel* yang nilainya kurang dari *threshold*

### 2.3.3 Operasi Morfologi

Operasi morfologi dalam tahap *image preprocessing* dilakukan terhadap serangkaian *pixel-pixel* citra dengan tujuan untuk meningkatkan aspek khusus dari struktur citra sehingga dapat lebih mudah untuk dihitung dan dikenali. Operasi morfologi digital pada *image processing* diantaranya adalah erosi dan dilasi serta *opening* dan *closing*.

#### a. Erosi dan Dilasi

Erosi merupakan proses penghapusan titik-titik batas objek menjadi bagian dari latar, berdasarkan *structuring element* yang digunakan. Pada operasi ini, ukuran objek diperkecil dengan mengikis sekeliling objek. Ada 2 cara yang dapat dilakukan untuk proses erosi, yaitu:

1. Dengan mengubah semua titik batas menjadi titik latar
2. Dengan menseset semua titik di sekeliling titik latar menjadi titik latar

Jika diketahui himpunan  $A$  dan  $B$ , maka erosi  $A$  oleh  $B$  (dinotasikan  $A \ominus B$ ) didefinisikan didalam persamaan (2.14) (Gonzalez & Wood, 2008).

$$A \ominus B = \{w: B_w \subseteq A\} \quad (2.14)$$

Dengan kata lain, erosi  $A$  oleh  $B$  terdiri atas semua titik  $w = (x, y)$  dimana  $B_w$  ada di dalam himpunan  $A$ . Untuk melakukan erosi,  $B$  dipindah-pindahkan dalam  $A$  dan dicari dimana saja  $B$  benar-benar ada di dalam  $A$ . Untuk kondisi-kondisi yang memenuhi syarat tersebut maka tandailah titik  $(0,0)$  yang bersesuaian dengan  $B$ . Titik-titik inilah yang merupakan hasil erosi  $A$  oleh  $B$  (Gonzalez & Wood, 2008).

Sementara itu, dilasi merupakan proses penggabungan titik-titik latar menjadi bagian dari objek, berdasarkan *structuring element* yang digunakan. Proses ini adalah kebalikan dari erosi, yaitu merubah latar disekeliling objek menjadi bagian dari objek tersebut. Ada 2 cara untuk melakukan operasi ini, yaitu:

1. Dengan cara mengubah semua titik latar yang bertetangga dengan titik batas menjadi titik objek, atau lebih mudahnya setiap titik yang tetangganya adalah titik objek menjadi titik objek.
2. Dengan mengubah semua titik di sekeliling titik batas menjadi titik objek, atau lebih mudahnya set semua titik tetangga sebuah titik objek menjadi titik objek.

Misalkan  $A$  dan  $B$  adalah himpunan-himpunan piksel. Dilasi  $A$  oleh  $B$  dinotasikan dengan  $A \oplus B$  dan didefinisikan didalam persamaan (2.15).

$$A \oplus B = \bigcup_{x \in B} A_x \quad (2.15)$$

Hal ini menunjukkan bahwa untuk setiap titik  $x \in B$ , maka dilakukan translasi atau pergeseran dan kemudian menggabungkan seluruh hasilnya (*union*) (Prasetyo, 2011).

#### b. *Opening* dan *Closing*

*Opening* merupakan proses morfologi yang menggabungkan proses dilasi dan erosi. Pada mulanya, citra dilakukan erosi kemudian dilasi. Tujuan *opening* adalah menghilangkan objek-objek yang ukurannya lebih kecil dari *structure element* dan memisahkan objek yang

semula bergandengan (Gonzalez & Wood, 2008). *Opening* dapat didefinisikan di dalam persamaan (2.16).

$$A \circ B = (A \ominus B) \oplus B \quad (2.16)$$

Sehingga operasi *opening* merupakan sebuah operasi yang terdiri atas operasi erosi diikuti oleh operasi dilasi. Definisi ekuivalennya dapat dinyatakan didalam persamaan (2.17) (Gonzalez & Wood, 2008).

$$A \circ B = \bigcup \{B_w : B_w \subseteq A\} \quad (2.17)$$

dimana  $A \circ B$  merupakan gabungan (*union*) dari seluruh pergeseran  $B$  yang benar-benar tercakup (*fit in*) dalam  $A$ .

Sementara itu, *closing* didefinisikan sebagai operasi dilasi yang dilanjutkan dengan operasi erosi, dinotasikan sebagai  $A \bullet B$ , sehingga dapat dinyatakan didalam persamaan (2.18) (Gonzalez & Wood, 2008).

$$A \bullet B = (A \oplus B) \ominus B \quad (2.18)$$

Tujuan dari morfologi *closing* adalah mengisi lubang-lubang kecil pada objek dan menggabungkan objek yang berdekatan.

## 2.4 Model Based Clustering

Analisis *cluster* adalah metode untuk menemukan kelompok kohesif berdasarkan karakteristik terukur menggunakan pengukuran numerik. Penelitian biasanya didasarkan pada seperangkat metode heuristik, seperti metode partisi dan pengelompokan hierarkis. Sebagai pendekatan yang sangat mudah, dalam metode heuristik sulit untuk mengetahui berapa banyak *cluster* yang kita butuhkan, bagaimana membandingkan kinerja antara metode, dan tidak ada cara untuk menghadapi *outlier*. Untuk lebih memahami kinerja *clustering*, perlu digunakan model probabilitas untuk mengakses semua ketidakpastian ini. Apakah metode ini cenderung lebih baik daripada yang lain atau terdapat metode lain yang lebih baik (Shi, 2005).

*Model Based Clustering* (*clustering* berbasis model) adalah metode yang didasarkan pada model probabilitas dari data. Diasumsikan data berasal dari beberapa fungsi distribusi. Sehingga konsep untuk membagi data ke dalam dua kelompok adalah data berasal dari gabungan dua model probabilitas yang berbeda. Untuk memodelkan data, model *mixture* sangat berguna dalam praktik. Dalam penelitian data umumnya heterogen, bukan satu kelompok homogen, dan sering kali dapat dimodelkan dengan distribusi *mixture*. Diasumsikan jika



hanya terdapat jumlah *cluster* yang terbatas (*finite*) dalam model, jumlah *cluster* tetap, dan data tersebut berasal dari kelompok yang berbeda, maka di setiap *cluster* data diasumsikan berasal dari beberapa distribusi probabilitas (Shi, 2005).

*Finite Mixture Model* (FMM) adalah kombinasi linier tertimbang dari sejumlah distribusi komponen sederhana, dapat didekati dengan formula (Iriawan, 2011) (McLachlan & Basford, 1988):

$$f(y | \boldsymbol{\theta}, \boldsymbol{\pi}) = \sum_{j=1}^K \pi_j f_j(y | \boldsymbol{\theta}_j) \quad (2.19)$$

dimana,

$f(y | \boldsymbol{\theta}, \boldsymbol{\pi})$  = fungsi densitas distribusi *mixture*

$f_j(y | \boldsymbol{\theta}_j)$  = fungsi densitas komponen ke- $j$

$\boldsymbol{\theta}_j$  = vektor semua parameter yang tidak diketahui dalam distribusi *mixture*  $\{\theta_1, \theta_2, \dots, \theta_K\}$

$\boldsymbol{\pi}$  = vektor parameter proporsi dengan elemen-elemen  $\{\pi_1, \pi_2, \dots, \pi_K\}$

$\pi_j$  = parameter proporsi komponen distribusi *mixture* dengan

$$\sum_{j=1}^K \pi_j = 1 \text{ dan } 0 \leq \pi_j \leq 1$$

$K$  = banyaknya distribusi sebagai komponen penyusun distribusi *mixture* yang telah ditentukan

Untuk mengestimasi parameter distribusi *mixture*, banyak pendekatan telah dikembangkan, seperti metode grafis, metode momen, metode jarak minimum, *maximum likelihood*, dan pendekatan Bayesian. Namun rumus eksplisit untuk estimasi parameter biasanya tidak tersedia (McLachlan & Peel, 2000). Metode estimasi *maximum likelihood* sejauh ini merupakan pendekatan yang paling umum digunakan untuk menaksir distribusi campuran dengan fungsi *likelihood* sebagai berikut.

$$L(\mathbf{y} | \boldsymbol{\theta}) \propto \prod_{i=1}^N f_{\text{Mixture}}(y_i | \boldsymbol{\theta}_j) = \prod_{i=1}^N \sum_{j=1}^K \pi_j f_j(y_i | \boldsymbol{\theta}_j) \quad (2.20)$$

Dalam hal ini sering lebih mudah untuk bekerja dengan fungsi log dari *likelihood* dimana konstanta aditif sering berubah-ubah, dengan formula:

$$\ell(\boldsymbol{\theta}; \mathbf{y}) = \prod_{i=1}^N \log(f_{\text{Mixture}}(y_i | \boldsymbol{\theta}_j)) = \sum_{i=1}^N \log \left( \sum_{j=1}^K \pi_j f_j(y_i | \boldsymbol{\theta}_j) \right) \quad (2.21)$$

Umumnya tidak ada solusi bentuk tertutup langsung untuk memaksimalkan fungsi log *likelihood* ini karena banyak parameter di dalam logaritma model dasarnya adalah distribusi campuran. Fungsi log *likelihood* menyebabkan masalah optimasi non-linier. Banyak metode telah diterapkan untuk memecahkan masalah ini (Shi, 2005). Teknik analisis laten yang dapat digunakan adalah algoritma relokasi seperti *Expectation-Maximization* (EM).

## 2.5 Gaussian Mixture Model (GMM)

*Gaussian Mixture Model* (GMM) merupakan suatu FMM ketika distribusi Normal atau *Gaussian* dipilih sebagai fungsi komponennya (McLachlan & Peel, 2000). Misal diketahui  $y_1, y_2, \dots, y_N$  merupakan kumpulan *pixel* dalam citra MRI yang diasumsikan berdistribusi *mixture Gaussian* atau biasa disebut *Gaussian Mixture Model* (GMM). Fungsi densitas GMM adalah sebagai berikut:

$$f(y | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \sum_{j=1}^K \pi_j f(y_i | \mu_j, \sigma_j^2) \quad (2.22)$$

dengan fungsi  $f(y_i | \mu_j, \sigma_j^2)$  diberikan oleh pdf dari *Gaussian* univariat sebagai berikut:

$$f(y_i | \mu_j, \sigma_j^2) = \frac{1}{\sqrt{2\pi\sigma_j}} \exp \left( -\frac{1}{2} \left( \frac{y_i - \mu_j}{\sigma_j} \right)^2 \right) \quad (2.23)$$

dimana  $\boldsymbol{\pi} = (\pi_1, \dots, \pi_K)$ ,  $0 \leq \pi_j \leq 1$ ,  $j = 1, 2, \dots, K$ ,  $\sum_{j=1}^K \pi_j = 1$ ,  $0 < y_i < 255$ ,  $0 < \mu_j < 255$ , dan  $\sigma_j^2 > 0$ . Dengan demikian, fungsi *joint conditional density* dari kumpulan *pixel*  $y = y_1, y_2, \dots, y_N$  dapat dimodelkan menjadi:

$$p(\mathbf{y} | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \prod_{i=1}^N f(y_i | \boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\sigma}) = \prod_{i=1}^N \left[ \sum_{j=1}^K \pi_j f(y_i | \mu_j, \sigma_j) \right] \quad (2.24)$$

Dari fungsi *joint conditional density* pada persamaan (2.24), fungsi *log-likelihood* dari GMM diberikan oleh:

$$L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y}) = \sum_{i=1}^N \log \left( \sum_{j=1}^K \pi_j \frac{1}{\sqrt{2\pi}\sigma_j} \exp \left( -\frac{1}{2} \left( \frac{y_i - \mu_j}{\sigma_j} \right)^2 \right) \right) \quad (2.25)$$

dapat dilihat dari fungsi *likelihood* pada persamaan (2.25), salah satu keuntungan digunakannya GMM adalah bentuk fungsi densitas yang sederhana, dan membutuhkan jumlah parameter yang tidak banyak (Nguyen, 2011).

## 2.6 Optimasi Model

Optimasi (*Optimization*) adalah aktivitas untuk mendapatkan hasil terbaik di bawah keadaan yang diberikan. Tujuan akhir dari semua aktivitas tersebut adalah meminimumkan usaha (*effort*) atau memaksimalkan manfaat (*benefit*) yang diinginkan. Karena usaha yang diperlukan atau manfaat yang diinginkan dapat dinyatakan sebagai fungsi dari variabel keputusan, maka optimasi dapat didefinisikan sebagai proses untuk menemukan kondisi yang memberikan nilai minimum atau maksimum dari sebuah fungsi.

Di dalam dunia statistika, optimasi dapat diterapkan pada suatu model atau persamaan. Hal ini dimaksudkan untuk mengoptimalkan bentuk model atau persamaan tersebut agar model atau persamaan tersebut dapat sedekat mungkin dalam merepresentasikan populasi. Caranya adalah dengan mengoptimalkan parameter-parameter yang ada pada suatu model atau persamaan.

Ada banyak metode dalam pengestimasi parameter model *mixture* normal. Salah satu metode yang sering digunakan adalah EM, MCMC dan lain-lain. Pada penelitian ini optimasi model yang dilakukan adalah dengan menggunakan algoritma EM dalam pengestimasi parameter model dan algoritma RJMCMC dalam penentuan jumlah kluster optimum secara otomatis.

### 2.6.1 *Expectation-Maximization (EM)*

Algoritma EM merupakan sebuah metode optimasi iteratif untuk estimasi *Maximum Likelihood (ML)* yang digunakan pada permasalahan data yang tidak lengkap (*incomplete data*). Algoritma ini banyak diterapkan pada berbagai implementasi keilmuan Statistika, karena algoritma EM cukup sederhana namun pencapaian konvergen dalam iterasi stabil (Dempster, Laird, & Rubin, 1977). Untuk memaksimalkan fungsi *likelihood* pada persamaan (2.25) dan menentukan parameter dari GMM akan digunakan algoritma EM.

Dalam setiap iterasi pada algoritma EM terdapat 2 tahap, yaitu tahap Ekspektasi atau *E-step* dan tahap Maksimisasi atau *M-step*. Pada tahap *E-step* dilakukan perhitungan nilai ekspektasi dari parameter data, sementara itu pada *M-step* dilakukan pengoptimalan nilai estimasi parameter dengan menggunakan nilai ekspektasi yang ditemukan pada tahap sebelumnya. Kedua tahapan ini akan terus dilakukan sampai mencapai nilai konvergen (Dempster, Laird, & Rubin, 1977)

Pada GMM, persoalan yang menjadi fokus penelitian adalah menentukan berapa peluang suatu *pixel* pengamatan  $y_i$  akan masuk ke *cluster* tertentu. Besaran peluang tersebut dapat diukur melalui sebuah *hidden variable*/variabel laten  $z$ . *E-step* dalam algoritma EM adalah untuk mengukur nilai ekspektasi dari variabel laten  $z$  tersebut (Dempster, Laird, & Rubin, 1977).

Dalam menyusun algoritma EM untuk GMM, langkah pertama yang dilakukan adalah menurunkan fungsi *log-likelihood* pada persamaan (2.25), terhadap *mean*  $\mu_j$  dari komponen Gaussian.

$$\frac{\partial L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y})}{\partial \mu_j} = \frac{\partial \left( \sum_{i=1}^N \log \left( \sum_{j=1}^K \pi_j f_j(y_i | \boldsymbol{\theta}_j) \right) \right)}{\partial \mu_j} \quad (2.26)$$

$$0 = - \sum_{i=1}^N \frac{\pi_j f(y_i | \mu_j, \sigma_j)}{\sum_{l=1}^K \pi_l f(y_i | \mu_l, \sigma_l)} \frac{y_i - \mu_j}{\sigma_j^2}$$

dimana  $\sum_{l=1}^K \pi_l f(y_i | \mu_l, \sigma_l)$  merupakan variabel laten  $z_{ij}$ , dengan ekspektasi probabilitas posterior:

$$\begin{aligned} z_{ij}^{(t)} &= f(z_{ij} | y_i, \mu_j^{(t)}, \sigma_j^{(t)}) \\ &= \frac{f(y_i | z_{ij}, \mu_j^{(t)}, \sigma_j^{(t)}) f(z_{ij} | \mu_j^{(t)}, \sigma_j^{(t)})}{f(y_i | \mu_j^{(t)}, \sigma_j^{(t)})} \\ &= \frac{\pi_j^{(t)} f(y_i | \mu_j^{(t)}, \sigma_j^{(t)})}{\sum_{l=1}^K \pi_l^{(t)} f(y_i | \mu_l^{(t)}, \sigma_l^{(t)})} \end{aligned} \quad (2.27)$$

dengan  $t$  mengindikasikan banyaknya langkah iterasi. Solusi dari  $\partial L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y}) / \partial \mu_j = 0$  menghasilkan nilai minimum dari  $\mu_j$  pada langkah iterasi ke  $(t+1)$ :

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^N z_{ij}^{(t)} y_i}{\sum_{i=1}^N z_{ij}^{(t)}} \quad (2.28)$$

Selanjutnya, jika fungsi *likelihood* pada persamaan (2.25) diturunkan terhadap  $\sigma_j$ , maka didapatkan:

$$\left[ \sigma_j^2 \right]^{(t+1)} = \frac{\sum_{i=1}^N z_{ij}^{(t)} (y_i - \mu_j^{(t+1)})^2}{\sum_{i=1}^N z_{ij}^{(t)}} \quad (2.29)$$

Kemudian, dapat diturunkan fungsi *likelihood*  $L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y})$  terhadap distribusi prior  $\pi_j$ . Pada tahap ini, domain  $0 \leq \pi_j \leq 1$  harus dipertimbangkan dan  $\sum_{j=1}^K \pi_j = 1$ . Hal ini dapat dicapai dengan menggunakan sebuah *Lagrange multiplier*  $\eta$  dan memaksimumkan turunan berikut.

$$\frac{\partial}{\partial \pi_j} \left[ L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y}) - \eta \left( \sum_{j=1}^K \pi_j - 1 \right) \right] = 0 \quad (2.30)$$

dimana selanjutnya didapatkan:

$$\sum_{i=1}^N \frac{f(y_i | \mu_j, \sigma_j)}{\sum_{l=1}^K \pi_l f(y_i | \mu_l, \sigma_l)} - \eta = 0 \quad (2.31)$$

Jika kedua sisi dikalikan dengan  $\pi_j$  dan digunakan domain  $0 \leq \pi_j \leq 1$  dan  $\sum_{j=1}^K \pi_j = 1$ , maka diperoleh  $\eta = N$ . Dengan demikian, untuk mengeliminasi  $\eta$  dan menyusun kembali turunan terhadap  $\pi_j$  didapatkan:

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^N z_{ij}^{(t)} \quad (2.32)$$

(Nguyen, 2011)

**Algoritma 2.1** Algoritma EM untuk Segmentasi dengan GMM

1. Inialisasi parameter  $\{\mu_j, \sigma_j, \pi_j\}$ , *mean*  $\mu_j$ ; nilai *varians*  $\sigma_j$ ; dan distribusi prior  $\pi_j$ .
2. E-Step : Menghitung nilai  $z_{ij}$  pada persamaan (2.27) menggunakan nilai parameter awal.
3. M-Step : Estimasi ulang parameter  $\{\mu_j, \sigma_j, \pi_j\}$ 
  - a) *Update mean*  $\mu_j$  dengan menggunakan persamaan (2.28).
  - b) *Update varians*  $\sigma_j$  dengan menggunakan persamaan (2.29).
  - c) *Update distribusi prior*  $\pi_j$  dengan menggunakan persamaan (2.32).

Evaluasi fungsi *log-likelihood* pada persamaan (2.25) dan periksa konvergen, baik dari fungsi *log-likelihood* maupun nilai parameter. Jika kriteria konvergen nilai *error*  $< 0,0001$  belum terpenuhi, kembali ke langkah 2.

### 2.6.2 Reversible Jump Markov Chain Monte Carlo (RJMCMC)

Dalam berbagai kasus ditemukan bahwa dimensi vektor parameter model bersifat tidak tetap terutama yang disebabkan oleh model *mixture* dengan jumlah komponen *mixture* yang tidak/belum diketahui banyaknya. RJMCMC pada pemodelan *mixture* dapat digunakan untuk menentukan jumlah komponen *mixture* yang belum diketahui secara pasti. Dalam hal ini berarti bahwa dimensi vektor dari parameter model *mixture* bersifat tidak tetap (tidak diketahui besarnya). Richardson dan Green (1997) serta Wati (2006) memanfaatkan konsep RJMCMC untuk mendapatkan model *mixture* yang optimum dimana banyaknya komponen penyusun *mixture* tidak/belum diketahui. Dalam aplikasinya, Algoritma RJMCMC menggunakan konsep *birth/death* dan *split/merge*. Menurut Richardson dan Green (1997) serta Wati (2006), terdapat enam langkah dalam Algoritma RJMCMC yaitu:

1. *Updating*  $\mu_j$
2. *Updating*  $\sigma_j$
3. *Updating*  $\pi_j$
4. *Updating*  $z_j$
5. *Updating Split/Merge* komponen-komponen
6. *Updating Birth/Death* dari komponen kosong

Langkah 1 sampai dengan Langkah 4 menggunakan algoritma EM yang telah dijelaskan sub-bab sebelumnya, sedangkan langkah 5 dan 6 mengubah banyaknya komponen penyusun *mixture* ke-  $k$  dengan cara satu per satu. Langkah *split/merge* merupakan pilihan acak antara *split* ( $j_1 \rightarrow j_1 + 1$ ) atau *merge* ( $j_1 + 1 \rightarrow j_1$ ) dengan  $j_1 = 1, 2, \dots, k$

Adapun langkah-langkah dalam proses *merge* adalah:

1. Misalkan ada dua grup yang akan bergabung yakni  $j_1$  dan  $j_2$ , maka gabungkan pasangan yang berseberangan  $(j_1, j_2) \rightarrow j^*$
2. Tentukan nilai untuk  $(\pi_{j^*}, \mu_{j^*}, \sigma_{j^*})$  dengan persamaan berikut:

$$\begin{aligned}\pi_{j^*} &= \pi_{j_1} + \pi_{j_2}, \\ \pi_{j^*} \mu_{j^*} &= \pi_{j_1} \mu_{j_1} + \pi_{j_2} \mu_{j_2}, \\ \pi_{j^*} (\mu_{j^*} + \sigma_{j^*}) &= \pi_{j_1} (\mu_{j_1} + \sigma_{j_1}) + \pi_{j_2} (\mu_{j_2} + \sigma_{j_2}).\end{aligned}\tag{2.33}$$

Langkah-langkah proses *split* terdiri atas:

1. Misalkan ada dua grup yang akan memisah yakni  $j_1$  dan  $j_2$ , maka komponen terpilih secara acak dipisah sehingga

$$j^* \rightarrow j_1 + j_2$$

2. Membangun vektor random  $u = (u_1, u_2, u_3)$  dengan  $0 < u_c < 1$   $c = 1, 2, 3$  untuk membentuk parameter baru yang sesuai dengan persamaan berikut:

$$\pi_{j_1} = \pi_{j^*} u_1$$

$$\pi_{j_2} = \pi_{j^*} (1 + u_1)$$

$$\mu_{j_1} = \mu_{j^*} - u_2 \sigma_{j^*} \sqrt{\frac{\pi_{j_2}}{\pi_{j_1}}}$$

$$\mu_{j_2} = \mu_{j^*} - u_2 \sigma_{j^*} \sqrt{\frac{\pi_{j_1}}{\pi_{j_2}}} \quad (2.34)$$

$$\sigma_{j_1}^2 = u_3 (1 - u_2^2) \frac{\sigma_{j^*}^2 \pi_{j^*}}{\pi_{j_1}}$$

$$\sigma_{j_2}^2 = (1 - u_3) (1 - u_2^2) \frac{\sigma_{j^*}^2 \pi_{j^*}}{\pi_{j_2}}$$

3. Ulangi langkah 2 sampai diperoleh kondisi  $\mu_{j_1} < \mu_{j_2}$ .

Adapun langkah-langkah proses *birth/death* merupakan pilihan acak antara *birth* ( $j_1 \rightarrow j_1 + 1$ ) atau *death* ( $j_1 + 1 \rightarrow j_1$ ). Secara lengkap, langkah-langkah dari proses *birth* dan *death* adalah sebagai berikut:

Langkah-langkah dalam proses *birth* adalah:

1. Membentuk parameter  $(\pi_{j^*}, \mu_{j^*}, \sigma_{j^*})$  sebagaimana rumusan pada proses *merge* untuk komponen baru
2. Melakukan penyekalaan kembali untuk parameter pembobot

$$\pi_j \text{ sehingga diperoleh } \sum_{j=1}^2 \pi_j = 1.$$



Langkah-langkah dalam proses *death* terdiri atas:

1. Menghapus komponen kosong dari proses yang dilakukan
2. Melakukan penyekalaan kembali dari  $\pi_j$  yang tersisa.

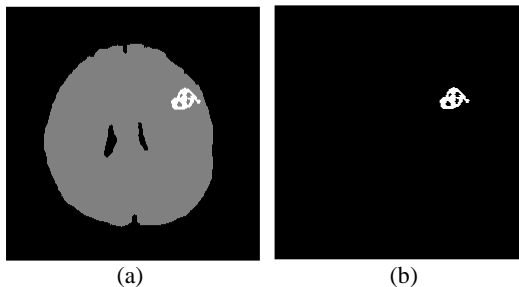
## 2.7 Correct Classification Ratio (CCR)

Dalam mengevaluasi *Region of Interest* (ROI) dari hasil model segmentasi yang dihasilkan, pada penelitian ini akan digunakan ukuran *Correct Classification Ratio* (CCR). CCR merupakan ukuran evaluasi yang digunakan untuk menguji apakah ROI pada hasil segmentasi yang terbentuk telah sesuai dengan *ground truth*-nya. Digunakannya CCR sebagai ukuran evaluasi, karena target (*ground truth*) dalam penelitian ini telah ditetapkan. Ukuran CCR dapat didefinisikan pada persamaan (2.35).

$$CCR = \sum_{j=1}^2 \frac{|GT_j \cap Seg_j|}{|GT|} \quad (2.35)$$

dimana  $GT_j$  adalah *ground truth* untuk  $j=1$  merupakan area non-ROI dan  $j=2$  merupakan area ROI tumor otak, sementara itu  $Seg_j$  dideskripsikan sebagai *pixel* yang disegmentasi terhadap area non-ROI ( $j=1$ ) dan ROI tumor otak ( $j=2$ ) dengan algoritma segmentasi yang digunakan, dan  $GT = \bigcup_{j=1}^2 GT_j$ . Evaluasi citra dilihat dari semakin besar

nilai CCR yang dihasilkan, maka hasil segmentasi citra semakin baik (Nikou, Galatsanos, & Likas, 2007). Dimana citra yang dijadikan sebagai *ground truth* dapat dilihat pada Gambar 2.4.



**Gambar 2.4** *Ground Truth* (a) Segmentasi MRI Tumor Otak (b) ROI Tumor Otak

Penetapan *ground truth* dilakukan secara manual dengan validasi oleh ahli medis. Selanjutnya nilai CCR didapatkan melalui proses perhitungan *confusion matrix* yang merupakan matriks penjabaran letak ketepatan dan kesalahan dari hasil segmentasi. Dari *confusion matrix* tersebut dapat dihitung nilai AUC. Perhitungan ketepatan hasil klasifikasi menggunakan AUC seperti pada Tabel 2.1.

**Tabel 2.1** Tabulasi Silang Ketepatan Klasifikasi

Kelas Aktual Y	Kelas Prediksi Y			Total
	0	1	2	
0	$m_{00}$	$m_{01}$	$m_{02}$	$M_0$
1	$m_{10}$	$m_{11}$	$m_{12}$	$M_1$
2	$m_{20}$	$m_{21}$	$m_{22}$	$M_2$
Total	$M_{.0}$	$M_{.1}$	$M_{.2}$	$M$

Keterangan:

$m_{ij}$  = jumlah pengamatan dari variabel Y kelas  $i$  yang tepat diprediksi sebagai variabel Y kelas  $j$ . ( $i = j$ )

$M_i$  = jumlah pengamatan dari variabel Y kelas  $i$ .

$M_j$  = jumlah prediksi dari variabel Y kelas  $j$ .

$M$  = jumlah total pengamatan atau prediksi.

Berikut merupakan rumus perhitungan AUC pada klasifikasi biner dan AUC pada klasifikasi *multiclass*. (Hand & Till, 2001)

$$\hat{A}(c_i|c_j) = \frac{1}{mn} \sum_{i=1}^m \sum_{j=1}^n \Psi(\rho_i > \rho_j) \quad (2.7)$$

$$\Psi(\rho_i > \rho_j) = \begin{cases} 1, \rho_i > \rho_j \\ \frac{1}{2}, \rho_i = \rho_j \\ 0, \rho_i < \rho_j \end{cases} \quad (2.8)$$

$$AUC(c_i, c_j) = \frac{\hat{A}(c_i|c_j) + \hat{A}(c_j|c_i)}{2} \quad (2.9)$$

$$AUC_{total} = \frac{2}{C(C-1)} \sum_{i < j} AUC(c_i, c_j) \quad (2.10)$$

Keterangan

$\rho_i$  = Peluang suatu observasi dengan kelas positif ke- $k$  diklasifikasikan ke kelas positif

$\rho_j$  = Peluang suatu observasi dengan kelas negatif ke- $l$  diklasifikasikan ke kelas positif

$m$  = Jumlah observasi kelas positif

$n$  = Jumlah observasi kelas negatif.

## BAB III METODOLOGI PENELITIAN

Bab ini berisikan tentang kerangka penelitian, sumber data yang digunakan, variabel penelitian, struktur data, dan langkah analisis yang dilakukan dalam penelitian ini. Langkah analisis dalam penelitian ini kemudian disajikan dalam bentuk diagram alir.

### 3.1 Sumber Data

Sumber Data yang digunakan di dalam penelitian ini merupakan data sekunder, berupa data rekam medis pasien tumor otak di RS X. Data tersebut merupakan hasil *scan* mesin MRI berkekuatan 1,5 Tesla. *Sequence* citra yang akan digunakan, dipilih berdasarkan rekomendasi oleh ahli dari sudut pandang medis, yaitu *ax T1 memp+C slice* ke-18.

### 3.2 Variabel Penelitian

Variabel yang digunakan dalam penelitian ini adalah besarnya intensitas keabuan di tiap-tiap *pixel* citra MRI yang disebut *grayscale*. Skala pengukuran *grayscale* adalah interval dengan skala 0-255.

### 3.3 Struktur Data

Variabel *Grayscale* mewakili intensitas keabuan pada tiap *pixel* citra MRI, dimana intensitas citra MRI dapat direpresentasikan menjadi sebuah matriks berukuran 256 x 256. Selanjutnya dibentuk menjadi sebuah struktur data seperti pada persamaan (2.1).

**Tabel 3.1** Struktur Data

Baris $p$	Kolom $q$	Intensitas <i>Greyscale</i>
0	0	$f(0,0)$
0	1	$f(0,1)$
...	...	...
0	256	$f(0,256)$
1	0	$f(1,0)$
...	...	...
$P$	$q$	$f(p,q)$
...	...	...
255	255	$f(255,255)$

Berdasarkan struktur data tersebut, akan dilakukan analisis *clustering* yang melibatkan tingkat intensitas *grayscale* dan frekuensinya dalam suatu citra.

### 3.4 Langkah Analisis Data

Adapun langkah-langkah yang dilakukan dalam penelitian ini adalah sebagai berikut.

1. *Input* data citra MRI.

Data citra MRI yang diinput dengan ukuran *256-pixel* dengan merubah komposisi *pixel* RGB menjadi *grayscale* menggunakan Persamaan 2.2. Kemudian direpresentasikan menjadi sebuah matriks berukuran  $256 \times 256$ .

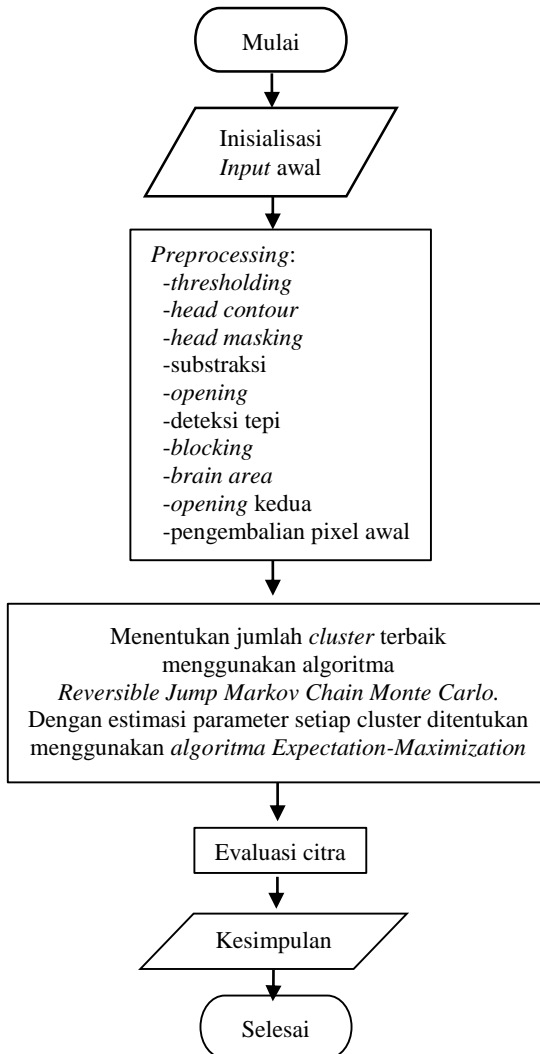
2. *Preprocessing* data citra MRI dengan tahapan sebagai berikut.

- a) Melakukan *thresholding*, di dalam proses ini *pixel* citra yang nilainya kurang dari atau sama dengan nilai *threshold* akan digantikan dengan warna hitam. Sedangkan, *pixel* citra yang nilainya lebih dari *threshold* akan digantikan dengan putih. *Thresholding* yang dilakukan sesuai dengan Persamaan 2.12 dan 2.13.
- b) Menentukan *head contour*, penentuan *head contour* dilakukan dengan cara mencari *pixel-pixel* yang berwarna putih pada setiap baris. *Pixel* yang berwarna putih pertama adalah *head contour*.
- c) Melakukan proses *head masking* dengan cara *blocking* area *head contour*.
- d) Melakukan substraksi antara *head mask* dan *threshold*.
- e) Melakukan proses *morphological opening* untuk menghilangkan *pixel* yang kecil.
- f) Melakukan proses deteksi tepi menggunakan *operator sobel* agar *pixel-pixel* area otak tidak terputus.
- g) Melakukan *blocking* area kepala setelah proses deteksi tepi.
- h) Menentukan area otak dengan cara substraksi.
- i) Melakukan proses *morphological opening* kedua untuk menghilangkan *pixel* yang kecil.

- j) Mengembalikan setiap *pixel* yang bernilai 1 dengan nilai intensitas awalnya.
3. Menentukan jumlah *cluster* optimum menggunakan algoritma *Reversible Jump Markov Chain Monte Carlo*. Dengan estimasi parameter pada setiap *cluster* ditentukan menggunakan algoritma *Expectation-Maximization* untuk GMM dengan Algoritma 2.1.
4. mengevaluasi hasil model segmentasi citra dengan menghitung nilai *Correct Classification Ratio* (CCR).
5. Membuat kesimpulan.

### 3.5 Diagram Alir

Langkah-langkah analisis dalam penelitian ini dapat disajikan dalam Gambar 3.1 diagram alir sebagai berikut.



**Gambar 3.1** Diagram Alir Penelitian

## BAB IV ANALISIS DAN PEMBAHASAN

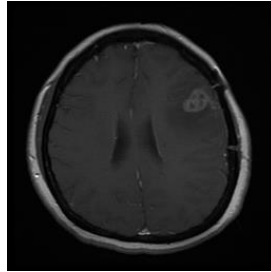
Bab ini berisikan tentang proses pemilihan data citra MRI untuk dilakukan segmentasi, selanjutnya dilakukan tahapan *preprocessing*, implementasi metode *Gaussian Mixture Model* (GMM) dengan menggunakan algoritma *Expectation Maximization* (EM) serta *Reversible Jump Markov Chain Monte Carlo* (RJCMCMC) untuk menentukan area ROI tumor otak, evaluasi hasil segmentasi menggunakan nilai CCR, serta perbandingan metode dalam segmentasi citra MRI tumor otak.

### 4.1 Pemilihan Data dan *Preprocessing*

Tahapan awal yang dilakukan dalam segmentasi citra adalah pemilihan data citra MRI tumor. Namun, sebelum memasuki tahapan segmentasi, dilakukan *preprocessing* terlebih dahulu untuk memperbaiki kualitas citra yang akan digunakan. Citra MRI dipilih berdasarkan *sequence* dan *slice* tertentu yang sebelumnya telah mendapatkan rekomendasi dari sudut pandang medis. Pada penelitian ini, *sequence* citra yang dipilih adalah *sequence ax T1 memp+C*, yang mana *sequence* ini merupakan citra MRI hasil potongan *axial* (atas) otak pasien dengan pemberian *contrast media*. Citra dengan pemberian *contrast* menghasilkan tampilan yang lebih fokus pada tumor otak, sehingga area tumor dan pembengkakannya dapat terlihat lebih jelas. Kemudian, *slice* yang dipilih dalam penelitian ini adalah *slice* ke-18, karena *slice* ini merupakan hasil citra MRI dengan tumor otak yang paling besar (*slice* pusat tumor otak). Dengan demikian, data citra MRI yang akan digunakan dalam penelitian ini yakni, *sequence ax T1 memp+C slice* ke-18.

#### 4.1.1 *Preprocessing*

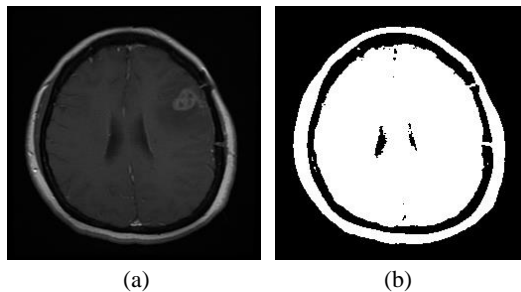
Data citra MRI yang telah dipilih berdasarkan rekomendasi dari ahli medis, selanjutnya akan digunakan dalam tahapan *preprocessing*. Tahapan *preprocessing* ini terbilang cukup penting mengingat proses transfer gambar atau citra MRI tumor yang dilakukan melalui saluran transmisi dapat menyebabkan adanya gangguan pada *pixel-pixel* citra. Berikut adalah data citra MRI yang digunakan dalam penelitian ini direpresentasikan pada Gambar 4.1.



**Gambar 4.1** Citra MRI *Sequence ax T1 memp+C slice* ke-18

Dapat dilihat pada Gambar 4.1 masih berupa citra otak dengan tulang kepala. Tahap *preprocessing* dilakukan agar menghasilkan citra MRI dengan intensitas *greyscale* dan berupa area otak saja, sehingga data lebih mudah dilakukan segmentasi citra untuk memisahkan area tumor otak (ROI) dan bukan area tumor (Non-ROI). *Preprocessing* data citra MRI dilakukan dengan tahapan sebagai berikut.

- 1) Melakukan *thresholding*, dimana dalam proses ini *pixel* citra yang nilainya kurang dari atau sama dengan nilai *threshold* akan diganti dengan warna hitam. Sedangkan, *pixel* citra yang nilainya melebihi nilai *threshold* akan diganti dengan putih. *Thresholding* yang dilakukan sesuai dengan Algoritma 2.2, adapun citra hasil *Thresholding* ditampilkan pada Gambar 4.2.

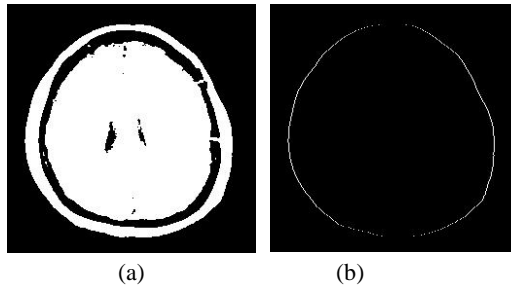


**Gambar 4.2** Citra Tahapan *Thresholding* (a) Citra Masukan (b) Citra Setelah *Thresholding*

Pada Gambar 4.2 yang bagian (a) merupakan citra masukan MRI tumor otak yang berupa citra 2 dimensi dengan komposisi RGB. Sementara itu, Gambar 4.2 bagian (b) merupakan hasil proses *thresholding* yang berupa citra biner.



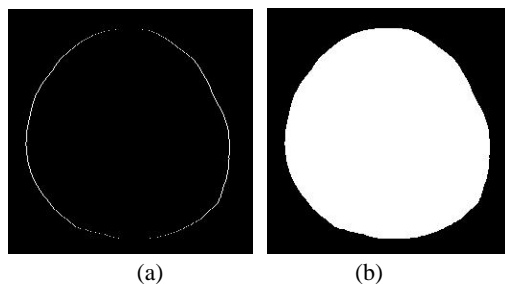
- 2) Menentukan *head contour*, penentuan *head contour* dilakukan dengan cara mencari *pixel* yang berwarna putih pada setiap baris. Selain *pixel* putih pertama dan terakhir akan dibiarkan berwarna hitam. *Pixel* yang berwarna putih pertama adalah *head contour*. Hasil dari proses *head contour* ditampilkan pada Gambar 4.3.



**Gambar 4.3** Citra Tahapan *Head Contour* (a) Citra Hasil *Thresholding* (b) Citra Setelah Penentuan *Head Contour*

Dapat dilihat Gambar 4.3 bagian (a) merupakan citra masukan hasil *thresholding* untuk proses penentuan area *contour* kepala. *Contour* kepala merupakan batas antara citra dengan *background*. Sementara itu, hasil dari proses *head contour* dapat dilihat pada Gambar 4.3 bagian (b), dimana penentuan *contour* kepala dengan cara ini lebih efektif dibandingkan dengan deteksi tepi, karena yang dibutuhkan hanya batas-batas luar kepala saja.

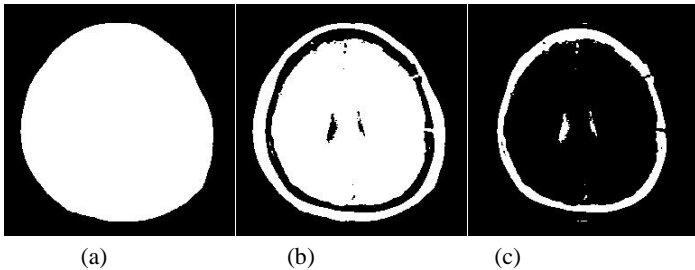
- 3) Melakukan proses *head masking* dengan cara *blocking* area *head contour*. Citra hasil head masking dapat dilihat pada Gambar 4.4.



**Gambar 4.4** Citra Tahapan *Head Masking* (a) Citra Hasil *Head Contour* (b) Citra Setelah dilakukan *Head Masking*

*Pixel* yang berwarna putih pada *head masking* merupakan area kepala, sedangkan *pixel* yang berwarna hitam merupakan area *background*.

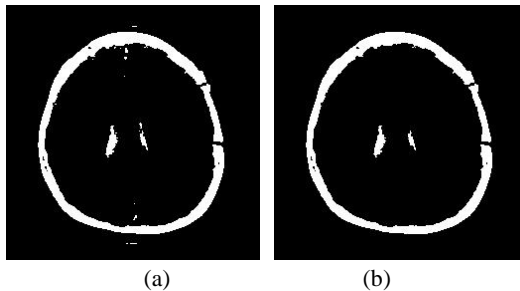
- 4) Melakukan substraksi antara *head mask* dengan *threshold*. Substraksi dilakukan dengan mengurangkan antara citra *head mask* dengan citra *thresholding*. Hasil dari tahap substraksi head mask dengan *thresholding* ditampilkan pada Gambar 4.5.



**Gambar 4.5** Citra Tahapan Substraksi (a) Citra Head Mask (b) Citra Threshold (c) Citra Hasil Substraksi

Pada Gambar 4.5 bagian (a) merupakan citra masukan untuk proses substraksi hasil proses head mask, sementara bagian (b) merupakan citra masukan hasil proses *thresholding*. Hasil proses substraksi yang telah dihilangkan bagian tengkorak kepala dapat dilihat pada Gambar 4.5 bagian (c).

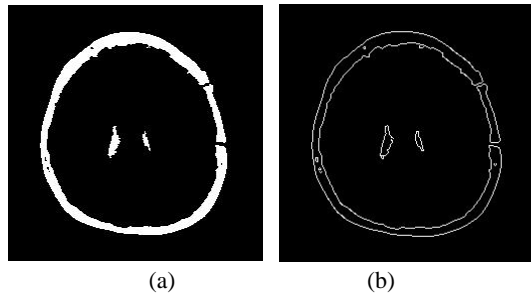
- 5) Melakukan proses *morphological opening* untuk menghilangkan *pixel* yang kecil. Citra hasil opening morfologi dapat dilihat pada Gambar 4.6.



**Gambar 4.6** Citra Tahapan *Morphological Opening* (a) Citra Hasil Substraksi (b) Citra Hasil *Opening*

Pada tahap ini pixel yang berukuran kurang dari 16 dihilangkan dalam pengamatan, sehingga dapat dihasilkan citra yang lebih bersih untuk dilakukan proses selanjutnya.

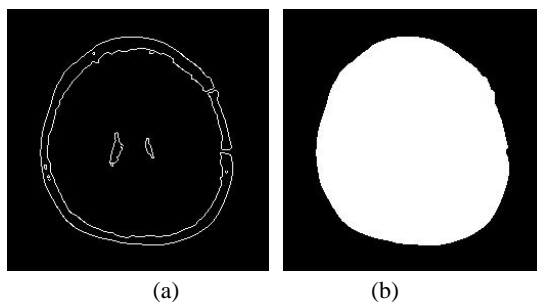
- 6) Melakukan tahapan deteksi tepi (*edge detection*) menggunakan *operator sobel* agar *pixel-pixel* di area otak tidak terputus. Hasil *edge detection* ditampilkan pada Gambar 4.7.



**Gambar 4.7** Citra Tahapan *Edge Detection* (a) Citra Hasil *Opening* (b) Citra Hasil *Edge Detection*

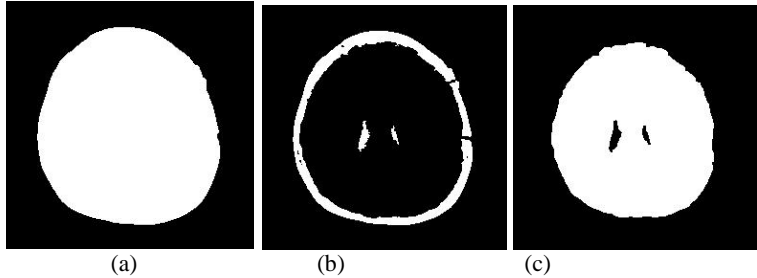
Dapat dilihat pada Gambar 4.7 bagian (a) merupakan citra hasil proses morphological opening sebagai masukan teruntuk proses deteksi tepi, dan Gambar 4.7 bagian (b) merupakan citra hasil proses deteksi tepi.

- 7) Melakukan *blocking area* kepala setelah proses *edge detection*. Citra MRI hasil *blocking* akan digunakan untuk proses subtraksi penentuan area otak. Hasil tahapan *blocking* ditampilkan pada Gambar 4.8.



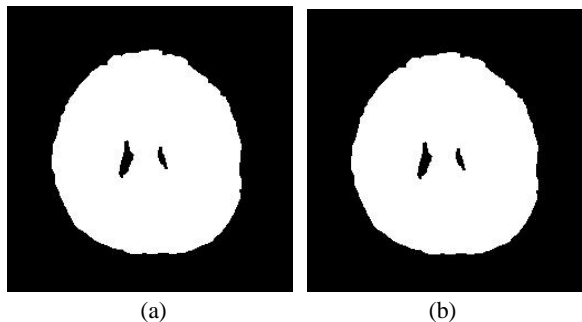
**Gambar 4.8** Citra Tahapan *Blocking* (a) Citra Hasil Penggabungan (b) Citra Hasil *Blocking*

- 8) Menentukan area otak dengan cara substraksi. pada tahapan ini, dilakukan pengurangan citra hasil *blocking* dengan citra hasil penggabungan, sehingga didapatkan citra yang merupakan area otak saja. Citra hasil substraksi pada tahapan ini, dapat dilihat pada Gambar 4.9.



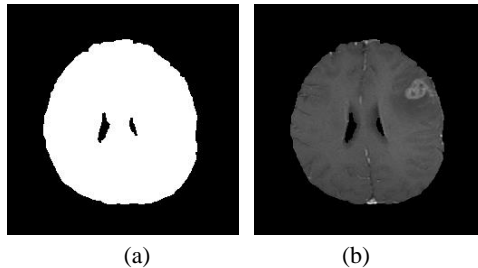
**Gambar 4.9** Citra Tahapan Substraksi Area Otak (a) Citra Hasil *Blocking* (b) Citra Hasil Penggabungan (c) Citra Hasil Substraksi Area Otak

- 9) Melakukan proses *morphological opening* yang kedua kali untuk menghilangkan *pixel-pixel* yang kecil dari hasil substraksi yang telah didapatkan dari sebelumnya. Citra hasil *opening* morfologi dapat dilihat pada Gambar 4.10.



**Gambar 4.10** Citra Tahapan *Morphological Opening* (a) Citra Hasil Substraksi (b) Citra Hasil *Opening*

- 10) Mengembalikan *pixel-pixel* yang mempunyai nilai 1 (berwarna putih) dengan nilai intensitas awalnya, dimana hasil perubahan ditunjukkan pada Gambar 4.11.



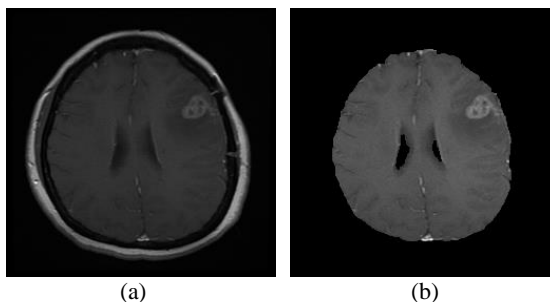
**Gambar 4.11** Citra Akhir *Preprocessing* (a) Citra Hasil *Preprocessing* (b) Citra Area Otak

Data citra MRI yang dihasilkan setelah *preprocessing* adalah berupa sebuah matriks yang memiliki skala data ratio. Data citra ini mempunyai *range* antara 0-255, seperti diilustrasikan pada Lampiran 7.

Tahapan *preprocessing* citra MRI dengan data yang sama sudah pernah dilakukan pada penelitian sebelumnya (Sandra, 2018), namun pada penelitian ini tahapan yang dilakukan agak sedikit berbeda karena dilakukan proses *opening* untuk menghilangkan *pixel-pixel* yang kecil dua kali agar menghilangkan *noise*.

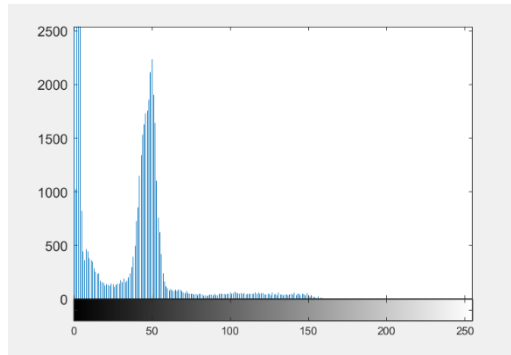
#### 4.1.2 Histogram Citra MRI Tumor Otak

Dalam segmentasi citra MRI, eksplorasi data dilakukan dengan membangun histogram dari matriks citra yang telah terbentuk. Dari histogram inilah selanjutnya dapat dilakukan pendugaan awal bahwa data yang akan digunakan adalah berpola *mixture*. Adapun citra yang digunakan dalam penelitian ini, perbandingan citra awal dan citra hasil *preprocessing* ditampilkan pada Gambar 4.12.



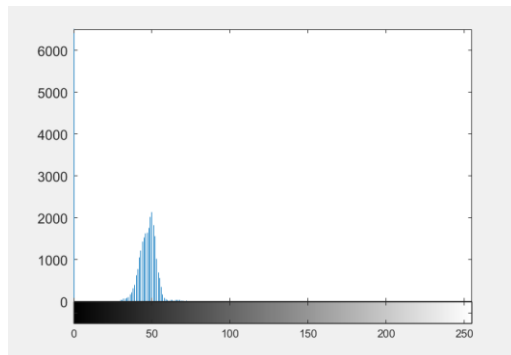
**Gambar 4.12** Citra Hasil *Preprocessing* (a) Citra Awal (b) Citra Setelah *Preprocessing*

Dapat terlihat perubahan citra sebelum dilakukan preprocessing pada Gambar 4.12 (a) dengan citra hasil *preprocessing* merupakan citra dengan komposisi intensitas *greyscale* yang berupa area otak saja pada Gambar 4.12 (b). Selanjutnya histogram citra MRI masukan sebelum *preprocessing* dapat ditampilkan pada Gambar 4.13.



**Gambar 4.13** Histogram Citra Sebelum *Preprocessing*

Pada Gambar 4.13, dapat dilihat bahwa dari histogram citra awal sebelum dilakukan *preprocessing* memiliki 3 puncak. Diduga puncak dengan intensitas tertinggi dalam histogram akan membuat sebuah *cluster* baru, sehingga perlu dihilangkan dahulu dari penelitian dengan *preprocessing*. Selanjutnya, histogram citra hasil *preprocessing* dapat dilihat pada Gambar 4.14.



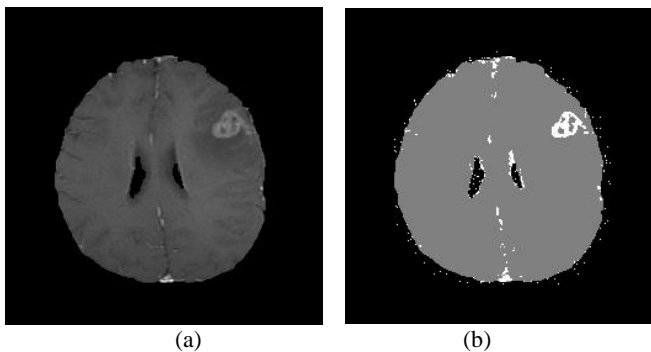
**Gambar 4.84** Histogram Citra Setelah *Preprocessing*

Dapat dilihat bahwa pada Gambar 4.14, terjadi perubahan bentuk histogram pada citra MRI setelah dilakukan *preprocessing*. Setelah dilakukan *preprocessing*, diduga data MRI berpola *mixture* sehingga untuk segmentasi citra MRI digunakan metode *clustering* berbasis *mixture model* dengan GMM. Data inilah yang selanjutnya digunakan untuk memisahkan area ROI tumor otak dengan *noise* menggunakan metode GMM dengan algoritma EM dan RJMCMC.

#### 4.2 Segmentasi Citra dengan Metode *Gaussian Mixture Model* dengan pendekatan *Reversible Jump Markov Chain Monte Carlo* (GMM-RJMCMC)

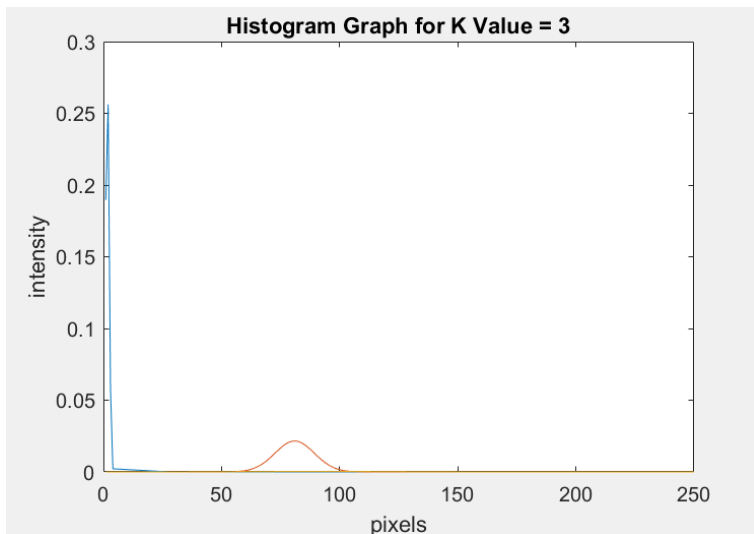
Setelah dilakukan *preprocessing*, selanjutnya dapat dilakukan segmentasi citra MRI untuk menentukan area tumor otak. Segmentasi citra dilakukan dengan *model based clustering* menggunakan metode *Gaussian Mixture Model* (GMM) dengan algoritma EM (*Expectation-Maximization*) untuk mengestimasi parameter model serta *Reversible Jump Markov Chain Monte Carlo* (RJMCMC) untuk otomatisasi dalam penentuan jumlah *cluster* optimum. Data citra yang telah dilakukan *preprocessing* sebelumnya, kemudian dilanjutkan untuk segmentasi dengan masing-masing metode.

Banyaknya *cluster* yang dicoba pada metode RJMCMC sebanyak 5 *cluster*, yakni *cluster* 3 sampai dengan 7 *cluster*. Hasil segmentasi citra dengan metode GMM menggunakan algoritma RJMCMC ditampilkan pada Gambar 4.15.



**Gambar 4.15** Citra Hasil Segmentasi dengan GMM (a) Citra *Preprocessing* (b) Citra Hasil Segmentasi GMM-RJMCMC

Pada Gambar 4.15 bagian (a) dapat dilihat citra masukan untuk segmentasi dengan GMM dari hasil *preprocessing*. Sementara itu, pada Gambar 4.15 (b), dapat dilihat bahwa hasil segmentasi GMM dengan *cluster* optimum adalah 3 *cluster* menghasilkan segmentasi area tumor otak yang diindikasikan dengan warna abu-abu muda. Adapun distribusi data dari hasil segmentasi GMM dengan 3 *cluster* dapat dilihat melalui Gambar 4.16.



**Gambar 4.16** Histogram Hasil Segmentasi Citra menggunakan GMM-RJMCMC

Berdasarkan histogram pada Gambar 4.16 dapat diduga bahwa terdapat 3 puncak untuk data hasil segmentasi. Parameter-parameter GMM dengan 3 komponen distribusi yang diperoleh dengan metode RJMCMC ditampilkan pada Tabel 4.1.

**Tabel 4.1** Estimasi Parameter Model Segmentasi GMM-RJMCMC 3 *Cluster*

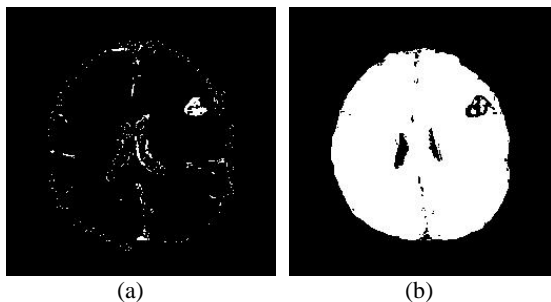
<i>Cluster</i>	$\pi$	$\mu$	$\sigma$
1	0,5299	$2,7410 \times 10^{-95}$	$2,2204 \times 10^{-16}$
2	0,4479	79,9822	66,3078
3	0,0222	83,3117	1305,4

Dan adapun fungsi densitas dari model hasil segmentasi citra dengan menggunakan metode GMM-EM dapat dituliskan pada persamaan (4.1).



$$\begin{aligned}
f(\mathbf{y} | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi}) &= \sum_{j=1}^3 \pi_j f(y | \mu_j, \sigma_j) \\
&= \pi_1 N(y | \mu_1, \sigma_1) + \pi_2 N(y | \mu_2, \sigma_2) + \pi_3 N(y | \mu_3, \sigma_3) \\
&= 0,5299 \frac{1}{\sqrt{2\pi(2,2204 \times 10^{-16})}} \exp\left(-\frac{1}{2} \frac{(y_i - (2,7410 \times 10^{-95}))^2}{2,2204 \times 10^{-16}}\right) \\
&\quad + 0,4479 \frac{1}{\sqrt{2\pi 66,3078}} \exp\left(-\frac{1}{2} \frac{(y_i - 79,9822)^2}{66,3078}\right) \\
&\quad + 0,0222 \frac{1}{\sqrt{2\pi 1305,4}} \exp\left(-\frac{1}{2} \frac{(y_i - 83,3117)^2}{1305,4}\right)
\end{aligned} \tag{4.1}$$

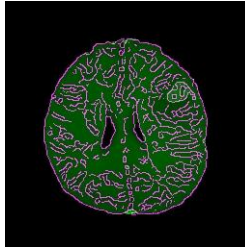
Melalui persamaan (4.1), dapat dilihat ada tiga distribusi normal yang menyusun model segmentasi citra MRI tumor. Dengan bantuan penilaian subjektif dari pihak medis, dapat ditentukan distribusi dari area tumor otak atau yang biasa disebut ROI (*Region of Interest*), yaitu  $N(y | 83,3117; 1305,4)$ . Sementara itu, komponen distribusi lainnya  $N(y | 2,7410 \times 10^{-95}; 2,2204 \times 10^{-16})$  serta  $N(y | 79,9822; 66,3078)$  merupakan bagian otak yang bukan tumor atau Non-ROI (*Non-Region of Interest*). Untuk melihat dengan jelas area ROI dan Non-ROI, hasil segmentasi disajikan pada Gambar 4.17.



**Gambar 4.17** Citra Hasil Segmentasi GMM-RJMCMC (a) Area ROI (b) Area Non-ROI

Pada Gambar 4.17, didapatkan area yang diduga menjadi ROI tumor otak (a) dan area otak yang bukan tumor atau Non-ROI (b). Hasil segmentasi dengan GMM masih menunjukkan bahwa metode GMM tidak *robust* terhadap *noise*, hal ini diindikasikan dari masih terdapat titik-titik kecil yang berada jauh dari area tumor otak aslinya.

Selanjutnya, untuk mendeteksi perubahan intensitas pada area ROI, maka dilakukan *subjective analysis* dengan deteksi tepi. Tampilan *subjective analysis* dapat dilihat pada Gambar 4.18.

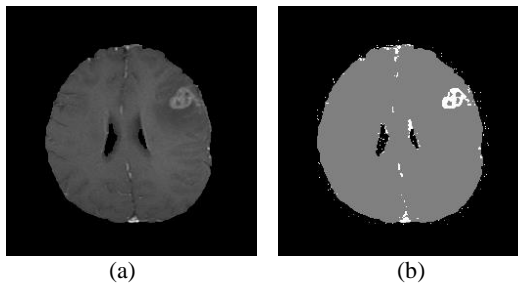


**Gambar 4.18** *Subjective Analysis* GMM-RJMCMC ROI Citra MRI

Berdasarkan Gambar 4.18, dapat terlihat tepi-tepi yang menjadi area ROI pada tumor otak. *Subjective analysis* dilakukan dengan tujuan membantu ahli medis menilai secara subjektif dalam menentukan ROI.

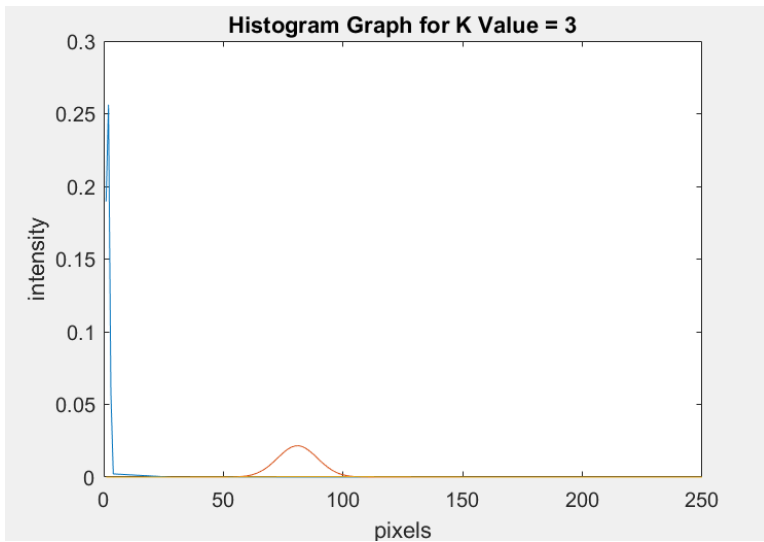
#### 4.3 Segmentasi Citra dengan Metode *Gaussian Mixture Model* dengan algoritma *Expectation Maximization* (GMM-EM)

Selanjutnya segmentasi citra dilakukan dengan *model based clustering* menggunakan metode *Gaussian Mixture Model* (GMM) dengan algoritma *Expectation-Maximization* (EM) untuk mengestimasi parameter model. Hasil segmentasi citra ditampilkan pada Gambar 4.19.



**Gambar 4.19** Citra Hasil Segmentasi dengan GMM (a) Citra *Preprocessing* (b) Citra Hasil Segmentasi GMM-EM

Pada Gambar 4.19 (a) dapat dilihat bahwa citra masukan untuk segmentasi dengan GMM dari hasil *preprocessing*. Sementara itu, pada Gambar 4.19 (b), dapat dilihat bahwa hasil segmentasi GMM dengan *cluster* optimum adalah 3 *cluster* menghasilkan segmentasi area tumor otak yang diindikasikan dengan warna abu-abu muda. Distribusi data dari hasil segmentasi GMM dengan 3 *cluster* dapat dilihat melalui Gambar 4.20.



**Gambar 4.20** Histogram Hasil Segmentasi Citra menggunakan GMM-EM

Melalui histogram pada Gambar 4.20 diduga bahwa terdapat 3 puncak pada data hasil segmentasi. Parameter-parameter GMM dengan 3 komponen distribusi diperoleh dengan metode EM ditampilkan pada Tabel 4.2.

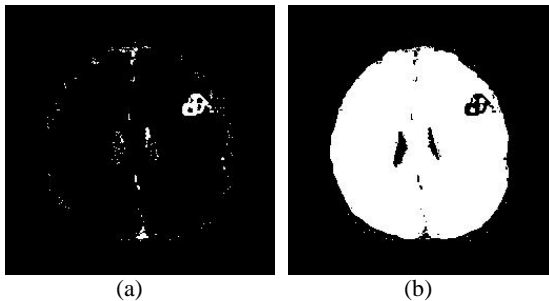
**Tabel 4.2** Estimasi Parameter Model Segmentasi GMM-EM 3 *Cluster*

<i>Cluster</i>	$\pi$	$\mu$	$\sigma$
1	0,5325	0,6684	0,5591
2	0,4487	79,9858	68,2674
3	0,0188	94,4135	769,0575

Dan adapun fungsi densitas dari model hasil segmentasi citra dengan menggunakan metode GMM-EM dapat dituliskan pada persamaan (4.2).

$$\begin{aligned}
f(\mathbf{y} | \boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi}) &= \sum_{j=1}^3 \pi_j f(y | \mu_j, \sigma_j) \\
&= \pi_1 N(y | \mu_1, \sigma_1) + \pi_2 N(y | \mu_2, \sigma_2) + \pi_3 N(y | \mu_3, \sigma_3) \\
&= 0,5325 \frac{1}{\sqrt{2\pi}0,5591} \exp\left(-\frac{1}{2} \frac{(y_i - 0,6684)^2}{0,5591}\right) \\
&\quad + 0,4487 \frac{1}{\sqrt{2\pi}68,2674} \exp\left(-\frac{1}{2} \frac{(y_i - 79,9858)^2}{68,2674}\right) \\
&\quad + 0,0188 \frac{1}{\sqrt{2\pi}769,0575} \exp\left(-\frac{1}{2} \frac{(y_i - 94,4135)^2}{769,0575}\right)
\end{aligned} \tag{4.2}$$

Melalui persamaan (4.2) dapat dilihat bahwa ada tiga distribusi normal yang menyusun model segmentasi citra MRI. Dengan bantuan penilaian subjektif dari pihak medis, dapat ditentukan distribusi dari area tumor otak atau yang biasa disebut ROI (*Region of Interest*), yaitu  $N(y | 94,4135; 769,0575)$ . Sementara itu, kedua komponen distribusi lainnya  $N(y | 0,6684; 0,5591)$  dan  $N(y | 79,9858; 68,2674)$  merupakan bagian otak yang bukan tumor otak atau Non-ROI (*Non-Region of Interest*). Untuk melihat dengan jelas area ROI dan Non-ROI, hasil segmentasi dapat disajikan pada Gambar 4.21.

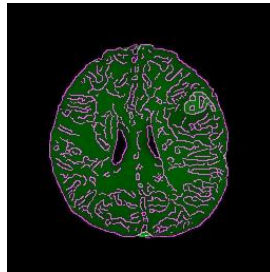


**Gambar 4.21** Citra Hasil Segmentasi GMM-EM (a) Area ROI (b) Area Non-ROI

Pada Gambar 4.21, didapatkan area yang diduga menjadi ROI tumor otak (a) dan area otak yang bukan tumor atau Non-ROI (b). Hasil segmentasi dengan GMM masih menunjukkan bahwa metode GMM

tidak *robust* terhadap *noise*, hal ini diindikasikan dari masih terdapat titik-titik kecil yang berada jauh dari area tumor otaknya.

Selanjutnya, untuk mendeteksi perubahan intensitas pada area ROI dilakukan *subjective analysis* menggunakan deteksi tepi. Tampilan *subjective analysis* dapat dilihat pada Gambar 4.22.



**Gambar 4.22** *Subjective Analysis* GMM-EM ROI Citra MRI

Mengacu pada Gambar 4.22, dapat terlihat bahwa tepi-tepi yang menjadi area ROI tumor otak. *Subjective analysis* dilakukan dengan tujuan membantu ahli medis dalam menentukan area ROI.

#### 4.4 Perbandingan Metode *Gaussian Mixture Model* (GMM) dengan EM dan RJMCMC

Dalam mengevaluasi model segmentasi yang telah dilakukan dengan metode GMM-EM dan GMM-RJMCMC, akan dibandingkan nilai *correct classification ratio* (CCR). Penggunaan CCR sebagai ukuran evaluasi hasil segmentasi ROI tumor otak, karena tumor otak pada pasien yang citra MRI-nya digunakan dalam penelitian ini telah ditangani oleh RS X, sehingga dapat didefinisikan *target/ground truth* yang menjadi ROI tumor otak. Berikut adalah hasil perhitungan *confusion matrix* dari kedua metode, yang kemudian digunakan untuk mendapatkan nilai CCR yang disajikan pada Tabel 4.3 dan Tabel 4.4.

**Tabel 4.3** *Confusion Matrix* untuk Metode GMM-EM

<i>cluster</i>		<i>Predicted</i>		
		1	2	3
<i>Real</i>	1	41139	148	105
	2	64	23545	211
	3	0	51	273

Berdasarkan *confusion matrix* untuk metode GMM-EM yang telah didapatkan pada Tabel 4.3, selanjutnya dapat dihitung nilai AUC yang merupakan representatif dari nilai CCR. Dengan bantuan software R didapatkan nilai AUC sebesar 0,85 untuk GMM-EM. Sedangkan perhitungan untuk metode GMM-RJMCMC adalah sebagai berikut.

**Tabel 4.4** *Confusion Matrix* untuk Metode GMM-RJMCMC

<i>cluster</i>		<i>Predicted</i>		
		1	2	3
<i>Real</i>	1	41053	137	202
	2	59	23492	269
	3	0	51	273

Berdasarkan *confusion matrix* untuk metode GMM-RJMCMC yang telah didapatkan pada Tabel 4.4, selanjutnya dapat dihitung nilai AUC yang merupakan representatif dari nilai CCR. Dengan bantuan software R didapatkan nilai AUC sebesar 0,803 untuk metode GMM-RJMCMC.

Untuk melakukan perbandingan secara lebih jelas secara komputasi antara kedua metode, dapat dilihat pada Tabel 4.5.

**Tabel 4.5** Perbandingan Metode GMM-EM dan GMM-RJMCMC

Metode	<i>Cluster</i>	Banyak Iterasi	<i>Running Time</i>	CCR	Tingkat Kesulitan
GMM- EM	3	100	83,20 s	0,85	Sederhana
GMM-RJMCMC	3-7	100	73,42 s	0,803	Kompleks

Apabila dilihat dari Tabel 4.5, dapat disimpulkan bahwa dengan menggunakan *software* yang sama yakni *Mathlab* R2018b, *hardware* yang sama pula yakni prosesor Intel (R) Core (TM) i7-7500U CPU @ 2.70GHz 2.9 GHz dengan RAM 8.00GB dan jumlah iterasi yang sama, metode GMM-RJMCMC menghasilkan segmentasi ROI tumor otak yang hampir sama namun dengan waktu yang terbilang cukup singkat dibandingkan dengan metode GMM-EM. Hal ini diindikasikan dari nilai evaluasi ROI hasil segmentasi menggunakan CCR yang hampir sama yaitu sebesar 0,85 untuk GMM-EM dan 0,803 untuk GMM-RJMCMC. Perhitungan nilai CCR didapatkan dengan cara membandingkan citra hasil segmentasi dengan citra *ground truth* yang telah

didapatkan dari ahli medis. Semakin banyak *pixel* dari segmentasi yang sesuai dengan *ground truth* maka akan semakin besar nilai CCR. Dari segi waktu *running* juga metode GMM-RJMCMC ini terbilang lebih cepat dan otomatis, tanpa harus melakukan *running* untuk setiap *cluster* seperti algoritma EM. Dengan waktu *running* 73,42 detik untuk *cluster* 3-7, sedangkan untuk EM masih memerlukan waktu *running* 83,20 detik untuk satu *cluster* saja. Dan algoritma EM masih perlu untuk membandingkan setiap *cluster* secara manual sehingga memerlukan waktu yang lebih lama lagi.

*“Halaman ini sengaja dikosongkan”*



## BAB V KESIMPULAN DAN SARAN

Bab ini berisi tentang kesimpulan dari hasil analisis yang telah dilakukan sebelumnya dan saran. Saran yang diberikan ditujukan untuk penelitian selanjutnya.

### 5.1 Kesimpulan

Kesimpulan yang didapatkan dari penelitian ini yaitu sebagai berikut. Setelah dilakukan *preprocessing* pada data citra MRI *sequence ax T1 memp+C* terjadi perubahan bentuk histogram menjadi lebih baik dan hasil citra MRI yang didapatkan sudah tidak mengandung lemak sehingga mempermudah dalam proses segmentasi. Selanjutnya pada saat segmentasi citra MRI, kedua metode menghasilkan jumlah *cluster* optimum yang sama yakni 3 *cluster*. Evaluasi hasil segmentasi melalui nilai CCR menunjukkan bahwa hasil segmentasi citra dengan metode GMM-RJMCMC hampir sama dengan metode GMM-EM. Dan hal ini diindikasikan dengan nilai CCR untuk metode GMM-RJMCMC sebesar 0,803, sedangkan untuk metode GMM-EM sebesar 0,85.

Apabila ditinjau dari segi komputasi, metode GMM-EM lebih mudah diterapkan dibandingkan karena memiliki algoritma yang relatif lebih sederhana dibandingkan GMM-RJMCMC yang cukup kompleks. Namun, metode GMM-EM terbilang kurang praktis dan tidak adaptif karena harus membandingkan setiap *cluster* secara manual agar dapat mengetahui berapa *cluster* optimumnya. Hal ini memakan waktu yang cukup lama dan harus bekerja dua kali untuk menyelesaikan metode ini. Sedangkan metode GMM-RJMCMC dapat secara otomatis menentukan berapa jumlah *cluster* optimum dari hasil segmentasi citra. Sehingga membuat metode GMM-RJMCMC lebih praktis dan adaptif. Apabila dilihat dari waktu yang diperlukan untuk *running* program segmentasi citra MRI untuk mendeteksi tumor otak dengan GMM-EM membutuhkan *running time* selama 83,20 detik untuk satu *cluster* saja, sementara bagi metode GMM-RJMCMC membutuhkan *running time* selama 73,42 detik untuk 5 *cluster* yakni 3, 4, 5, 6 dan 7 *cluster*. Perbedaan ini terbilang cukup jauh karena untuk satu *cluster* GMM-EM saja masih lebih lama 10 detik dibandingkan 5 *cluster* GMM-RJMCMC.

## 5.2 Saran

Berdasarkan analisis yang telah dilakukan pada penelitian ini, adapun beberapa saran untuk penelitian selanjutnya adalah sebagai berikut.

1. Menggunakan model *mixture* yang memiliki bentuk distribusi mengikuti pola distribusi pada citra asli.
2. Menggunakan metode lain yang dapat mengurangi noise pada hasil segmentasi citra.
3. Menganalisis *sequence/slice* lainnya pada MRI otak sehingga dapat memberikan visualisasi tumor yang lebih bervariasi.

## DAFTAR PUSTAKA

- Adipranata, R. (2005). *Kombinasi Metyode Morphological Gradient Dan Transformasi Watershed Pada Proses Segmentasi Citra Digital*. Surabaya: Program Studi Sarjana Departemen Informatika Universitas Kristen Petra.
- Ahmad, U. (2005). *Pengolahan Citra Digital & Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.
- American Brain Tumor Association. (2015). *About Brain Tumors A Primer for Patients and Caregivers*. Chicago: ABTA.
- Astuti, E. Y. (2006). Implementasi Bayesian Markov Chain Monte Carlo pada Pemodelan Portopolio Optimal dengan Pendekatan Model *Mixture* dari Beberapa *Mixture*. Surabaya: Program Magister Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam ITS.
- Astuti, A. B. (2017). *Bayesian mixture model averaging untuk mengidentifikasi perbedaan ekspresi gen percobaan microarray*. Surabaya: Program Doktor Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam ITS.
- Canadian Agency for Drugs and Technologies in Health. (2011). *1.5 Tesla Magnetic Resonance Imaging Scanners Compared with 3.0 Tesla Magnetic Resonance Imaging Scanners: Systematic Review of linical Effectiveness*. Ottawa: CADTH.
- Cook, L. J., & Freedman, J. (2012). *Brain Tumors, Understanding Brain Diseases and Disorders*. New York: The Rosen Publishing Group.
- Edelman, R. E., Hesselink, J. R., Zlatkin, M. B., & Crues, J. V. (1990). *Clinical Magnetic Resonance Imaging*. Toronto: Saunders.
- Gelman, A., Carlin, J. B., Stern, H. S., Dunson, D. B., Vehtari, A., & Rubin, D. B. (2014). *Bayrsian Data Analysis*. Boca Raton: CRC Press.
- Gonzalez, R. C., & Wood, R. E. (2008). *Digital Image Processing, 3rd Edition*. New Jersey: Pearson Prentice Hall.
- Green, P. J. (1995). Reversible Jump Markov Chain Monte Carlo Computation and Bayesian Model Determination. *Biometrika*. 82 (4), 711-732.

- Iriawan, N. (2001). Penaksiran Model Mixture Normal Univariabel: Suatu Pendekatan Metode Bayesian dengan MCMC. *Prosiding Seminar Nasional dan Konferda VII Matematika Wilayah DIY & Jawa Tengah*. Yogyakarta. 4, 105-110.
- Lin, H., & Peng, H. (2008). Machine Recognition for Broad- Leaved Trees Based on Synthetic Features of Leaves Using Probabilistic Neural Network. *2008 International Conference on Computer Science and Software Engineering*. 24, 871-877. Hubei: IEEE.
- McLachlan, G. J. & Basford, K. E. (1988). *Mixture Models Inference and Applications to Clustering*. Marcel Dekker, New York.
- Munir, R. (2004). *Pengolahan Citra Digital*. Bandung: Informatika.
- Munir, R. (2006). Aplikasi Image Thresholding Untuk Segmentasi Objek. *Seminar Nasional Aplikasi Teknologi Informasi 2006* (pp. F-1 - F-6). Yogyakarta: SNATI.
- Nguyen, T. M., & Jonathan Wu, Q. M. (2013). Fast and Robust Spatially Constrained Gaussian Mixture Model for Image Segmentation. *IEEE Trans. Circuits Syst. Video Technol.* 23, 61-635.
- Prasetyo, E. (2011). *Pengolahan Citra Digital dan Aplikasinya Menggunakan MATLAB*. Yogyakarta: Andi.
- Prihartanti, W. (2002). Studi Realibilitas Model Kerusakan Ganda: Pendekatan *Mixture Weibull*. *Tesis*. Surabaya: Program Magister Departemen Statistika Fakultas Matematika, Komputasi, dan Sains Data ITS.
- Richardson, S. & Green, P. J. (1997). On Bayesian Analysis with an Unknown Number of Components. *Journal of the Royal Statistical Society*. B. 59 (4), 731–792.
- Santosa Hospital Bandung Kopo. (2018). *MRI 1.5 Tesla*. Retrieved February 10, 2019, from <http://shbk.santosahospital.com/index.php/en/medical-service/radiodiagnostic-imaging/113-mri-1-5-tesla>
- Shapiro, L., & Stockman, L. (2001). *Computer Vision*. New Jersey: Prentice Hall.

- Shi, H. S. (2005). *Model-Based CLustering*. Ontario: University of Waterloo.
- Sianipar, W. H. (2017). *Komparasi Metode EM-GMM (Expectation Maximization-Gaussian Mixture Model) dan FCM (Fuzzy C-Means) dalam Segmentasi Citra Otak MRI (Magnetic Resonance Imaging) di RSUD Soetomo dalam Menentukan Area Tumor Otak*. Surabaya: Program Sarjana Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam ITS.
- Sonka, M., Hlavac, V., & Boyle, R. (1993). *Image Processing, Analysis, and Machine Vision*. New York: Chapman & Hall Computing - London.
- Tribun Jatim. (2018). *Peringati Hari Tumor Otak Sedunia*. Diakses 20 January 2019, from <http://jatim.tribunnews.com/2018/06/08/peringati-hari-tumor-otak-sedunia-surabaya-neuroscience-club-bagi-takjil-edukasi>
- Wahjoepramono, E. J. (2006). *Tumor Otak*. Tangerang: Fakultas Kedokteran Universitas Pelita Harapan.
- Wati, D. M. (2006). Analisis Bayesian Markov Chain Monte Carlo pada Pemodelan Mixture Normal dengan Banyak Komponen Tidak Diketahui (Studi Kasus: Tingkat Inflasi di Indonesia). Surabaya: Program Magister Departemen Statistika Fakultas Matematika dan Ilmu Pengetahuan Alam ITS.
- Wijaya, T. A., & Prayudi, Y. (2010). Implementasi Visi Komputer dan Segmentasi Citra untuk Klasifikasi Bobot Telur Ayam Ras. *Seminar Nasional Aplikasi Teknologi Informasi 2010* (pp. G-1 - G-5). Yogyakarta: SNATI.

*“Halaman ini sengaja dikosongkan”*

## LAMPIRAN

### Lampiran 1 *Syntax Preprocessing – Thresholding*

```

function [threshold]=Thresholding(input)
threshold=input;
imasize = size(input)
input=input(:);
t_in=mean(input);
input=input(:);
G=input;
L=input;
G(G<t_in)=[];
L(L>t_in)=[];
sum=mean(G)+mean(L)
t_new=sum/2;
while(true)
    old_t=t_new;
    G=input;
    L=input;
    G(G<=old_t)=[];
    L(L>old_t)=[];
    sum=mean(G)+mean(L);
    t_new=sum/2;
if(abs(old_t-t_new)<0.1)
    break;
end
end
for i=1:imasize(1)
for j=1:imasize(2)
    if(threshold(i,j)>=t_new)
        threshold(i,j)=1;
    else
        threshold(i,j)=0;
    end
end
end
threshold=double(threshold);
%imwrite(threshold,'thresholding.jpg','Quality',10
0,'Mode','lossy');
imshow(threshold , []);

```

### Lampiran 2 Syntax Preprocessing – Head Contour dan Head Masking

```
function
[headCont, headMask]=ContourMask(tresholding)
imasize=size(tresholding);
headCont=zeros(imasize(1), imasize(2));
headMask=zeros(imasize(1), imasize(2));
for i=1:imasize
    [r,c]=find(tresholding(i:i, 1:imasize(2)));
    left=min(c);
    right=max(c);
    headCont(i, left)=1;
    headCont(i, right)=1;
    %imshow(headCont , []);

    %imwrite(headCont, 'headContour.jpg', 'Quality', 100,
    'Mode', 'lossy');
    headMask(i:i, left:right)=1;
    %imshow(headMask , []);

    %imwrite(headMask, 'headMask.jpg', 'Quality', 100, 'Mo
    de', 'lossy');
end
    imshow(headCont , []);
    imshow(headMask , []);
end
```

### Lampiran 3 Syntax Preprocessing – Penentuan Area Otak

```
function
[substract, openingIn, eDetection, closing, brain, hole
, opening2, prep]=BrainArea(headMask, tresholding, inp
ut)

substract=headMask-tresholding;
%imwrite(substract, 'substract.jpg', 'Quality', 100, '
Mode', 'lossy');
%figure, imshow(headMask , []);
%figure, imshow(threshold , []);
%figure, imshow(substract , []);
```



### Lampiran 3 Syntax Preprocessing – Penentuan Area Otak (lanjutan)

```

openingIn=bwareaopen(subtract,16);
%imwrite(openingIn,'openingInitial.jpg','Quality',
100,'Mode','lossy');
%figure,imshow(openingIn , []);

eDetection=edge((openingIn),'sobel');
%figure,imshow(eDetection , []);
%imwrite(eDetection,'edgeDetection.jpg','Quality',
100,'Mode','lossy');

se=strel('disk',3);
closing=imclose(eDetection,se);
figure,imshow(closing , []);

imasize=size(openingIn);
combine=openingIn;
for i=1:imasize(1);
    for j=1:imasize(2)
        if(closing(i,j)==1)
            combine(i,j)=1;
        end
    end
end
figure,imshow(combine , []);
hole=imfill(combine,'holes');
figure,imshow(hole , []);
%imwrite(hole,'hole.jpg','Quality',100,'Mode','los
sy');
brain=hole-combine;
figure,imshow(brain , []);
%imwrite(brain,'brainArea.jpg','Quality',100,'Mode
','lossy');
opening2=bwareaopen(brain,1);
figure,imshow(opening2 , []);
%imwrite(opening2,'opening2.jpg','Quality',100,'Mo
de','lossy');
prep=input;
prep(~opening2)=0;

```

### Lampiran 3 Syntax Preprocessing – Penentuan Area Otak (lanjutan)

```
figure, imshow (prep, [])
restore = mat2gray (prep);
%imshow (return);
%I2 = imcrop (return, [75 50 360 400]);
%imshow (I2)
    imwrite (restore, 'outputPrep.jpg', 'Quality', 10
0, 'Mode', 'lossy');
```

### Lampiran 4 Syntax Preprocessing Utama

```
[filename, pathname]=uigetfile({'*.dcm'});
input=dicomread([pathname, filename]);
input=rgb2gray(input)
K=7;
%imshow (input , []);
%preprocessing citra
[threshold]=Thresholding (input);
[headCont, headMask]=ContourMask (threshold);
[substract, openingIn, eDetection, closing, brain, hole
, opening2, prep]=BrainArea (headMask, threshold, input
);
%histogram citra
figure, imhist (input)
figure, imhist (prep)
%output preprocessing
imwrite (input, 'inputIma.jpg', 'Quality', 100, 'Mode',
'lossy');
imwrite (threshold, 'thresholding.jpg', 'Quality', 100
, 'Mode', 'lossy');
imwrite (headCont, 'headContour.jpg', 'Quality', 100, '
Mode', 'lossy');
imwrite (headMask, 'headMask.jpg', 'Quality', 100, 'Mod
e', 'lossy');
imwrite (substract, 'substract.jpg', 'Quality', 100, 'M
ode', 'lossy')
imwrite (openingIn, 'openingInitial.jpg', 'Quality', 1
00, 'Mode', 'lossy');
imwrite (eDetection, 'edgeDetection.jpg', 'Quality', 1
00, 'Mode', 'lossy');
```

**Lampiran 4 Syntax Preprocessing Utama (lanjutan)**

```

imwrite(hole,'hole.jpg','Quality',100,'Mode','lossy');
imwrite(brain,'brainArea.jpg','Quality',100,'Mode','lossy');
imwrite(opening2,'opening2.jpg','Quality',100,'Mode','lossy');
    imwrite(prepmask,'prepmask.jpg','Quality',100,'Mode','lossy')

```

**Lampiran 5 Syntax Segmentasi Metode Gaussian Mixture Model (GMM) dengan Algoritma Expectation-Maximization (EM)**

```

function [mask,mu,v,p]=EMSeg4(ima,k)
%%
%check image
ima=double(imread(ima));
copy=ima;
copy2=ima;           % make a copy
ima=ima(:);         % vectorize ima
mi=min(ima);        % deal with negative
ima=ima-mi+1;       % and zero values
m=max(ima);
s=length(ima);
%%
% histogram citra
h=histogram(ima);
x=find(h);
h=h(x);
x=x(:);h=h(:);
%%
% inisiasi parameter
mu=(1:k)*m/(k+1);
v=ones(1,k)*m;
p=ones(1,k)*1/k;

% mulai proses
sml = mean(diff(x))/1000;
while(1)

```

**Lampiran 5** *Syntax* Segmentasi Metode *Gaussian Mixture Model* (GMM) dengan Algoritma *Expectation-Maximization* (EM) (lanjutan)

```

% Expectation
prb = distribution(mu, v, p, x);
scal = sum(prb, 2) + eps;
loglik = sum(h .* log(scal));
%Maximization
for j=1:k
    pp=h.*prb(:,j)./scal;
    p(j) = sum(pp);
    mu(j) = sum(x.*pp)/p(j);
    vr = (x-mu(j));
    v(j)=sum(vr.*vr.*pp)/p(j)+sml;
%update variance
end
p = p + 1e-3;
p = p/sum(p);

% Exit condition
prb = distribution(mu, v, p, x);
scal = sum(prb, 2) + eps;
nloglik=sum(h .* log(scal));
if ((nloglik-loglik)<0.0001)
    break;
end;
clf
%plot(x,h);
hold on
plot(x,prb)
%plot(x, sum(prb, 2), 'r')
drawnow

end
save nloglik

%%
% menghitung mask
mu=mu+mi-1;
s=size(copy);
mask=zeros(s);

```

**Lampiran 5** *Syntax* Segmentasi Metode *Gaussian Mixture Model* (GMM) dengan Algoritma *Expectation-Maximization* (EM) (lanjutan)

```

for i=1:s(1),
for j=1:s(2),
    for n=1:k
        c(n)=distribution(mu(n),v(n),p(n),copy(i,j));
    end
    a=find(c==max(c));
    mask(i,j)=a(1);
end
end
figure,imshow(mask/255,[]);

%%
%subjective analysis
orgimg=copy;
im_bin=mask;
[r,c]=find(im_bin==3);
rc=[r,c];
for j=1:(numel(rc)/2)
    copy(r(j),c(j))=0;
end
figure, subplot(1,2,1);
imshow(copy,[]), title('NROI-Non region of
interest');
nroi_image=copy;
logimg=imsubtract(orgimg,copy);
roi_image=logimg;
subplot(1,2,2);
imshow(logimg,[]), title('ROI-region of
interest');
%%
[h,f]=size(im_bin==3);
tumour_area=(1/h)*(1/f)*(sum(sum(im_bin)));
display('area of tumour is :');
display(tumour_area);
D.tumour_area=tumour_area;
save tumour_area
%%

```

**Lampiran 5** *Syntax* Segmentasi Metode *Gaussian Mixture Model* (GMM) dengan Algoritma *Expectation-Maximization* (EM) (lanjutan)

```

%%
%pixel calculation
[h,g]=size(im_bin);
number_pixel=(sum(sum(im_bin)));
display('number of pixel in the tumor area :');
display (number_pixel);
D.number_pixel=number_pixel;
save number_pixel
%%
varr1 = edge(logimg, 'canny');
varr2 = imfuse(copy2, varr1);
fuse_image=varr2;
figure, imshow(varr2), title('subjective
analysis');
%%
function y=distribution(m,v,g,x)
%%
x=x(:);
m=m(:);
v=v(:);
g=g(:);
for i=1:size(m,1)
    d = x-m(i);
    amp = g(i)/sqrt(2*pi*v(i));
    y(:,i) = amp*exp(-0.5 * (d.*d)/v(i));
end
%%
function [h]=histogram(datos)
%%
datos=datos(:);
ind=find(isnan(datos)==1);
datos(ind)=0;
ind=find(isinf(datos)==1);
datos(ind)=0;
tam=length(datos);
m=ceil(max(datos))+1;
h=zeros(1,m);

```

**Lampiran 5** *Syntax* Segmentasi Metode *Gaussian Mixture Model* (GMM) dengan Algoritma *Expectation-Maximization* (EM) (lanjutan)

```

for i=1:tam,
    f=floor(datos(i));
    if (f>0 & f<(m-1))
        a2=datos(i)-f;
        a1=1-a2;
        h(f) =h(f) + a1;
        h(f+1)=h(f+1)+ a2;
    end;
end;
h=conv(h, [1, 2, 3, 2, 1]);
h=h(3:(length(h)-2));
h=h/sum(h);

```

**Lampiran 6** *Syntax* Segmentasi Metode *Gaussian Mixture Model* – *Reversible Jump Markov Chain Monte Carlo* (GMM-RJCMCMC)

```

function [mu, junk, covar, p, z, likelihood,
llChangeRatio] = evaluateGaussianMix(X, K,
imageSize, groundTruth, oldState)

%% size data
D = size(X, 1);
N = size(X, 2);

%% Initial Parameter from Old State
mu = oldState.mu;
junk = oldState.junk;
covar = oldState.covar;

p = oldState.p;
z = oldState.z;
% u = oldState.u;

likelihood = oldState.likelihood;
% beta = oldState.beta;
% wDiff2 = oldState.wDiff2;

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC)*  
(lanjutan)

```

%% E-step
    for i = 1:K
        z(:, i) = p(:, i) .* gaussianValue(X, mu(:, i),
            covar(:, :, i));
            likelihood = likelihood + z(:, i);
    end
    likelihood = sum(log(likelihood));
    %logp(X|Pi), il nous faudra aussi +log(p(Pi))
    z = z ./ (sum(z, 2) * ones(1, K));

%% Special -- Compute the current segmentation
score (Rand)
    if exist('groundTruth', 'var') &&
    isempty(groundTruth) == 0
        [junk segmentation] = max(z'); currentRI =
        randIndex(segmentation, groundTruth(:)');
        [junk segmentationCont] = max(p');
        currentRICont = randIndex(segmentationCont,
        groundTruth(:)');
    else
        currentRI = nan; currentRICont = nan;
    end

%% M-step
    for i = 1:K
        if(sum(z(:,i)) > 0) %To catch empty clusters
            mu(:, i) = (X * z(:, i)) / sum(z(:, i));
        end
        newCovar = inv(sum(z(:, i))) * ( ...
            (X - mu(:, i))*ones(1,N))* ...
            sparse(1:N, 1:N, z(:, i), N, N) * ...
            (X - mu(:, i))*ones(1,N))' );
        newCovar = newCovar + eps*eye(D);
        if rcond(newCovar) > 1e3 * eps
            covar(:, :, i) = newCovar;
        end
    end

```



**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC)*  
(lanjutan)

```

clear newCovar;
if(sum(z(:,i)) > 0) %To catch empty
clusters
    pi(:, i) = (z(:, i)) / sum(z(:, i));
end
end
return;

%RJMCMC Step
function [newState, accept] = RJMCMCStep(X, K,
imageSize, groundTruth, oldState)
    newState = oldState;

    %Evaluate joint log posterior using Gaussian
Mix
    [mu, junk, covar, p, z, likelihood,
llChangeRatio] = evaluateGaussianMix(X, K,
imageSize, groundTruth, oldState);

    %Compute log acceptance probability
    alpha = likelihood - oldState.likelihood +
llChangeRatio;

    %Accept draw with probability alpha
    val = rand;
    val2 = exp(min(alpha,0));

    if val <= val2
        % Accept the candidate
        accept = 1;
    else
        %reject the candidate
        accept = 0; % Note the
rejection
    end;

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

newState.K = K;

newState.mu = mu;
newState.junk = junk;
newState.covar = covar;

newState.p = p;
newState.z = z;

newState.likelihood = likelihood;
newState.accept = accept;
newState.alpha = alpha;
end

%DoRJMCMC
function [states, AR] = doRJMCMC(X, K, imageSize,
groundTruth, draws, awal)

%% initial parameter
D = size(X, 1);
N = size(X, 2);

%% initial state
[mu p junk covar] = initParameter(X, K);
p = p'*ones(1, N);
p = (p ./ (ones(K, 1) * (sum(p, 1)+eps)))';
z = inv(N) * ones(N, K);

    newState.K = K;
    newState.mu = mu;
    newState.junk = junk;
    newState.covar = covar;
    newState.p = p;
    newState.z = z;
    newState.likelihood = awal;
    newState.accept = 0;
    newState.alpha = 0;

    states(1) = newState;

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

%% Draw and evaluate RJMCMC step
    accepted = 0;

    w = waitbar(0, ['Draw for (K = ' num2str(K)
    ') - ' num2str(0) ' to ' num2str(draws)]);
    waitbar(1/draws, w, ['Draw for (K = '
    num2str(K) ') - ' num2str(1) ' to '
    num2str(draws)]);
    WinOnTop(w);

    for cntrDraws = 2:draws
        [state, accept] = RJMCMCStep(X, K,
    imageSize, groundTruth, states(cntrDraws - 1));

        old = states(cntrDraws-1);
        new = state;

        accepted = accepted + accept;
        states(cntrDraws) = state;

        waitbar(cntrDraws/draws, w, ['Draw
    for (K = ' num2str(K) ') - ' num2str(cntrDraws) '
    to ' num2str(draws)]);
        WinOnTop(w);

        disp('K = ')
        disp(state.K)
        disp('likelihood = ');
        disp(state.likelihood)
        disp('alpha = ');
        disp(state.alpha)
        disp('accept = ');
        disp(state.accept)

    end;

    waitbar(1, w, ['Draw for (K = ' num2str(K)
    ') - ' num2str(cntrDraws) ' to ' num2str(draws)]);
    WinOnTop(w);

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

%Acceptance Rate
    AR = accepted / cntrDraws;
    info = ['Acceptance Rate (K = ' num2str(K)
') = ' num2str(AR)];

    %msgbox(info);
    close(w);
end

function [centroids weights idx
covars]=initParameter(ima,k)

%%
%check image
row=256;
col=256;
copy=reshape(ima,row,col);
copy2=reshape(ima,row,col);
ima=reshape(ima,row*col,1); % vectorize ima
mi=min(ima); % deal with negative
ima=ima-mi+1; % and zero values
m=max(ima);
s=length(ima);

%%
% histogram citra
h=histogram(ima);
x=find(h);
h=h(x);
x=x(:);h=h(:);

%%
% insiasi parameter

centroids=(1:k)*m/(k+1);
v=ones(1,k)*m;
weights=ones(1,k)*1/k;

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

% mulai proses

sml = mean(diff(x))/1000;
while(1)

    % Expectation

    prb = distribution(centroids,v,weights,x);
    scal = sum(prb,2)+eps;
    loglik=sum(h.*log(scal));

    %Maximization

    for j=1:k
        pp=h.*prb(:,j)./scal;
        weights(j) = sum(pp);
        centroids(j) =
sum(x.*pp)/weights(j);
        vr = (x-centroids(j));

v(j)=sum(vr.*vr.*pp)/weights(j)+sml; %update
variance

        end
    for j=1:k
        covars(:, :, j)=v(j);
    end
    weights = weights + 1e-3;
    weights = weights/sum(weights);

    % Exit condition
    prb = distribution(centroids,v,weights,x);
    scal = sum(prb,2)+eps;
    nloglik=sum(h.*log(scal));
    if((nloglik-loglik)<0.0001)
        break;
    end;
end;

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

fig = figure(k);
clf(fig);
set(gca, 'FontSize', 8);
set(gca, 'xtick', [], 'ytick', [], 'xcolor', 'w', 'ycolor', 'w');

set(fig, 'Name', 'Histogram');
set(fig, 'NumberTitle', 'off');
set(fig, 'ToolBar', 'none');
set(fig, 'MenuBar', 'none');
set(fig, 'Position', [0 0 500 350]);

WinOnTop(fig);

plot(x, prb)
title(['Histogram Graph for K Value = '
num2str(k)]);
xlabel('pixels');
ylabel('intensity');
%plot(x, sum(prb, 2), 'r')
drawnow

end
save nloglik
%%
% menghitung mask
centroids=centroids+mi-1;
s=size(copy);
idx=zeros(s);

for i=1:s(1),
for j=1:s(2),
    for n=1:k
c(n)=distribution(centroids(n),v(n),weights(n),copy(i,j));
        end
        a=find(c==max(c));
        idx(i,j)=a(1);
    end
end
end

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC)*  
(lanjutan)

```

% MainProcess
function [ final_img, nroi_image, roi_image, varr2
] = MainProcess( img_path_ori, gt_path, iter,
handles )
%CLASSIFYPROCESS Summary of this function goes
here
clc;
addpath(fullfile(pwd, 'lib'));
addpath(fullfile(pwd, 'rjmcmc'));

%% read a test image and a ground truth file
B=imread(img_path_ori);

[row,col]=size(B);
nsize=row*col;
Y=reshape(B,1,nsize);
Y=double(Y);

gt=importdata(gt_path);
%[mu, covar, p, z, u] = gaussianMixDCASV2(Y, k,
size(B), gt, iter);

%% RJMCMC Process using Different K Value
k_values = [3,4,5,6,7];

val = -999999999999999;
z = B;
k = k_values(1);
accept = 0;

awal = 0;
for i = 1:size(k_values,2)
    K = k_values(i);
    [states, AR] = doRJMCMC(Y, K, size(B), gt,
iter, awal);

    for j = 1:size(states,2)
        state = states(j);
        if(state.accept == 1)

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

        %awal = state.likelihood;
        if(state.alpha > val)
            val = state.alpha;
            select_state = state;
        end
    end
end

end
end

z = select_state.z;
z = z';
k = select_state.K;
lk = select_state.likelihood;
accept = select_state.accept;

%%
ima=double(B);
copy=ima;
copy2=ima;           % make a copy
ima=ima(:);          % vectorize ima
mi=min(ima);         % deal with negative
ima=ima-mi+1;        % and zero values
m=max(ima);
s=length(ima);

%% calculating mask
[ final_img, mask ] = masking( z, row, col, k );

axes(handles.axOutput);
cla(handles.axOutput);
final_img = final_img/255;
imshow(final_img);
drawnow;

```



**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

%% subjective analysis
orgimg=copy;
im_bin=mask;
[r,c]=find(im_bin==2);
rc=[r,c];
for j=1:(numel(rc)/2)
    copy(r(j),c(j))=0;
end

axes(handles.axOutput3);
cla(handles.axOutput3);
%subplot(1,2,1);
imshow(copy);
drawnow;

nroi_image=copy;
logimg=imsubtract(orgimg,copy);
roi_image=logimg;

axes(handles.axOutput2);
cla(handles.axOutput2);
%subplot(1,2,2);
imshow(logimg);
drawnow;

display('selected K value : ');
display(num2str(k));
display('selected Likelihood value : ');
display(num2str(lk));

%%
[h,g]=size(im_bin==2);
tumour_area=(1/h)*(1/g)*(sum(sum(im_bin)));
display('area of tumour is :');
display(tumour_area);
D.tumour_area=tumour_area;
save tumour_area;

```

**Lampiran 6** *Syntax Segmentasi Metode Gaussian Mixture Model – Reversible Jump Markov Chain Monte Carlo (GMM-RJMCMC) (lanjutan)*

```

%%
%pixel calculation
[h,g]=size(im_bin);
number_pixel=(sum(sum(im_bin)));
display('number of pixel in the tumor area :');
display (number_pixel);
%display(nsize);
D.number_pixel=number_pixel;
save number_pixel;

%%
varr1 = edge(logimg,'canny');
varr2 = imfuse(copy2,varr1);
fuse_image=varr2;
axes(handles.axOutput4);
cla(handles.axOutput4);
imshow(varr2);

info = ['Selected K Value = ' num2str(k) char(10)
'Selected Likelihood Value = ' num2str(lk)
char(10) 'Tumor Area Detected !'];

display(info);
msgbox(['Process Finished -- '
info],'Information','help');
end

```

**Lampiran 7** *Syntax Perhitungan AUC dengan software R*

```

data=read.csv('E:/datafix.csv')
rjmcmmc=data$rjmcmmc
em=data$em
y=data$gt

#AUC
multiclass.roc(rjmcmmc,y)
multiclass.roc(em,y)

```

**Lampiran 8** *Output* Perhitungan AUC dengan *software* R

```
> data=read.csv('E:/datafix.csv')
> rjmcmc=data$rjmcmc
> em=data$em
> y=data$gt
>
> #AUC
> multiclass.roc(rjmcmc,y)
Setting direction: controls < cases
Setting direction: controls < cases
Setting direction: controls < cases

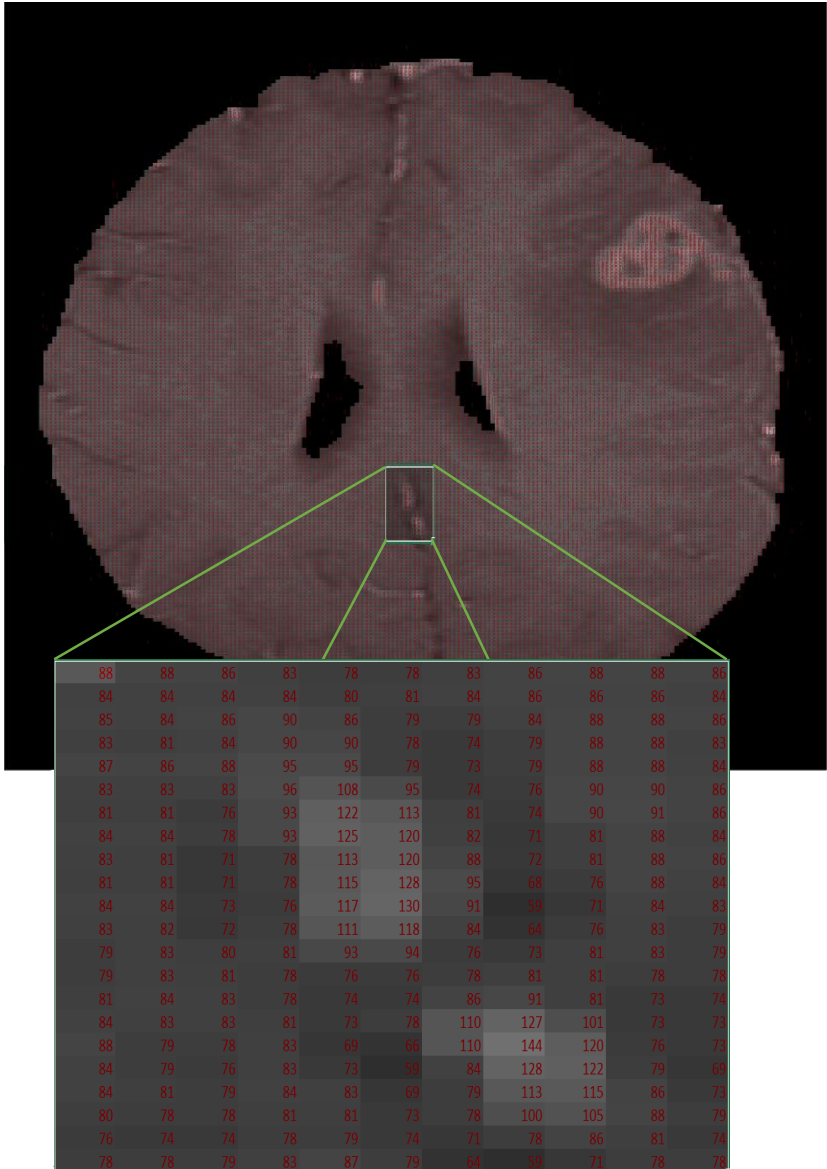
Call:
multiclass.roc.default(response = rjmcmc, predictor = y)

Data: y with 3 levels of rjmcmc: 1, 2, 3.
Multi-class area under the curve: 0.803
> multiclass.roc(em,y)
Setting direction: controls < cases
Setting direction: controls < cases
Setting direction: controls < cases

Call:
multiclass.roc.default(response = em, predictor = y)

Data: y with 3 levels of em: 1, 2, 3.
Multi-class area under the curve: 0.85
```

## Lampiran 9 Gambar Nilai Numerik Matriks *Grayscale* Citra MRI Data Inputan



**Lampiran 10 Surat Pernyataan Pengambilan Data Tugas Akhir****SURAT PERNYATAAN**

Saya yang bertanda tangan di bawah ini, mahasiswa Program Sarjana Departemen Statistika FMKSD ITS.

Nama : Yusuf Puji Hermanto

NRP : 0621154000095

menyatakan bahwa data yang digunakan dalam Tugas Akhir ini, merupakan data sekunder bagian penelitian Disertasi mahasiswa Program Doktor Departemen Statistika FMKSD-ITS yaitu:

Judul : *Spatially Constrained Neo-Normal Mixture Model* dengan Pendekatan Bayesian pada Segmentasi Citra MRI Tumor Otak

Oleh : Anindya Apriliyanti Pravitasari

NRP : 1315301003

Surat pernyataan ini dibuat dengan sebenarnya. Apabila terdapat pemalsuan data maka mahasiswa bersedia menerima sanksi sesuai aturan yang berlaku.

Mengetahui,  
Pembimbing Tugas Akhir



Prof. Drs. Nur Iriawan, Mkom., Ph.D  
NIP. 19621015 198803 1 002

Surabaya, 26 Mei 2019



Yusuf Puji Hermanto  
NRP. 0621154000095

*“Halaman ini sengaja dikosongkan”*

## BIODATA PENULIS



Penulis penelitian ini bernama Yusuf Puji Hermanto dilahirkan di Kota Pasuruan pada tanggal 24 Agustus 1996 sebagai anak kedua dari dua bersaudara. Ayah Pujiono dan Ibu Asia Wati. Penulis menempuh pendidikan formal di SDN 2 Tawangrejo, SMP Negeri 2 Pandaan, dan Madrasah Bertaraf Internasional (MBI) Amanatul Ummah. Kemudian penulis diterima sebagai Mahasiswa jurusan Statistika FMKSD ITS melalui jalur SBMPTN pada tahun 2015 dengan NRP 0621154000095. Selama masa perkuliahan, penulis memiliki pengalaman dalam berpartisipasi di berbagai kepanitiaan diantaranya sebagai Wakil Ketua *Data Analysis Competition* (DAC) dalam *event* tahunan PRS (Pekan Raya Statistika) 2017, yang merupakan kompetisi analisis data tingkat Asia Tenggara. Penulis pernah terlibat beberapa organisasi mahasiswa seperti organisasi tingkat jurusan dan tingkat fakultas yakni HIMASTA-ITS dan juga BEM FMKSD. Selama menjabat di HIMASTA-ITS penulis menjadi staf Departemen PSDM (Pengembangan Sumber Daya Mahasiswa) yang kemudian ilmu tersebut diimplementasikan pada tahun selanjutnya sebagai Ketua Departemen PSDM BEM FMKSD masa jabatan tahun 2018. Penulis juga pernah mengikuti pelatihan Pengembangan diri yaitu LKMM Pra-TD dan LKMM TD.. Apabila pembaca ingin memberi kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini, dapat menghubungi penulis melalui email [yusufph.yh@gmail.com](mailto:yusufph.yh@gmail.com).