



TUGAS AKHIR - SS141501

**SEGMENTASI CITRA MRI TUMOR OTAK MENGGUNAKAN
GAUSSIAN MIXTURE MODEL DAN HYBRID GAUSSIAN
MIXTURE MODEL - SPATIALLY VARIANT FINITE
MIXTURE MODEL DENGAN ALGORITMA
EXPECTATION-MAXIMIZATION**

**SANDRA FIRDA QONITA
NRP 062114 4000 0119**

**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 2018**



TUGAS AKHIR - SS141501

**SEGMENTASI CITRA MRI TUMOR OTAK MENGGUNAKAN
GAUSSIAN MIXTURE MODEL DAN HYBRID GAUSSIAN
MIXTURE MODEL - SPATIALLY VARIANT FINITE
MIXTURE MODEL DENGAN ALGORITMA
EXPECTATION-MAXIMIZATION**

**SANDRA FIRDA QONITA
NRP 062114 4000 0119**

**Dosen Pembimbing
Prof. Drs. Nur Iriawan, MIKom., Ph.D.**

**PROGRAM STUDI SARJANA
DEPARTEMEN STATISTIKA
FAKULTAS MATEMATIKA, KOMPUTASI, DAN SAINS DATA
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**

“Halaman ini sengaja dikosongkan”



FINAL PROJECT - SS141501

**MRI-BASED BRAIN TUMOR SEGMENTATION USING
GAUSSIAN MIXTURE MODEL AND HYBRID GAUSSIAN
MIXTURE MODEL - SPATIALLY VARIANT FINITE MIXTURE
MODEL WITH EXPECTATION-MAXIMIZATION ALGORITHM**

**SANDRA FIRDA QONITA
SN 062114 4000 0119**

**Supervisor
Prof. Drs. Nur Iriawan, MIKom., Ph.D.**

**UNDERGRADUATE PROGRAMME
DEPARTMENT OF STATISTICS
FACULTY OF MATHEMATICS, COMPUTING, AND DATA SCIENCE
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA 2018**

“Halaman ini sengaja dikosongkan”

LEMBAR PENGESAHAN

SEGMENTASI CITRA MRI TUMOR OTAK MENGGUNAKAN *GAUSSIAN MIXTURE MODEL DAN HYBRID GAUSSIAN MIXTURE MODEL – SPATIALLY VARIANT FINITE MIXTURE MODEL DENGAN ALGORITMA EXPECTATION- MAXIMIZATION*

TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat
Memperoleh Gelar Sarjana Sains
pada

Program Studi Sarjana Departemen Statistika
Fakultas Matematika, Komputasi, dan Sains Data
Institut Teknologi Sepuluh Nopember

Oleh:

Sandra Firda Qonita
NRP. 062114 4000 0119

Disetujui Oleh Pembimbing:

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

()



Mengetahui,
Kepala Departemen

Dr. Suhartono
NIP. 19710929 199512 1 001

SURABAYA, JULI 2018

**SEGMENTASI CITRA MRI TUMOR OTAK MENGGUNAKAN
GAUSSIAN MIXTURE MODEL DAN HYBRID GAUSSIAN
MIXTURE MODEL – SPATIALLY VARIANT FINITE MIXTURE
MODEL DENGAN ALGORITMA EXPECTATION-
MAXIMIZATION**

Nama : Sandra Firda Qonita
NRP : 062114 4000 0119
Departemen : Statistika
Dosen Pembimbing : Prof. Drs. Nur Iriawan, MIKom., Ph.D.

Abstrak

Tumor otak merupakan salah satu bagian dari tumor pada sistem saraf. Berbagai penelitian telah dilakukan untuk membantu tenaga medis dalam menangani tumor otak, salah satunya dengan melakukan pendeteksian tumor otak melalui segmentasi citra medis berdasarkan MRI. Pada kasus citra MRI, segmentasi dilakukan untuk memisahkan Region of Interest (ROI) atau segmen yang dianggap penting dalam sudut pandang medis, dengan segmen-segmen lainnya (Non-ROI) termasuk noise. Metode segmentasi citra yang umum digunakan adalah model based clustering dengan Gaussian Mixture Model (GMM). Namun, kelemahan GMM adalah antar pixel pada citra dianggap independen sehingga hasil segmentasi tidak memiliki ketahanan terhadap noise dalam segmentasi citra. Untuk mengurangi efek negatif dari noise, dalam penelitian ini akan digunakan model Markov Random Field (MRF) yang secara penuh mempertimbangkan dependensi spasial antara pixel dan proporsi probabilitas label secara eksplisit akan dimodelkan sebagai vektor probabilitas. Sehingga metode yang digunakan adalah Gaussian Mixture Model (GMM) dan GMM yang dibatasi secara spasial oleh Markov Random Fields, atau yang diberi nama Spatially Variant Finite Mixture Model (SVFMM), dimana inisial parameter didapatkan dari GMM, sehingga model yang diajukan adalah hybrid GMM-SVFMM. Dalam proses inferensi, metode estimasi maximum likelihood digunakan untuk mengestimasi parameter model yang diusulkan menggunakan algoritma Expectation-Maximization (EM). Hasil penelitian menunjukkan bahwa segmentasi citra MRI tumor otak dengan hybrid GMM-SVFMM mampu memberikan hasil yang lebih akurat untuk memisahkan ROI dengan noise, dibandingkan jika menggunakan metode GMM.

Kata Kunci: *EM, GMM, Markov Random Fields, Segmentasi Citra MRI, SVFMM*

“Halaman ini sengaja dikosongkan”

MRI-BASED BRAIN TUMOR SEGMENTATION USING GAUSSIAN MIXTURE MODEL AND HYBRID GAUSSIAN MIXTURE MODEL – SPATIALLY VARIANT FINITE MIXTURE MODEL WITH EXPECTATION-MAXIMIZATION ALGORITHM

Name : Sandra Firda Qonita
Student Number : 062114 4000 0119
Department : Statistics
Supervisor : Prof. Drs. Nur Iriawan, MIKom., Ph.D.

Abstract

A brain tumor is one part of the tumor in the nervous system. Various studies have been conducted to assist medical personnel in dealing with brain tumors, one of them is by performing brain tumor detection through image-based medical segmentation of MRI. In the case of MRI, segmentation is performed to separate the Region of Interest (ROI) or segments that are considered important in the medical point of view, with other segments (Non-ROI) including noise. The commonly used image segmentation method is the model-based clustering with Gaussian Mixture Model (GMM). However, the weakness of GMM is that between the pixels in the image are considered independent, so that the segmentation results do not have the noise robustness in image segmentation. To minimize the negative effects of the noise, in this research we will use the Markov Random Field (MRF) model which fully takes into account the spatial dependencies between pixels. The proportion of label of pixels probabilities will be explicitly modeled as probability vectors. At the same time, pixel component functions are also relatively related to neighboring pixels. This scenario could be implemented as the GMM that is spatially limited by MRF, called the Spatially Variant Finite Mixture Model (SVFMM), in which the initial parameter generated from the GMM, so the proposed model is hybrid GMM-SVFMM.. In the inference process, the maximum likelihood estimation method is used to estimate the proposed model parameters using the Expectation-Maximization (EM) algorithm. The results from the correct classification ratio (CCR) showed that MRI-based brain image segmentation couple with hybrid GMM-SVFMM was able to provide more accurate results to separate the ROI with noise compared to GMM.

Keywords: EM, GMM, MR Image Segmentation, Markov Random Fields, SVFMM

“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 MENGGUNAKAN GAUSSIAN MIXTURE MODEL DAN HYBRID GAUSSIAN MIXTURE MODEL – SPATIALLY VARIANT FINITE MIXTURE MODEL DENGAN ALGORITMA EXPECTATION-MAXIMIZATION”**

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. Ayah Wahjudi Swandaru, Ibu Ulfa Atin, adik-adik, 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 Dr. Irhamah, S.Si., M.Si. selaku dosen penguji atas saran dan kritiknya yang sangat membangun.
5. Ibu Dr. Dra. Ismaini Zain, M.Si. 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. Widiani 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 Siti Aisyah, Kiki N. Aisyah, L. Kareninna Padaga, Nadia Insyira, Nikita D. Septiana, Izzan Rasyadi, dan Ria

Retna yang senantiasa memberikan waktu dan dukungan yang diberikan dalam menyelesaikan Tugas Akhir ini.

9. Teman-teman seperjuangan Tugas Akhir Segmentasi Citra, Mbak IImi, Mbak Azizah, dan Mas Ardi yang selalu memberikan waktunya untuk diskusi selama penyelesaian Tugas Akhir ini.
10. Teman-teman seperjuangan PW 118 serta Respect 2014 Sigma 25, yang telah memberikan rasa kekeluargaan selama 4 tahun penulis menempuh studi di ITS.
11. Sahabat-sahabat yang setia mendengar keluh kesah penulis dan memberi semangat tanpa henti Dinda Alfatan S. Z., Aldora Sindila F., Elsa Alfianti, Ira Rachmawati, Yunia Desy A., dan Anggita Adeliyani.
12. 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 2018

Penulis

DAFTAR ISI

	Halaman
HALAMAN JUDUL	i
TITLE PAGE	iii
LEMBAR PENGESAHAN	v
ABSTRAK	vii
ABSTRACT	ix
KATA PENGANTAR	xi
DAFTAR ISI	xiii
DAFTAR GAMBAR	xv
DAFTAR TABEL	xvii
DAFTAR LAMPIRAN	xix
DAFTAR NOTASI	xxi
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	5
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	6
1.5 Batasan Masalah	6
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Magnetic Resonance Imaging</i> (MRI).....	7
2.2 Segmentasi Citra	8
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.....	15
2.4 <i>Model Based Clustering</i>	16
2.5 <i>Gaussian Mixture Model</i> (GMM).....	18
2.6 <i>Markov Random Fields</i>	19
2.7 <i>Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model</i> (GMM-SVFMM).....	22
2.8 <i>Expectation-Maximization</i> (EM).....	24
2.8.1 Algoritma <i>Expectation-Maximization</i> (EM) untuk GMM.....	24

2.8.2	Algoritma <i>Expectation-Maximization</i> (EM) untuk <i>Hybrid GMM-SVFMM</i>	27
2.9	<i>Correct Classification Ratio</i> (CCR).....	30
BAB III	METODOLOGI PENELITIAN	33
3.1	Sumber Data.....	33
3.2	Ruang Implementasi	33
3.3	Variabel Penelitian dan Struktur Data.....	33
3.4	Langkah-langkah Analisis Data	34
3.5	Diagram Alir	36
BAB IV	IMPLEMENTASI, UJI COBA, ANALISIS DAN PEMBAHASAN	37
4.1	Pemilihan Data dan <i>Preprocessing</i>	37
4.1.1	Data Masukan <i>Preprocessing</i>	37
4.1.2	Histogram Citra MRI Tumor Otak.....	43
4.2	Uji Coba, Pemilihan, dan Validasi <i>Cluster</i>	45
4.2.1	Validasi Jumlah <i>Cluster</i> untuk Metode <i>Gaussian Mixture Model</i> (GMM)	45
4.2.2	Validasi Jumlah <i>Cluster</i> untuk Metode <i>Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model</i> (GMM-SVFMM)	46
4.3	Segmentasi Citra MRI dengan <i>Model Based Clustering</i>	47
4.3.1	Segmentasi Citra dengan Metode <i>Gaussian Mixture Model</i> (GMM).....	47
4.3.2	Segmentasi Citra dengan Metode <i>Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model</i> (GMM-SVFMM)	50
4.4	Perbandingan Metode <i>Gaussian Mixture Model</i> (GMM) dan <i>Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model</i> (GMM-SVFMM).....	55
BAB V	KESIMPULAN DAN SARAN	57
5.1	Kesimpulan	57
5.2	Saran	58
DAFTAR PUSTAKA		
LAMPIRAN		
BIODATA PENULIS		

DAFTAR GAMBAR

	Halaman
Gambar 2.1	Perubahan Nilai <i>Pixel</i> Citra RGB ke dalam Citra Abu-Abu 10
Gambar 2.2	Histogram Citra..... 11
Gambar 2.3	Citra Hasil <i>Thresholding</i> 13
Gambar 2.4	Model grafis probabilistik: (a) <i>Mixture model</i> ; (b) <i>Mixture model</i> dengan dependensi spasial..... 20
Gambar 2.5	Sistem orde pertama <i>neighbourhood pixel</i> 21
Gambar 2.6	<i>Ground Truth</i> (a) Segmentasi MRI Tumor Otak (b) ROI Tumor Otak..... 30
Gambar 3.1	Diagram Alir Penelitian 36
Gambar 4.1	Citra MRI <i>Sequence ax T1 memp+C slice</i> ke-18 38
Gambar 4.2	Citra Tahapan <i>Thresholding</i> (a) Citra Masukan (b) Citra Setelah <i>Thresholding</i> 38
Gambar 4.3	Citra Tahapan <i>Head Contour</i> (a) Citra Hasil <i>Thresholding</i> (b) Citra Setelah Penentuan <i>Head Contour</i> 39
Gambar 4.4	Citra Tahapan <i>Head Masking</i> (a) Citra Hasil <i>Head Contour</i> (b) Citra Setelah dilakukan <i>Head Masking</i> .. 39
Gambar 4.5	Citra Tahapan Substraksi (a) Citra Head Mask (b) Citra Threshold (c) Citra Hasil Substraksi 40
Gambar 4.6	Citra Tahapan <i>Morphological Opening</i> (a) Citra Hasil Substraksi (b) Citra Hasil <i>Opening</i> 40
Gambar 4.7	Citra Tahapan <i>Edge Detection</i> (a) Citra Hasil <i>Opening</i> (b) Citra Hasil <i>Edge Detection</i> 41
Gambar 4.8	Citra Tahapan Penggabungan (a) Citra Hasil <i>Opening</i> (b) Citra Hasil <i>Edge Detection</i> (c) Citra Hasil Penggabungan..... 41
Gambar 4.9	Citra Tahapan <i>Blocking</i> (a) Citra Hasil Penggabungan (b) Citra Hasil <i>Blocking</i> 42
Gambar 4.10	Citra Tahapan Substraksi Area Otak (a) Citra Hasil <i>Blocking</i> (b) Citra Hasil Penggabungan (c) Citra Hasil Substraksi Area Otak 42
Gambar 4.11	Citra Akhir <i>Preprocessing</i> (a) Citra Hasil <i>Preprocessing</i> (b) Citra Area Otak 43

Gambar 4.12	Citra Hasil <i>Preprocessing</i> (a) Citra Awal (b) Citra Setelah <i>Preprocessing</i>	44
Gambar 4.13	Histogram Citra Sebelum <i>Preprocessing</i>	44
Gambar 4.14	Histogram Citra Setelah <i>Preprocessing</i>	45
Gambar 4.15	Perbandingan Nilai <i>Correct Classification Ratio</i> (CCR) Metode GMM.....	46
Gambar 4.16	Perbandingan Nilai <i>Correct Classification Ratio</i> (CCR) Metode <i>Hybrid</i> GMM-SVFMM.....	47
Gambar 4.17	Citra Hasil Segmentasi dengan GMM (a) Citra <i>Preprocessing</i> (b) Citra Hasil Segmentasi GMM 3 <i>Cluster</i>	48
Gambar 4.18	Histogram Hasil Segmentasi Citra menggunakan GMM-3 <i>cluster</i>	48
Gambar 4.19	Citra Hasil Segmentasi GMM-3 <i>Cluster</i> (a) Area ROI (b) Area Non-ROI.....	50
Gambar 4.20	<i>Subjective Analysis</i> GMM ROI Citra MRI Sequence ax T1 memp+C.....	50
Gambar 4.21	Citra Hasil Segmentasi dengan <i>Hybrid</i> GMM-SVFMM (a) Citra <i>Preprocessing</i> (b) Citra Hasil Segmentasi <i>Hybrid</i> GMM-SVFMM 5 <i>Cluster</i>	51
Gambar 4.22	Histogram Hasil Segmentasi Citra menggunakan <i>Hybrid</i> GMM-SVFMM-5 <i>cluster</i>	52
Gambar 4.23	Citra Hasil Segmentasi <i>Hybrid</i> GMM-SVFMM-5 <i>Cluster</i> (a) Area ROI (b) Area Non-ROI.....	54
Gambar 4.24	<i>Subjective Analysis</i> <i>Hybrid</i> GMM-SVFMM ROI Citra MRI Sequence ax T1 memp+C.....	54

DAFTAR TABEL

	Halaman
Tabel 3.1 Spesifikasi Ruang Implementasi	33
Tabel 3.2 Variabel Penelitian	33
Tabel 3.3 Struktur Data	34
Tabel 4.1 Estimasi Parameter Model Segmentasi <i>Gaussian Mixture Model 5 Cluster</i>	49
Tabel 4.2 Estimasi Parameter Model Segmentasi <i>Hybrid GMM-SVFMM 5 Cluster</i>	52
Tabel 4.3 Perbandingan Metode GMM dan <i>Hybrid GMM-SVFMM</i> dengan Algoritma EM.....	55

“Halaman ini sengaja dikosongkan”

DAFTAR LAMPIRAN

	Halaman
Lampiran 1	<i>Syntax Preprocessing – Thresholding</i>63
Lampiran 2	<i>Syntax Preprocessing – Head Contour dan Head Masking</i>64
Lampiran 3	<i>Syntax Preprocessing – Penentuan Area Otak</i>64
Lampiran 4	<i>Syntax Preprocessing Utama</i>66
Lampiran 5	<i>Syntax Segmentasi Metode Gaussian Mixture Model (GMM) dengan Algoritma Expectation-Maximization (EM)</i>67
Lampiran 6	<i>Syntax Segmentasi Metode Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM) dengan Algoritma Expectation-Maximization (EM)</i>71
Lampiran 7	<i>Syntax Function MEX Conjugate Projection C++</i>81
Lampiran 8	Hasil Segmentasi Citra <i>Sequence ax T1 memp+C</i> dengan Metode GMM.....86
Lampiran 9	Hasil Segmentasi Citra <i>Sequence ax T1 memp+C</i> dengan Metode <i>Hybrid GMM-SVFMM</i>88
Lampiran 10	Nilai <i>Correct Classification Ratio (CCR)</i> Metode GMM dan <i>Hybrid GMM-SVFMM</i>91
Lampiran 11	Surat Pernyataan Pengambilan Data Sekunder92

“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_0	Nilai <i>threshold</i> pertama
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 dan Hybrid GMM-SVFMM

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
θ_j	Vektor parameter distribusi <i>mixture</i> ke- j
π	Vektor parameter proporsi
μ	Vektor parameter <i>mean</i>
σ	Vektor parameter varians
π_j	Parameter proporsi komponen ke- j

μ_j	Parameter <i>mean</i> komponen ke- j
σ_j	Parameter varians komponen ke- j
π_{ij}	Parameter proporsi data ke- i komponen ke- j
Σ_j	Parameter kovarians komponen ke- j
z_{ij}	Variabel laten data ke- i komponen ke- j
Z	<i>Normalizing constant</i>
V_{C_i}	Fungsi <i>clique</i> potensial
C	Himpunan <i>clique</i>
$d_{i,m}$	Jarak antar dua vektor label $\{\vec{\pi}_i\}$ dan $\{\vec{\pi}_m\}$
β	Nilai skalar
η	<i>Lagrange multiplier</i>

Correct Classification Ratio (CCR)

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

BAB I

PENDAHULUAN

Bab ini membahas mengenai latar belakang yang mendasari dilakukannya penelitian ini. Selanjutnya merumuskan masalah dan menetapkan tujuan penelitian. Selain itu, pada bab ini juga dibahas mengenai manfaat dan batasan masalah dari penelitian yang dilakukan.

1.1 Latar Belakang

Tumor otak merupakan salah satu bagian dari tumor pada sistem saraf, di samping tumor spinal dan tumor saraf perifer. Tumor otak dapat berupa tumor yang sifatnya primer ataupun yang merupakan metastasis dari tumor pada organ lainnya (Wahjoepramono, 2006). Menurut *The Central Brain Tumor Registry of the United States* (CBTRUS), tumor otak primer termasuk dalam 10 besar penyebab kematian terkait kanker. Diperkirakan sekitar 13.000 orang di Amerika Serikat meninggal dunia akibat tumor ini setiap tahunnya. Tumor otak meliputi sekitar 85-90% dari seluruh kanker susunan saraf pusat. Di Amerika Serikat insiden tumor otak ganas dan jinak adalah 21,42 per 100.000 penduduk per tahun (7,25 per 100.000 penduduk untuk tumor otak ganas, 14,17 per 100.000 penduduk per tahun untuk tumor otak jinak). Sementara itu, angka insiden untuk tumor otak ganas di seluruh dunia berdasarkan angka standar populasi dunia adalah 3,4 per 100.000 penduduk. Tumor otak merupakan penyebab kematian kedua pada kasus kanker yang terjadi pada anak-anak yang berusia di bawah 20 tahun. Tumor otak juga merupakan penyebab kematian yang kedua dari semua kasus kanker yang terjadi pada pria berusia 20-39 tahun. Selain itu tumor otak merupakan penyebab kematian nomor lima dari seluruh pasien kanker pada wanita yang berusia 20-39 tahun. Saat ini, diperkirakan 700.000 orang di Amerika Serikat hidup dengan tumor otak primer, dan lebih dari 79.000 lainnya akan didiagnosis pada tahun 2018 (ABTA, 2017). Tumor otak dapat mematikan, dan secara signifikan mempengaruhi kualitas hidup pasien dan orang terdekatnya.

Banyak penelitian telah dilakukan untuk membantu tenaga medis dalam menangani tumor otak, salah satunya dengan melakukan pendeteksian tumor otak melalui segmentasi citra medis berdasarkan MRI. *Magnetic Resonance Imaging* (MRI) adalah teknik pencitraan

medis tingkat lanjut yang memberikan informasi mengenai anatomi jaringan lunak otak manusia. Beberapa faktor kelebihan yang dimiliki MRI, terutama kemampuannya membuat potongan koronal, sagital, aksial dan oblik tanpa banyak memanipulasi posisi tubuh pasien sehingga sangat sesuai untuk diagnostik jaringan lunak. Teknik penggambaran MRI relatif kompleks karena gambaran yang dihasilkan tergantung pada banyak parameter. Bila pemilihan parameter tersebut tepat, kualitas gambar MRI dapat memberikan gambaran detail tubuh manusia dengan perbedaan yang kontras, sehingga anatomi dan patologi jaringan tubuh dapat dievaluasi secara teliti (Notosiswoyo & Suswati, 2004).

Di sisi lain, Jumlah data yang tersedia dari MRI terlalu banyak untuk dilakukan analisis/interpretasi secara manual dan hal ini merupakan salah satu hambatan terbesar dalam efisiensi MRI. Untuk alasan ini, teknik analisis komputasi secara otomatis atau semi otomatis menjadi sangat penting dalam menganalisis data MRI. Dalam praktiknya, MRI telah digunakan di Rumah Sakit Umum Daerah (RSUD) Dr. Soetomo di Surabaya. Alat radiologi yang digunakan di RSUD Dr. Soetomo merupakan MRI berkekuatan 1,5 Tesla, dimana saat ini MRI telah berkembang dan terdapat yang lebih baru yaitu MRI berkekuatan 3 Tesla. Alat MRI berkekuatan 3 Tesla tersebut memberikan hasil citra yang lebih baik daripada MRI 1,5 Tesla, namun MRI 3 Tesla belum digunakan di RSUD Dr. Soetomo dan hanya beberapa Rumah Sakit swasta di Surabaya yang telah menggunakannya.

Segmentasi citra medis merupakan langkah penting dalam analisis citra. Proses memartisi citra digital menjadi beberapa daerah atau kumpulan *pixel* disebut segmentasi citra. Pada citra MRI segmentasi citra dilakukan untuk memisahkan *Region of Interest* (ROI) berupa segmen yang dianggap penting secara medis, dengan segmen-segmen lainnya (Non-ROI). Pada prinsipnya, pemisahan dilakukan terhadap objek yang berbeda pada gambar, yang memiliki tekstur atau warna yang sama. Hasil segmentasi citra adalah kumpulan daerah yang secara kolektif menutupi keseluruhan gambar atau satu set kontur yang diambil dari gambar (Bouchet, Pastore, & Ballariv, 2007). Semua *pixel* di wilayah tertentu akan memiliki kesamaan ciri khas atau properti yang dapat dihitung, seperti warna, intensitas atau tekstur. Algoritma

numerik yang efisien dapat membantu peneliti, dokter dan ahli bedah saraf untuk menyelidiki dan mendiagnosis struktur dan fungsi otak, bila sehat atau terkena tumor. Berdasarkan hal tersebut, ada kebutuhan untuk menerapkan teknik segmentasi yang kuat agar dapat dideteksi gambar yang dipengaruhi tumor dan jenis gambar yang berbeda pada berbagai struktur anatomi (Middleton & Damper, 2004). Untuk mendapatkan hasil yang dapat diterima, peneliti tidak dapat hanya mengandalkan data yang diamati, namun harus memanfaatkan semua informasi awal (*prior information*) yang ada mengenai gambar atau kelas gambar yang sedang dianalisis.

Banyak model berdasarkan berbagai ilmu statistika untuk segmentasi citra telah diajukan di tahun-tahun sebelumnya. *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 citra MRI, model *mixture* sangat berguna dalam praktik. *Finite Mixture Model* (FMM) adalah salah satu model statistik yang paling banyak diterapkan karena bisa memodelkan fenomena yang sangat rumit. Dalam beberapa tahun terakhir, FMM menerima perhatian lebih dan lebih banyak dilakukan penelitian menggunakan model ini karena kesederhanaan dan fleksibilitasnya. Fungsi komponen FMM itu sendiri dapat berupa distribusi probabilistik. FMM telah berhasil diterapkan ke berbagai bidang, seperti genetika, ilmu sosial, astronomi, *data mining*, segmentasi citra (McLachlan & Peel, 2000).

Gaussian Mixture Model (GMM) merupakan suatu FMM ketika distribusi *Gaussian* dipilih sebagai fungsi komponennya. Pada segmentasi citra, sebuah *pixel* diproduksi oleh GMM, kemudian parameter GMM dapat diperoleh sesuai dengan citranya (Bishop, 2006). GMM dapat memperoleh hasil segmentasi yang baik untuk suatu citra tanpa *noise*, namun biasanya akan memperoleh hasil segmentasi yang tidak memuaskan jika terdapat *noise* dalam citra. Alasan utama yang mengarah pada hasil segmentasi yang tidak memuaskan dengan *noise* citra untuk GMM adalah mengasumsikan bahwa hubungan antara *pixel* dalam satu gambar bersifat independen secara statistik. Hubungan

(dependensi) spasial antara *pixel* tidak dipertimbangkan dalam GMM saat diterapkan pada segmentasi citra. Pada penelitian sebelumnya oleh Sanjay dan Hebert (1988), diterapkan suatu metode untuk meningkatkan ketahanan (*robust*) GMM terhadap *noise*, yaitu dengan menggunakan model *Markov Random Fields* (MRF) untuk mempertimbangkan dependensi spasial *pixel* (Sanjay & Hebert, 1988).

Model *Markov Random Fields* (MRF) merupakan alat yang relatif sederhana namun efektif untuk mencakup pengetahuan sebelumnya (*prior knowledge*) dalam proses segmentasi. Dengan memodelkan citra sebagai MRF, dengan asumsi bahwa setiap *pixel* yang diberikan dependen secara statistik pada *pixel* lainnya hanya melalui sekelompok ketetanggaan (*neighbor*) yang dipilih, peneliti dapat menghindari suatu kesulitan untuk menetapkan *prior* yang berarti, dan hanya boleh menentukan karakteristik gambar setempat. Dan yang lebih penting, dependensi spasial dapat dengan mudah diekspresikan melalui definisi fungsi potensial yang sesuai dalam distribusi *Gibbs*.

Penelitian mengenai implementasi GMM dalam segmentasi citra MRI telah dilakukan oleh Sianipar (2017), yang menghasilkan kesimpulan bahwa metode EM-GMM lebih *robust* terhadap *Salt and Pepper Noise* sementara FCM lebih *robust* terhadap *Gaussian Noise* (Sianipar, 2017). Meskipun demikian, model segmentasi citra MRI yang dihasilkan dengan GMM masih belum mampu mengakomodasi *robust* terhadap *noise*. Oleh karena itu, untuk mengatasi *robust* terhadap *noise* pada GMM, pada penelitian ini akan diterapkan model *Markov Random Fields* (MRF), sesuai dengan penelitian sebelumnya oleh Sanjay dan Hebert (1988) atau metode tersebut diberi nama *Spatially Variant Finite Mixture Model* (SVFMM) (Sanjay & Hebert, 1988). Pada tahun 2007, Nikou, Galatsanos, dan Likas telah melakukan penelitian mengenai segmentasi citra menggunakan model SVFMM pada beberapa gambar kehidupan nyata (Nikou, Galatsanos, & Likas, 2007). Selain itu, Nguyen juga telah melakukan penelitian mengenai konsep *Gaussian Mixture Model* berdasarkan informasi spasial untuk segmentasi citra (Nguyen, 2011).

Berdasarkan pemaparan di atas dalam penelitian ini akan dilakukan segmentasi citra MRI menggunakan metode *Gaussian Mixture Model* (GMM) dan metode GMM yang dibatasi secara spasial

dengan *Markov Random Fields* (MRF) dan inisial parameter berasal dari GMM atau diberi nama *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-SVFMM), dimana sebelumnya akan dilakukan *preprocessing* terlebih dahulu pada citra MRI, yaitu melakukan *skull stripping* dan *image enhancement* dengan *thresholding*, deteksi tepi, dan operasi morfologi. Kemudian akan dilakukan pemodelan segmentasi dengan GMM dan *hybrid* GMM-SVFMM. Selanjutnya akan dihitung estimasi parameter menggunakan algoritma *Expectation-Maximization* (EM).

. Citra yang telah *diinput* akan diubah menjadi sebuah matriks, dimana tiap-tiap elemen dari matriks tersebut berupa nilai-nilai intensitas keabuan (*grayscale*). Lalu, setelah ditentukan banyaknya (validasi) *cluster* dengan menggunakan nilai *Correct Classification Ratio* (CCR) dan penilaian subjektif dari ahli medis, kemudian setelah dipilih jumlah *cluster*, kedua hasil segmentasi algoritma tersebut akan dibandingkan dan dievaluasi dengan nilai CCR-nya. Dalam melakukan penelitian ini, akan dibuat program menggunakan *Integrated Development Environment* (IDE) Matlab.

1.2 Rumusan Masalah

Rumusan masalah yang digunakan sebagai acuan untuk pada penelitian ini adalah: berdasarkan citra MRI yang direpresentasikan menjadi sebuah matriks yang memiliki intensitas *grayscale*, bagaimana segmentasi citra MRI tumor otak menggunakan *Gaussian Mixture Model* (GMM) dan GMM yang dibatasi secara spasial dengan *Markov Random Fields* (MRF) atau diberi nama *Spatially Variant Finite Mixture Model* (SVFMM), dimana inisial parameter didapatkan dari GMM sehingga model yang digunakan adalah *Hybrid* GMM-SVFMM dengan metode estimasi maksimum *likelihood* melalui algoritma *Expectation-Maximization* (EM).

1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang telah diuraikan, maka tujuan yang ingin dicapai dalam penelitian ini adalah sebagai berikut.

1. Mengetahui implementasi model segmentasi citra menggunakan metode *Gaussian Mixture Model* (GMM) dan *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-

SVFMM) untuk mendeteksi tumor otak berdasarkan *sequence* MRI yang diamati.

2. Membandingkan hasil segmentasi citra MRI dengan *Gaussian Mixture Model* (GMM) dan *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-SVFMM) menggunakan algoritma *Expectation-Maximization* (EM).

1.4 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah memaksimalkan kualitas dan mereduksi *noise* dalam segmentasi citra, terutama pada citra medis/MRI.

1.5 Batasan Masalah

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

1. Lingkungan implementasi menggunakan IDE Matlab.
2. Data yang digunakan adalah satu *sequence* citra MRI otak untuk pasien kasus tumor otak, yang merupakan data sekunder dari RSUD Dr. Soetomo, Surabaya.
3. Mesin MRI yang digunakan adalah MRI 1,5 Tesla.
4. Penentuan jumlah *cluster* terbaik berdasarkan nilai validasi *correct classification ratio* (CCR) dan mempertimbangkan penilaian subjektif dari ahli medis.
5. *Output* yang dihasilkan adalah berupa citra yang sudah tersegmentasi berdasarkan model GMM dan *hybrid* GMM-SVFMM dengan algoritma *Expectation-Maximization* (EM) dan dapat ditentukan *Region of Interest* (ROI) dan *Non-Region Interest* (Non-ROI).

BAB II

TINJAUAN PUSTAKA

Bab ini membahas mengenai landasan teori yang digunakan untuk membantu dalam menyelesaikan permasalahan pada penelitian ini yaitu segmentasi citra MRI menggunakan metode Gaussian Mixture Model (GMM) dan Spatially Variant Finite Mixture Model (SVFMM) dengan algoritma Expectation-Maximization (EM) untuk mendeteksi area tumor otak. Beberapa landasan teori yang dibahas pada bab ini adalah *magnetic resonance imaging* (MRI), segmentasi citra, *preprocessing*, *model-based clustering*, GMM, *markov random fields*, *Hybrid GMM-SVFMM*, algoritma *Expectation-Maximization* (EM), dan evaluasi ROI tumor otak hasil segmentasi dengan *correct classification ratio* (CCR).

2.1 *Magnetic Resonance Imaging* (MRI)

Magnetic Resonance Imaging (MRI) adalah suatu alat kedokteran di bidang pemeriksaan diagnostik radiologi, yang menghasilkan rekaman gambar potongan penampang tubuh/organ manusia dengan menggunakan medan magnet berkekuatan antara 0,064 – 1,5 tesla (1 tesla = 1000 Gauss) dan resonansi getaran terhadap inti atom hidrogen (Stark, 1988).

Beberapa faktor kelebihan yang dimiliki MRI, terutama pada kemampuannya membuat potongan koronal, sagital, aksial dan oblik tanpa banyak memanipulasi posisi tubuh pasien sehingga sangat sesuai untuk diagnostik jaringan lunak. Teknik penggambaran MRI relatif kompleks karena gambaran yang dihasilkan tergantung pada banyak parameter. Bila pemilihan parameter tersebut tepat, kualitas gambar MRI dapat memberikan gambaran detail tubuh manusia dengan perbedaan yang kontras, sehingga anatomi dan patologi jaringan tubuh dapat dievaluasi secara teliti (Notosiswoyo & Suswati, 2004).

Pemeriksaan MRI bertujuan mengetahui karakteristik morfologi (lokasi, ukuran, bentuk, perluasan dan lain lain) dari keadaan patologis. Tujuan tersebut dapat diperoleh dengan menilai salah satu atau kombinasi gambar penampang tubuh aksial, sagital, koronal atau oblik, tergantung pada letak organ dan kemungkinan patologinya (Edelman, Hesselink, Zlatkin, & Crues, 1990).

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 pertama 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 dapat diambil secara terpisah sehingga dapat digunakan sebagai masukan proses yang lain. Segmentasi citra (*image segmentation*) merupakan suatu teknik pengelompokan (*clustering*) 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, 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 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 keabu-abuan (*grayscale*) dari citra tiap-tiap lokasi *pixel*nya (p,q).

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

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 *greyscale* 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	➔	71	52	53
G=0	G=50	G=0		198	80	7
B=100	B=80	B=100		5	71	0
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				

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 sebuah 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:

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

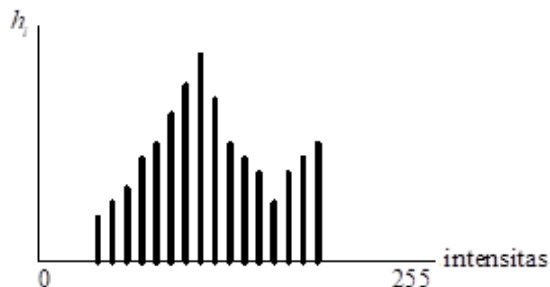
dimana,

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 gambar (*image enhancement*) adalah 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 atau menggunakan persamaan (2.6) (Prasetyo, 2011).

$$\begin{pmatrix} Z1 & Z2 & Z3 \\ Z4 & Z5 & Z6 \\ Z7 & Z8 & Z9 \end{pmatrix} \quad (2.4)$$

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

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

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

$$G_x = (Z7 + 2Z8 + Z9) - (Z1 + 2Z2 + Z3) \quad (2.7)$$

$$G_y = (Z3 + 2Z6 + Z9) - (Z1 + 2Z4 + Z7) \quad (2.8)$$

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

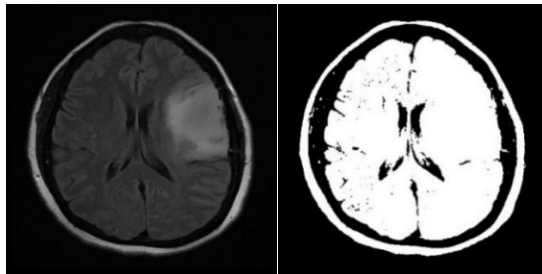
G_y = nilai *operator sobel* garis vertikal

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* 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).

Algoritma 2.1 Langkah-langkah *Thresholding* dengan Metode Ridler-Calvard

1. Tentukan Nilai *threshold* pertama

Nilai *threshold* pertama didapat berdasarkan persamaan (2.11).

$$T_0 = \frac{\sum_{i=1}^N y_i}{N} \quad (2.11)$$

dimana,

T_0 = 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).

$$\begin{aligned} y_i &\in G, \text{ jika } y_i \geq T_0 \\ y_i &\in L, \text{ jika } y_i < T_0 \end{aligned} \quad (2.12)$$

dimana:

y_i = nilai intensitas *pixel*

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

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

3. Tentukan nilai *threshold* baru

Untuk menentukan nilai *threshold* baru, diperlukan jumlah nilai rata-rata dari G dan L . Jumlah nilai rata-rata dari G dan L didapat berdasarkan persamaan (2.13).

$$\text{jumlah} = \frac{\sum_{i=1}^N G_i}{N} + \frac{\sum_{i=1}^N L_i}{N} \quad (2.13)$$

dimana:

G_i = Nilai intensitas G ke- i

L_i = Nilai intensitas L ke- i

N = jumlah data

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 menset 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 set 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 di dalam 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 dengan 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 $\{\pi_1, \pi_2, \dots, \pi_K\}$

π_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.

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

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

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

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 metode ini 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:

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

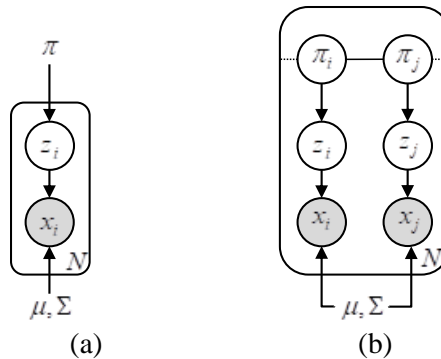
dapat dilihat dari fungsi *likelihood* pada persamaan (2.25), salah satu keuntungan digunakannya GMM adalah bentuk fungsi densitas yang sederhana, dan membutuhkan jumlah parameter yang tidak banyak (Nguyen, 2011).

2.6 *Markov Random Fields*

Untuk meningkatkan ketahanan (*robust*) terhadap *noise* pada segmentasi citra, model *mixture* dengan *Markov Random Fields* (MRF) telah banyak digunakan untuk pelabelan *pixel*, dimana distribusi *prior* bervariasi untuk setiap *pixel* yang bersesuaian dengan masing-masing label dan bergantung pada *neighborhood pixel*-nya (Nguyen &

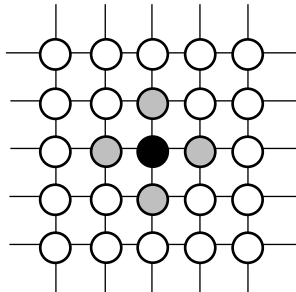
Jonathan Wu, 2013). Untuk memasukkan batasan *smoothing* spasial antar *neighbourhood pixel*, kelompok model *mixture* lain berdasarkan MRF telah diusulkan sebelumnya dimana MRF digunakan untuk memodelkan distribusi bersama (*joint distribution*) pada *prior* untuk setiap label *pixel* (Ji, Huang, Sun, & Cao, 2016).

Dengan menggunakan persamaan (2.19) diasumsikan setiap *pixel* i termasuk dalam satu kelompok atau *cluster* yang diberi indeks oleh variabel laten z_i (dimana variabel laten z diestimasi melalui nilai π). Maka, model grafis probabilistik GMM yang sesuai dapat ditunjukkan pada Gambar 2.4 (a) dan Model MRF-GMM dapat ditunjukkan secara grafis pada Gambar 2.4 (b) (Bishop, 2006).



Gambar 2.4 Model grafis probabilistik: (a) *Mixture model*; (b) *Mixture model* dengan dependensi spasial

Neighbourhood pixel dari sebuah *pixel* y_{ij} didefinisikan sebagai *pixel-pixel* yang mengelilingi *pixel* y_i (Kato & Zerubia, 2011). *Neighbourhood* pada *pixel* mengikuti sifat *markov*, karena hanya tergantung pada satu *pixel* sebelumnya. *Neighbourhood pixel* dapat diilustrasikan pada Gambar 2.5.



Gambar 2.5 Sistem orde pertama *neighbourhood pixel*

Untuk meningkatkan ketahanan terhadap *noise* GMM, distribusi MRF diterapkan untuk menggabungkan korelasi spasial di antara nilai-nilai *prior* (Nguyen & Jonathan Wu, 2013). Teorema Hammersley-Clifford menyatakan variabel random $\boldsymbol{\pi}$ adalah MRF sehubungan dengan sistem *neighbourhood* jika dan hanya jika $p(\boldsymbol{\pi})$ adalah distribusi *Gibbs* dengan tetangga terdekat *Gibbs* potensial V , yaitu:

$$p(\boldsymbol{\pi}) = \frac{1}{Z} \exp\left(-\beta \sum_{i=1}^N V_{C_i}(\boldsymbol{\pi})\right) \quad (2.26)$$

dengan,

$\boldsymbol{\pi}$ = himpunan vektor probabilitas

Z = *normalizing constant*

$V_{C_i}(\boldsymbol{\pi})$ = fungsi *clique* potensial dari vektor label *pixel* $\{\pi_i\}$ terhadap *neighbourhood pixelnya*

C = himpunan *clique* (dapat dituliskan sebagai $C = C_1 \cup C_2 \cup \dots \cup C_n$)

Secara umum, $V_{C_i}(\boldsymbol{\pi})$ didefinisikan sebagai fungsi pada persamaan (2.27) (Nikou, Galatsanos, & Likas, 2007).

$$V_{C_i}(\boldsymbol{\pi}) = \sum_{m \in C} g(d_{i,m}) \quad (2.27)$$

dimana $d_{i,m}$ menyatakan jarak antar dua vektor label $\{\pi_i\}$ dan $\{\pi_m\}$:

$$d_{i,m} = \|\{\pi_i\} - \{\pi_m\}\|^2 = \sum_{j=1}^K (\pi_{ij} - \pi_{mj})^2 \quad (2.28)$$

Sebuah *pixel* adalah *clique* jika setiap pasangan dari *pixel* yang berbeda adalah tetangga (*neighbourhood*). Fungsi V_c merepresentasikan kontribusi terhadap energi total dari *field* luar (*singleton clique*), interaksi berpasangan (*pair interaction/doubletons clique*), dan lain sebagainya (Kato & Zerubia, 2011).

2.7 Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)

Spatially Variant Finite Mixture Model (SVFMM) memberikan modifikasi pendekatan model *mixture* klasik untuk pelabelan *pixel*. SVFMM mengasumsikan model *mixture* dengan komponen K , masing-masing memiliki vektor densitas parameter θ_j (Sanjay & Hebert, 1988). Berdasarkan pendekatan SVFMM, probabilitas *pixel* ke- i masuk ke dalam *cluster* ke- j (label kelas) adalah sebagai berikut:

$$p(y_i = j) = \pi_{ij} \quad (2.29)$$

dimana $\sum_{j=1}^K \pi_{ij} = 1$ dan $0 \leq \pi_j^i \leq 1$; $i = 1, \dots, N$; $j = 1, 2, \dots, K$. Misal jika

$\pi = \{\pi_1, \pi_2, \dots, \pi_N\}$ menyatakan label probabilitas proporsi dengan

$\pi_i = (\pi_{i1}, \pi_{i2}, \dots, \pi_{iK})^T$ adalah vektor probabilitas dari *pixel* ke- i , maka

didefinisikan $\boldsymbol{\pi} = \{(\pi_1)^T, (\pi_2)^T, \dots, (\pi_N)^T\}$ sebagai himpunan dari

vektor probabilitas dan $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_K\}$ sebagai himpunan dari

parameter komponen (Sanjay & Hebert, 1988). Dengan demikian fungsi densitas pada persamaan (2.19) akan menjadi:

$$f(y_i | \boldsymbol{\pi}, \boldsymbol{\theta}) = \sum_{j=1}^K \pi_{ij} f(y_i | \theta_j) \quad (2.30)$$

dimana distribusi Gaussian $f(y_i | \theta_j)$ sama seperti pada persamaan

(2.23) dengan $\theta_j = \{\mu_j, \Sigma_j\}$ dan prior π_{ij} berbeda untuk tiap *pixel* i

serta dependen terhadap *neighbor pixel*.

Jika observasi y_i pada persamaan (2.30) dimodelkan sebagai independent terhadap label *cluster* ke- j , fungsi *joint conditional density* dari y_1, y_2, \dots, y_N dapat dimodelkan sebagai:

$$p(\mathbf{y} | \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{i=1}^N f(y_i | \boldsymbol{\pi}, \boldsymbol{\theta}) = \prod_{i=1}^N \left[\sum_{j=1}^K \pi_{ij} f(y_i | \boldsymbol{\theta}_j) \right] \quad (2.31)$$

dengan fungsi *log-likelihood* sebagai berikut.

$$L(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathbf{y}) = \sum_{i=1}^N \log \left(\sum_{j=1}^K \pi_{ij} f(y_i | \boldsymbol{\theta}_j) \right) \quad (2.32)$$

Seperti yang dijelaskan sebelumnya, untuk mengatasi segmentasi yang sensitif terhadap *noise* pada GMM, distribusi MRF pada persamaan (2.26) diterapkan untuk mengadaptasi korelasi spasial diantara nilai-nilai label. Sehingga fungsi densitas probabilitas posterior untuk SVFMM, dapat diberikan sesuai dengan aturan Bayes sebagai berikut.

$$p(\boldsymbol{\pi}, \boldsymbol{\theta} | \mathbf{y}) \propto p(\mathbf{y} | \boldsymbol{\pi}, \boldsymbol{\theta}) p(\boldsymbol{\pi}) \quad (2.33)$$

Dengan menerapkan persamaan (2.33), fungsi *log-likelihood* untuk SVFMM pada persamaan (2.32) dapat dituliskan kembali menjadi:

$$\begin{aligned} L(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathbf{y}) &= \log(p(\mathbf{y} | \boldsymbol{\pi}, \boldsymbol{\theta}) p(\boldsymbol{\pi})) \\ &= \sum_{i=1}^N \log \left\{ \sum_{j=1}^K \pi_{ij} f(y_i | \boldsymbol{\theta}_j) \right\} + \log p(\boldsymbol{\pi}) \\ &= \sum_{i=1}^N \log \left\{ \sum_{j=1}^K \pi_{ij} f(y_i | \boldsymbol{\theta}_j) \right\} - \log Z \\ &\quad - \beta \sum_{i=1}^N \sum_{j=1}^K \sum_{m \in C} (\pi_{ij} - \pi_{mj})^2 \end{aligned} \quad (2.34)$$

dimana Z merupakan suatu *normalizing constant* dan β merupakan skalar. Jika seluruh *constant* diabaikan, maka fungsi *log-likelihood* untuk SVFMM diberikan oleh:

$$\begin{aligned}
L(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathbf{y}) &= \sum_{i=1}^N \log \left\{ \sum_{j=1}^K \pi_{ij} f(y_i | \boldsymbol{\theta}_j) \right\} + \log p(\boldsymbol{\pi}) \\
&= \sum_{i=1}^N \log \left\{ \sum_{j=1}^K \pi_{ij} f(y_i | \boldsymbol{\theta}_j) \right\} - \beta \sum_{i=1}^N \sum_{j=1}^K \sum_{m \in C} (\pi_{ij} - \pi_{mj})^2
\end{aligned} \tag{2.35}$$

Pada penelitian ini, inisialisasi parameter yang digunakan untuk SVFMM didapatkan dari hasil estimasi parameter dengan GMM, sehingga metode SVFMM yang digunakan merupakan *hybrid* GMM-SVFMM. Berdasarkan penelitian sebelumnya, parameter model dapat diestimasi melalui *maximum likelihood* (ML) dengan menggunakan algoritma EM (Nguyen, 2011).

2.8 *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 ilmu Statistika, karena algoritma EM cukup sederhana namun pencapaian konvergen dalam iterasi stabil (Dempster, Laird, & Rubin, 1977). Untuk memaksimumkan fungsi *likelihood* pada persamaan (2.25) dan menentukan parameter dari GMM dan SVFMM 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 E-step dilakukan perhitungan nilai ekspektasi dari parameter data, sementara itu pada M-step dilakukan perhitungan nilai estimasi parameter dengan menggunakan nilai ekspektasi yang ditemukan pada tahap sebelumnya. Kedua tahap ini akan terus dilakukan sampai mencapai nilai konvergen (Dempster, Laird, & Rubin, 1977)

2.8.1 Algoritma *Expectation-Maximization* (EM) untuk GMM

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* μ_j dari komponen Gaussian.

$$\frac{\partial L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y})}{\partial \mu_j} = \frac{\partial \left(\sum_{i=1}^N \log \left(\sum_{j=1}^K \pi_j f_j(y_i | \boldsymbol{\theta}_j) \right) \right)}{\partial \mu_j} \quad (2.36)$$

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

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

$$\begin{aligned} z_{ij}^{(t)} &= f(z_{ij} | y_i, \mu_j^{(t)}, \sigma_j^{(t)}) \\ &= \frac{f(y_i | z_{ij}, \mu_j^{(t)}, \sigma_j^{(t)}) f(z_{ij} | \mu_j^{(t)}, \sigma_j^{(t)})}{f(y_i | \mu_j^{(t)}, \sigma_j^{(t)})} \\ &= \frac{\pi_j^{(t)} f(y_i | \mu_j^{(t)}, \sigma_j^{(t)})}{\sum_{l=1}^K \pi_l^{(t)} f(y_i | \mu_l^{(t)}, \sigma_l^{(t)})} \end{aligned} \quad (2.37)$$

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

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^N z_{ij}^{(t)} y_i}{\sum_{i=1}^N z_{ij}^{(t)}} \quad (2.38)$$

Selanjutnya, jika fungsi *likelihood* para persamaan (2.25) diturunkan terhadap σ_j , maka didapatkan:

$$\left[\sigma_j^2\right]^{(t+1)} = \frac{\sum_{i=1}^N z_{ij}^{(t)} \left(y_i - \mu_j^{(t+1)}\right)^2}{\sum_{i=1}^N z_{ij}^{(t)}} \quad (2.39)$$

Kemudian, dapat diturunkan fungsi *likelihood* $L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y})$ terhadap distribusi prior π_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* η dan memaksimumkan turunan berikut.

$$\frac{\partial}{\partial \pi_j} \left[L(\boldsymbol{\mu}, \boldsymbol{\sigma}, \boldsymbol{\pi} | \mathbf{y}) - \eta \left(\sum_{j=1}^K \pi_j - 1 \right) \right] = 0 \quad (2.40)$$

dimana selanjutnya didapatkan:

$$\sum_{i=1}^N \frac{f(y_i | \mu_j, \sigma_j)}{\sum_{l=1}^K \pi_l f(y_i | \mu_l, \sigma_l)} - \eta = 0 \quad (2.41)$$

Jika kedua sisi dikalikan dengan π_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 η dan menyusun kembali turunan terhadap π_j didapatkan:

$$\pi_j^{(t+1)} = \frac{1}{N} \sum_{i=1}^N z_{ij}^{(t)} \quad (2.42)$$

(Nguyen, 2011)

Algoritma 2.2 Algoritma EM untuk Segmentasi dengan GMM

1. Inisialisasi parameter $\{\mu_j, \sigma_j, \pi_j\}$, *mean* μ_j ; nilai *varians* σ_j ; dan distribusi prior π_j .

2. E-Step : Menghitung nilai z_{ij} pada persamaan (2.37) menggunakan nilai parameter awal.
3. M-Step : Estimasi ulang parameter $\{\mu_j, \sigma_j, \pi_j\}$
 - a) *Update mean* μ_j dengan menggunakan persamaan (2.38).
 - b) *Update varians* σ_j dengan menggunakan persamaan (2.39).
 - c) *Update distribusi prior* π_j dengan menggunakan persamaan (2.42).
4. 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.

2.8.2 Algoritma *Expectation-Maximization* (EM) untuk *Hybrid GMM-SVFMM*

Apabila dibandingkan dengan fungsi *log-likelihood* dari GMM pada persamaan (2.25), fungsi *log-likelihood* untuk SVFMM pada persamaan (2.35) cukup kompleks. Untuk memaksimumkan fungsi *likelihood* ini terhadap parameter-parameter $\{\mu_j, \Sigma_j, \pi_{ij}\}$, suatu algoritma EM iteratif diterapkan. Inisialisasi parameter pada algoritma EM untuk SVFMM didapatkan dari estimasi parameter dengan GMM, sehingga metode SVFMM yang digunakan merupakan *hybrid GMM-SVFMM*. Dalam penerapan memaksimumkan fungsi *log-likelihood* $L(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Sigma} | \mathbf{y})$ pada persamaan (2.35), akan digunakan variable laten z_{ij} terhadap model (Nguyen, 2011).

$$\begin{aligned}
 H(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathbf{y}) = & \sum_{i=1}^N \sum_{j=1}^K z_{ij}^{(t)} \left\{ \log \pi_{ij} + \log f(y_i | \theta_j) \right\} \\
 & - \beta \sum_{i=1}^N \sum_{j=1}^K \sum_{m \in C} (\pi_{ij} - \pi_{mj})^2
 \end{aligned} \tag{2.43}$$

Dimana nilai *conditional expectation* z_{ij} dari variabel laten, dapat dihitung sebagai berikut.

$$z_{ij}^{(t)} = \frac{\pi_{ij}^{(t)} f(y_i | \mu_j^{(t)}, \Sigma_j^{(t)})}{\sum_{k=1}^K \pi_{ik}^{(t)} f(y_i | \mu_k^{(t)}, \Sigma_k^{(t)})} \quad (2.44)$$

Selanjutnya, persamaan $H(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathbf{y})$ diturunkan terhadap mean μ_j pada tahap iterasi $(t + 1)$.

$$\frac{\partial H}{\partial \mu_j} = \sum_{i=1}^N z_{ij}^{(t)} \left[-\frac{1}{2} (2\Sigma_j^{-1} \mu_j - 2\Sigma_j^{-1} y_i) \right] \quad (2.45)$$

Penyelesaian untuk $\partial H / \partial \mu_j = 0$ menghasilkan nilai minimum dari μ_j pada tahap iterasi $(t + 1)$:

$$\mu_j^{(t+1)} = \frac{\sum_{i=1}^N z_{ij}^{(t)} y_i}{\sum_{i=1}^N z_{ij}^{(t)}} \quad (2.46)$$

Kemudian, derivatif fungsi pada persamaan (2.43) terhadap Σ_j^{-1} pada tahap iterasi $(t + 1)$ diberikan oleh:

$$\frac{\partial H}{\partial \Sigma_j^{-1}} = \sum_{i=1}^N z_{ij}^{(t)} \left[\frac{1}{2} \Sigma_j - \frac{1}{2} (y_i - \mu_j)(y_i - \mu_j)^T \right] \quad (2.47)$$

dan penyelesaian menghasilkan sama dengan nol:

$$\Sigma_j^{(t+1)} = \frac{\sum_{i=1}^N z_{ij}^{(t)} (y_i - \mu_j)(y_i - \mu_j)^T}{\sum_{i=1}^N z_{ij}^{(t)}} \quad (2.48)$$

Sementara itu, karena kompleksitas dari fungsi log-likelihood pada persamaan (2.35), M-step pada algoritma EM tidak dapat mengevaluasi distribusi prior π_{ij} dalam bentuk *closed form*. Dengan demikian, untuk memaksimalkan fungsi objektif $H(\boldsymbol{\theta}, \boldsymbol{\pi} | \mathbf{y})$ terhadap π_{ij} , maka dilakukan derivatif sama dengan nol, dan diperoleh bentuk kuadratik sebagai berikut.

$$4\beta N_i \left(\pi_{ij}^{(t+1)} \right)^2 - 4\beta \pi_{ij}^{(t+1)} \sum_{m \in C} \pi_{mj} - z_{ij}^{(t)} = 0 \quad (2.49)$$

Dimana N_i merupakan himpunan dari neighbour yang berada disekitar *pixel* Y_i . Persamaan (2.49) memiliki dua akar:

$$\pi_{ij}^{(t+1)} = \frac{\sum_{m \in C} \pi_{mj} \pm \sqrt{\left(\sum_{m \in C} \pi_{mj} \right)^2 + \frac{N_i}{\beta} z_{ij}^{(t)}}}{2N_i} \quad (2.50)$$

Berdasarkan persamaan (2.50), dipilih akar dengan notasi positif + karena akan menghasilkan $\pi_{ij} \geq 0$. Dalam mengestimasi distribusi prior π_{ij} , perlu diterapkan domain $0 \leq \pi_{ij} \leq 1$ dan $\sum_{j=1}^K \pi_{ij} = 1$. Persamaan-persamaan diatas merupakan formula untuk *update* parameter pada setiap M-step di algoritma EM (Nguyen, 2011).

Algoritma 2.3 Algoritma EM untuk Segmentasi dengan *Hybrid* GMM-SVFMM

1. Inisialisasi parameter $\{\mu_j, \Sigma_j, \pi_{ij}\}$, *mean* μ_j ; nilai *covarians* Σ_j ; dan distribusi prior π_{ij} , dari hasil estimasi parameter GMM.
2. E-Step : Menghitung nilai z_{ij} pada persamaan (2.44) menggunakan nilai parameter awal.
3. M-Step : Estimasi ulang parameter $\{\mu_j, \Sigma_j, \pi_{ij}\}$
 - d) *Update mean* μ_j dengan menggunakan persamaan (2.46).
 - e) *Update kovarians* Σ_j dengan menggunakan persamaan (2.48).
 - f) *Update distribusi prior* π_{ij} dengan menggunakan persamaan (2.50).
4. Evaluasi fungsi *log-likelihood* pada persamaan (2.35) dan periksa konvergen, baik dari fungsi *log-likelihood* maupun nilai parameter. Jika kriteria konvergen nilai *error* $< 0,0001$ belum terpenuhi, kembali ke langkah 2.

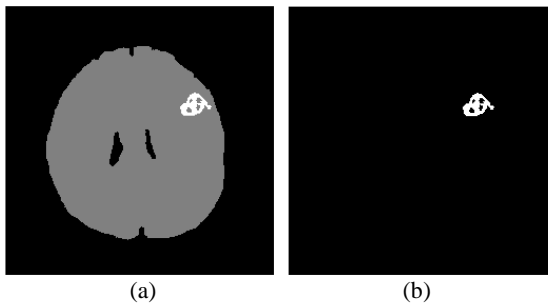
2.9 Correct Classification Ratio (CCR)

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 (2.51).

$$CCR = \sum_{j=1}^2 \frac{|GT_j \cap Seg_j|}{|GT|} \quad (2.51)$$

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 2.6.



Gambar 2.6 *Ground Truth* (a) Segmentasi MRI Tumor Otak (b) ROI Tumor Otak

Dapat dilihat pada Gambar 2.6 (a) yang merupakan *ground truth* model segmentasi untuk MRI tumor otak. Penetapan *ground truth* ini dilakukan secara manual dengan validasi oleh ahli medis. Selanjutnya

yang menjadi target pada perhitungan CCR adalah area ROI tumor otak saja, yang ditampilkan pada Gambar 2.6 (b). Area segmentasi ROI dan non-ROI tumor otak inilah yang akan dijadikan sebagai *ground truth* pada perhitungan nilai CCR untuk metode GMM dan *hybrid* GMM-SVFMM.

“Halaman ini sengaja dikosongkan”

BAB III METODOLOGI PENELITIAN

Bab ini membahas mengenai kerangka konsep segmentasi citra MRI, sumber data yang digunakan, ruang implementasi yang digunakan peneliti, variabel penelitian dan struktur data, serta langkah analisis yang dilakukan dalam penelitian. Langkah analisis dalam penelitian ini kemudian disajikan dalam bentuk diagram alir.

3.1 Sumber Data

Data yang digunakan dalam penelitian ini merupakan data sekunder, berupa data rekam medis pasien tumor otak di RSUD Dr. Soetomo, Surabaya. *Sequence* citra MRI yang akan digunakan dipilih berdasarkan rekomendasi oleh ahli dari sudut pandang medis, yaitu *ax T1 memp+C slice* ke 18.

3.2 Ruang Implementasi

Lingkungan implementasi program terdiri atas lingkungan perangkat keras dan lingkungan perangkat lunak yang digunakan dalam membangun sistem penelitian ini, yang diuraikan pada Tabel 3.1.

Tabel 3.1 Spesifikasi Ruang Implementasi

Perangkat Keras	Perangkat Lunak		
Prosesor	Intel (R) Core (TM) i5-4200U CPU @ 1.60GHz 1.6 GHz	Sistem Operasi	Windows 10 Home 64-bit
Memori RAM	6.00 GB	Paket Program	Matlab 9.0 R2016a

3.3 Variabel Penelitian dan Struktur Data

Variabel penelitian dan struktur data dari rekam medis citra MRI yang digunakan pada penelitian ini dijelaskan pada Tabel 3.2 dan Tabel 3.3.

Tabel 3.2 Variabel Penelitian

Variabel	Keterangan	Skala data
<i>Grayscale</i>	Variabel yang digunakan merupakan besarnya intensitas keabuan di tiap-tiap <i>pixel</i> citra MRI.	Interval

Variabel *Grayscale* mewakili intensitas keabuan pada tiap *pixel* citra MRI, dimana intensitas citra MRI dapat direpresentasikan menjadi sebuah matriks berukuran 256×256 . Elemen matriks tersebut merupakan besar intensitas keabuan yang diwakili dengan nilai 0-255, sehingga tiap-tiap besaran elemen akan dikumpulkan dan dipisahkan berdasarkan besar intensitasnya. Selanjutnya dibentuk menjadi sebuah struktur data seperti pada persamaan (2.1).

Tabel 3.3 Struktur Data

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

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

3.4 Langkah-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* direpresentasikan menjadi sebuah matriks berukuran 256×256 .

2. *Preprocessing* data citra MRI dengan tahapan sebagai berikut.
 - a) Melakukan *thresholding* sebagai tahapan dalam penghapusan atribut film dan pelepasan tengkorak (*skull stripping*), di dalam proses ini *pixel* citra yang nilainya kurang dari atau sama dengan nilai *threshold* akan diganti

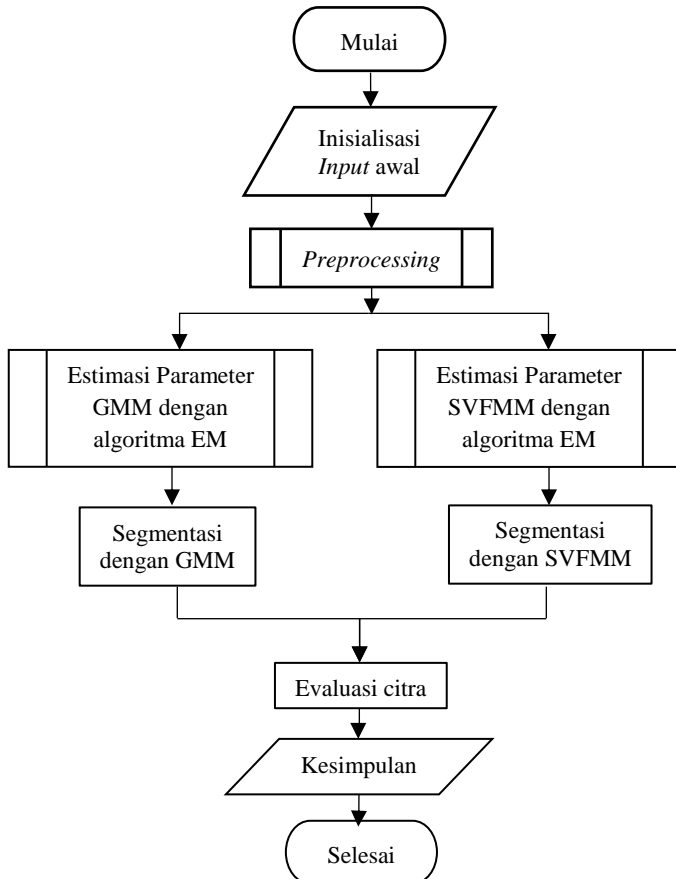
dengan warna hitam. Sedangkan, *pixel* citra yang nilainya lebih dari *threshold* akan diganti dengan putih. *Thresholding* yang dilakukan sesuai dengan Algoritma 2.2.

- b) Menentukan *head contour*, penentuan *head contour* dilakukan dengan cara mencari *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) Menggabungkan citra hasil deteksi tepi dan citra setelah proses *morphological opening*.
 - h) Melakukan *blocking area* kepala setelah proses penggabungan.
 - i) Menentukan area otak dengan cara substraksi.
 - j) Mengembalikan *pixel-pixel* yang bernilai 1 dengan nilai intensitasnya.
3. Melakukan estimasi parameter menggunakan *algoritma Expectation-Maximization* (EM) untuk GMM sesuai dengan Algoritma 2.2.
 4. Melakukan estimasi parameter menggunakan *algoritma Expectation-Maximization* (EM) untuk *hybrid* GMM-SVFMM sesuai dengan Algoritma 2.3.
 5. Menentukan jumlah *cluster* terbaik dan mengevaluasi hasil model segmentasi citra MRI dengan menghitung *Correct Classification Ratio* (CCR).
 6. Melakukan analisis segmentasi citra MRI menggunakan GMM dan *hybrid* GMM-SVFMM.

7. Membuat kesimpulan.

3.5 Diagram Alir

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



Gambar 3.1 Diagram Alir Penelitian

BAB IV IMPLEMENTASI, UJI COBA, ANALISIS DAN PEMBAHASAN

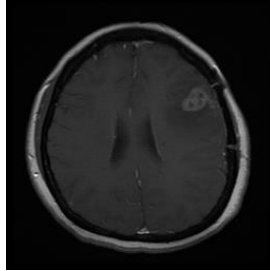
Bab ini membahas mengenai pemilihan data citra MRI untuk dilakukan segmentasi, tahapan *preprocessing* yang dilakukan, uji coba serta pemilihan jumlah *cluster* melalui validasi *cluster*, implementasi metode *Gaussian Mixture Model* (GMM) dan *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-SVFMM) dengan algoritma *Expectation-Maximization* (EM) untuk menentukan area ROI tumor otak, evaluasi hasil segmentasi, serta perbandingan kedua metode dalam segmentasi citra MRI tumor otak.

4.1 Pemilihan Data dan *Preprocessing*

Langkah pertama yang dilakukan dalam segmentasi citra MRI adalah pemilihan data citra MRI. Selanjutnya sebelum memasuki tahap segmentasi, dilakukan *preprocessing* terlebih dahulu untuk memperbaiki kualitas citra yang 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*, dimana *sequence* ini merupakan citra MRI hasil potongan *axial* otak pasien dengan pemberian *contrast media*. Alasan dipilihnya *sequence* ini adalah karena 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 adalah, *sequence ax T1 memp+C slice* ke-18.

4.1.1 Data Masukan *Preprocessing*

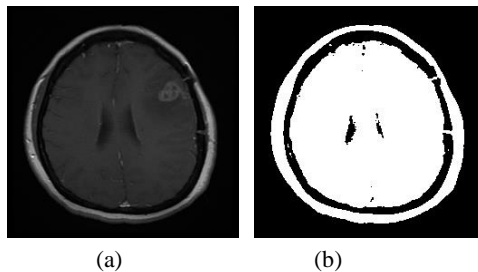
Syntax Preprocessing yang digunakan pada penelitian ini diperoleh dari penelitian sebelumnya oleh Sianipar (2017). Meskipun demikian, data masukan tahap *preprocessing* yang digunakan berbeda dengan penelitian sebelumnya untuk mendapatkan hasil yang bervariasi.



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

Dapat dilihat pada Gambar 4.1 citra MRI masih berupa area otak dengan tulang kepala. Tujuan dilakukannya tahap *preprocessing* adalah 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 tumornya (Non-ROI). *Preprocessing* data citra MRI dilakukan dengan tahapan berikut.

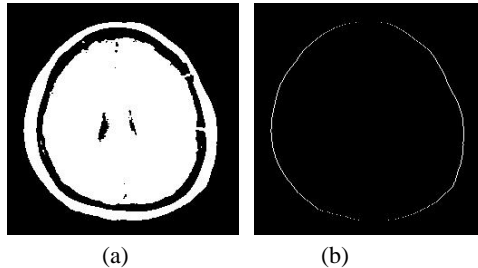
- 1) Melakukan *thresholding*, dimana dalam proses ini *pixel* citra yang nilainya kurang dari atau sama dengan nilai *threshold* akan diganti dengan warna hitam. Sedangkan, *pixel* citra yang nilainya lebih dari *threshold* akan diganti dengan putih. *Thresholding* yang dilakukan sesuai dengan Algoritma 2.2, dan 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 bagian (a) merupakan citra masukan MRI tumor otak yang berupa citra 2 dimensi komposisi RGB. Sementara itu, Gambar 4.2 bagian (b) merupakan hasil proses *thresholding* yang berupa citra biner.

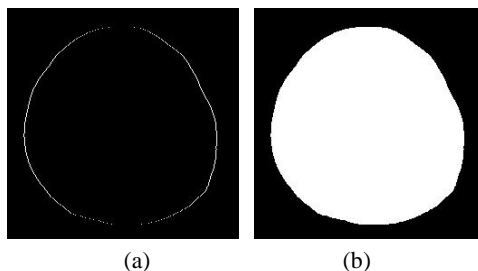
- 2) Menentukan *head contour*, penentuan *head contour* dilakukan dengan cara mencari *pixel* yang berwarna putih pada setiap baris. Selain *pixel* putih pertama dan terakhir akan dibiarkan berwarna hitam. *Pixel* yang berwarna putih pertama adalah *head contour*. Hasil dari proses *head contour* ditampilkan 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 *contour* kepala. *Contour* kepala merupakan batas antara citra dengan *background*. Sementara itu, hasil 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.

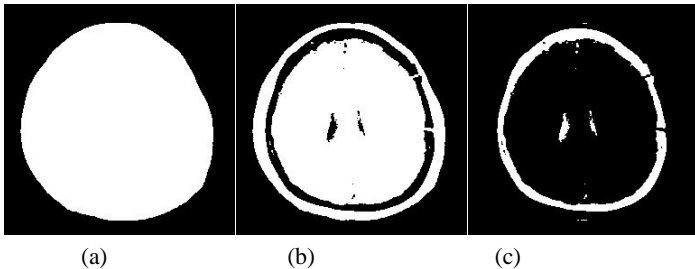
- 3) Melakukan proses *head masking* dengan cara *blocking* area *head contour*. Citra hasil head masking dapat dilihat pada Gambar 4.4.



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*.

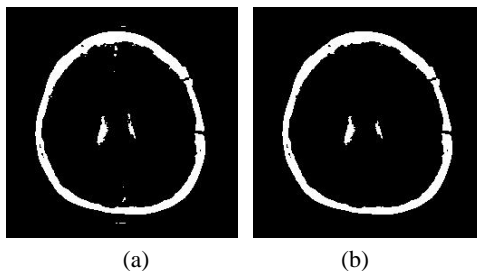
- 4) Melakukan substraksi antara *head mask* dan *threshold*.. Substraksi dilakukan dengan pengurangan antara citra *head mask* dengan citra *thresholding*. Hasil tahap substraksi head mask dengan *thresholding* ditampilkan pada Gambar 4.5.



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).

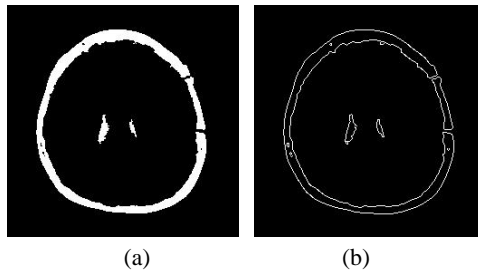
- 5) Melakukan proses *morphological opening* untuk menghilangkan *pixel* yang kecil. Citra hasil opening morfologi dapat dilihat pada Gambar 4.6.



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.

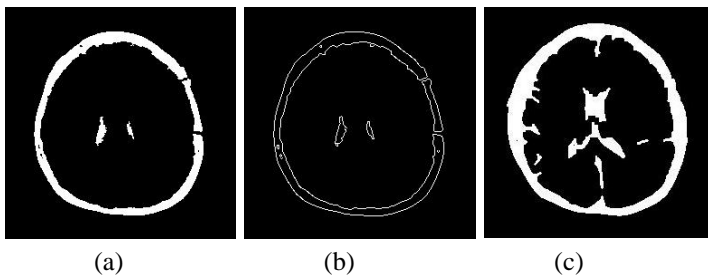
- 6) Melakukan proses deteksi tepi (*edge detection*) menggunakan *operator sobel* agar *pixel-pixel* area otak tidak terputus. Hasil *edge detection* ditampilkan pada Gambar 4.7.



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 untuk proses deteksi tepi, dan Gambar 4.7 bagian (b) merupakan citra hasil proses deteksi tepi.

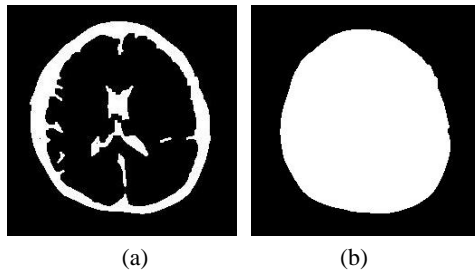
- 7) Menggabungkan citra hasil deteksi tepi dan citra setelah proses *morphological opening*, untuk menentukan area antara tulang kepala dan otak. Citra Hasil penggabungan dapat dilihat pada Gambar 4.8.



Gambar 4.8 Citra Tahapan Penggabungan (a) Citra Hasil *Opening* (b) Citra Hasil *Edge Detection* (c) Citra Hasil Penggabungan

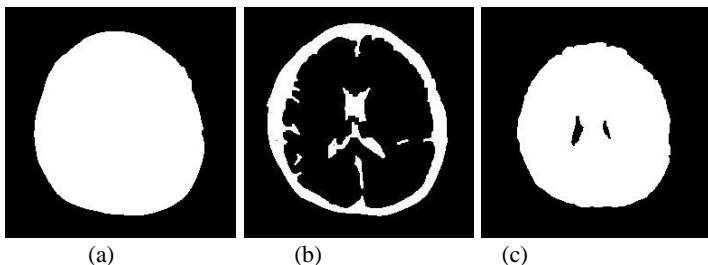
Pada tahap ini, penentuan area antara tulang kepala dan otak dilakukan agar selanjutnya dapat dilakukan proses untuk menentukan area otak saja.

- 8) Melakukan *blocking* area kepala setelah proses penggabungan. Citra hasil *blocking* akan digunakan untuk proses subtraksi penentuan area otak. Citra hasil *blocking* akan digunakan untuk proses subtraksi penentuan area otak. Hasil tahapan *blocking* ditampilkan pada Gambar 4.9.



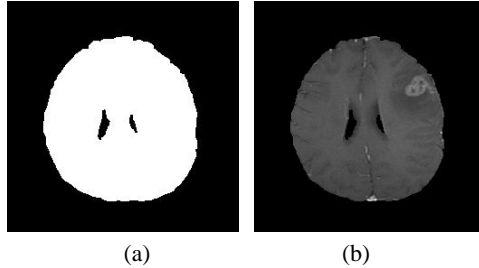
Gambar 4.9 Citra Tahapan *Blocking* (a) Citra Hasil Penggabungan (b) Citra Hasil *Blocking*

- 9) Menentukan area otak dengan cara subtraksi. pada tahap ini, dilakukan pengurangan citra hasil *blocking* dengan citra hasil penggabungan, sehingga didapatkan citra yang merupakan area otak saja. Citra hasil subtraksi pada tahapan ini, dapat dilihat pada Gambar 4.10.



Gambar 4.10 Citra Tahapan Subtraksi Area Otak (a) Citra Hasil *Blocking* (b) Citra Hasil Penggabungan (c) Citra Hasil Subtraksi Area Otak

- 10) Mengembalikan *pixel-pixel* yang bernilai 1 dengan nilai intensitasnya, dimana hasil perubahan ditunjukkan pada Gambar 4.11.



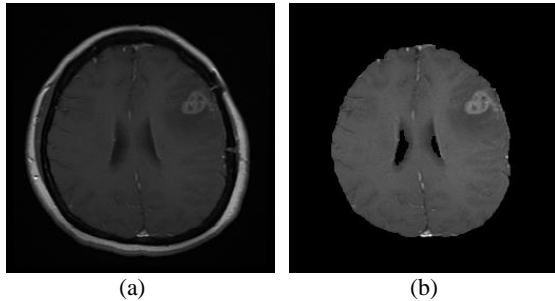
Gambar 4.11 Citra Akhir *Preprocessing* (a) Citra Hasil *Preprocessing* (b) Citra Area Otak

Area otak yang merupakan hasil proses substraksi masih berupa citra biner. Sementara itu, untuk melanjutkan analisis *model-based clustering* dibutuhkan citra yang memiliki komposisi warna *greyscale*. Sehingga proses ini merupakan proses pengembalian citra biner ke dalam komposisi warna *greyscale* asalnya.

Data citra MRI yang dihasilkan setelah *preprocessing* adalah berupa sebuah matriks yang memiliki skala data ordinal atau dalam program matlab terbaca sebagai *uint8*. Data citra ini mempunyai *range* sebesar 0-255. Dimana ketika memasuki proses segmentasi, data tersebut akan dirubah menjadi bilangan real dan dalam program matlab terbaca sebagai *double*. Tujuan dilakukannya perubahan jenis bilangan pada data citra MRI adalah agar citra dapat diolah menggunakan metode GMM dan SVFMM.

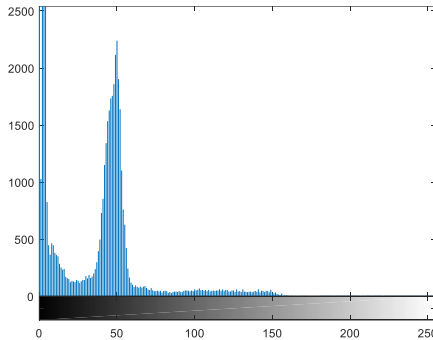
4.1.2 Histogram Citra MRI Tumor Otak

Dalam segmentasi citra, eksplorasi data dilakukan dengan membangun histogram dari matriks citra yang terbentuk. Dari histogram inilah selanjutnya dapat dilakukan pendugaan awal bawa 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.



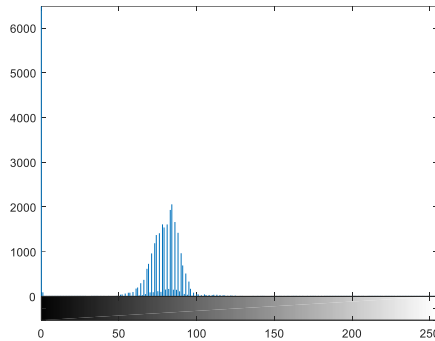
Gambar 4.12 Citra Hasil *Preprocessing* (a) Citra Awal (b) Citra Setelah *Preprocessing*

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



Gambar 4.13 Histogram Citra Sebelum *Preprocessing*

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



Gambar 4.14 Histogram Citra Setelah *Preprocessing*

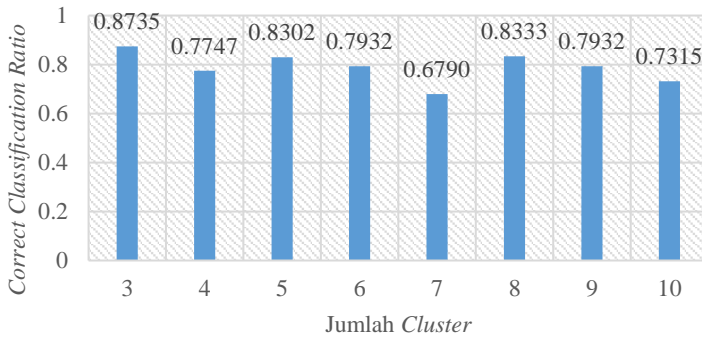
Dapat dilihat pada Gambar 4.14, terjadi perubahan bentuk histogram pada citra setelah dilakukan *preprocessing*. Setelah dilakukan *preprocessing*, diduga data MRI berpola *mixture* sehingga untuk segmentasi citra MRI digunakan metode clustering berbasis *mixture model* dengan GMM dan *hybrid* GMM-SVFMM.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

4.2 Uji Coba, Pemilihan, dan Validasi *Cluster*

Dalam menentukan jumlah *cluster* yang akan digunakan pada segmentasi citra MRI dengan GMM dan *Hybrid* GMM-SVFMM, dilakukan uji coba menggunakan beberapa jumlah *cluster*. Setelah dilakukan uji coba, pemilihan *cluster* terbaik akan ditentukan melalui nilai *Correct Classification Ratio* (CCR). Bersamaan dengan CCR, dalam validasi jumlah *cluster* penilaian subjektif dari pihak medis juga dipertimbangkan.

4.2.1 Validasi Jumlah *Cluster* untuk Metode *Gaussian Mixture Model* (GMM)

Validasi jumlah *cluster* yang akan digunakan pada segmentasi dengan GMM adalah dengan membandingkan nilai *Correct Classification Ratio* (CCR) terhadap uji coba yang dilakukan untuk jumlah *cluster* 2-10 *cluster*. Nilai CCR untuk validasi jumlah *cluster* ditampilkan pada Gambar 4.15.



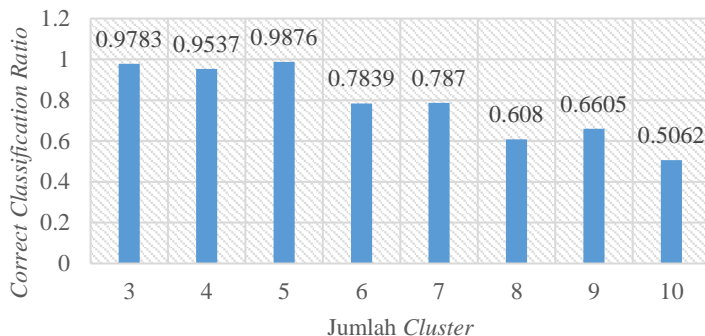
Gambar 4.15 Perbandingan Nilai *Correct Classification Ratio* (CCR) Metode GMM

Pada Gambar 4.15, dapat terlihat bahwa nilai CCR tertinggi diperoleh untuk jumlah *cluster* sebanyak 3 *cluster* dengan nilai CCR sebesar 0,8735. Sementara itu, untuk segmentasi dengan jumlah *cluster* sebanyak 2 *cluster* tidak dapat dihitung nilai CCR, karena model segmentasi GMM dengan 2 *cluster* belum mampu menangkap ROI-nya. Berdasarkan penilaian subjektif dari ahli medis, jumlah *cluster* sebanyak 3 *cluster* sudah mampu menangkap ROI dan dapat digunakan dalam pemilihan jumlah *cluster* model segmentasi.

4.2.2 Validasi Jumlah *Cluster* untuk Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-SVFMM)

Pada validasi jumlah *cluster* untuk segmentasi dengan *hybrid* GMM-SVFMM, juga dilakukan uji coba terhadap jumlah *cluster* 2-10 *cluster*. Selanjutnya dari hasil yang diperoleh, akan dibandingkan nilai *Correct Classification Ratio* (CCR) untuk menentukan jumlah *cluster* terbaik. Perbandingan nilai CCR untuk metode *hybrid* GMM-SFVMM dapat dilihat pada Gambar 4.16.

Berdasarkan Gambar 4.16, diketahui bahwa jumlah *cluster* sebanyak 3 *cluster* memiliki nilai CCR tertinggi diantara lainnya, yaitu sebesar 0,9783. Sementara itu seperti pada GMM sebelumnya, untuk segmentasi dengan jumlah *cluster* sebanyak 2 *cluster* tidak dapat dihitung nilai CCR, karena model segmentasi *hybrid* GMM-SVFMM dengan 2 *cluster* belum mampu menangkap ROI-nya.



Gambar 4.16 Perbandingan Nilai *Correct Classification Ratio* (CCR) Metode *Hybrid GMM-SVFMM*

Apabila ditinjau dari penilaian subjektif oleh pihak medis, jumlah *cluster* terbaik yang digunakan untuk segmentasi dengan metode *hybrid GMM-SVFMM* adalah sebanyak 3 *cluster*.

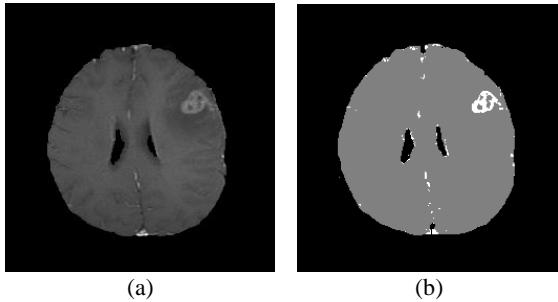
4.3 Segmentasi Citra MRI dengan *Model Based Clustering*

Setelah ditentukan jumlah *cluster* terbaik untuk masing-masing metode, selanjutnya dapat dilakukan segmentasi citra MRI untuk menentukan area tumor otak. Segmentasi citra dilakukan dengan *model based clustering* menggunakan dua metode, yaitu *Gaussian Mixture Model* (GMM) dan *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-SVFMM) dengan algoritma *Expectation-Maximization* (EM). Data yang digunakan adalah *sequence ax T1 memp+C* yang merupakan citra MRI otak dengan potongan *axial* yang telah ditambahkan *contrast*. Pemilihan *slice*, yaitu *slice* ke-18 merupakan citra MRI dengan pusat tumor otak. Data yang telah dilakukan *preprocessing* sebelumnya, kemudian dilanjutkan untuk segmentasi dengan masing-masing metode.

4.3.1 Segmentasi Citra dengan Metode *Gaussian Mixture Model* (GMM)

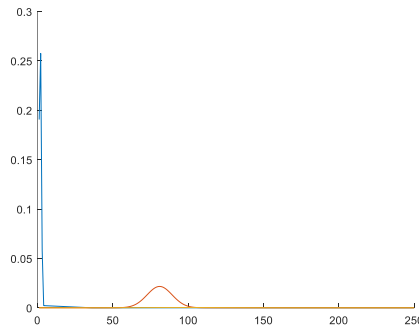
Segmentasi citra dengan metode *Gaussian Mixture Model* (GMM) dilakukan dengan asumsi bahwa data citra MRI terdiri dari beberapa komponen distribusi normal. *Syntax* pemrograman untuk segmentasi dengan metode GMM diperoleh dari penelitian sebelumnya oleh Sianipar (2017). Validasi *cluster* yang dilakukan sebelumnya,

menghasilkan jumlah *cluster* terbaik untuk segmentasi dengan GMM adalah sebanyak 3 *cluster*. Algoritma yang digunakan untuk mengestimasi parameter GMM adalah algoritma *Expectation-Maximization* (EM). Hasil segmentasi citra dengan metode GMM menggunakan algoritma EM ditampilkan pada Gambar 4.17.



Gambar 4.17 Citra Hasil Segmentasi dengan GMM (a) Citra *Preprocessing* (b) Citra Hasil Segmentasi GMM 3 *Cluster*

Pada Gambar 4.17 (a) dapat dilihat citra masukan untuk segmentasi dengan GMM dari hasil *preprocessing*. Sementara itu, pada Gambar 4.17 (b), dapat dilihat bahwa hasil segmentasi GMM dengan 3 *cluster* menghasilkan segmentasi area tumor otak yang diindikasikan dengan warna abu-abu muda. Distribusi data dari hasil segmentasi GMM dengan 5 *cluster* dapat dilihat melalui Gambar 4.18.



Gambar 4.18 Histogram Hasil Segmentasi Citra menggunakan GMM-3 *cluster*

Berdasarkan histogram pada Gambar 4.18 dapat diduga bahwa terdapat 3 puncak untuk data hasil segmentasi. Parameter-parameter

GMM dengan 3 komponen distribusi diperoleh dengan bantuan program ditampilkan pada Tabel 4.1.

Tabel 4.1 Estimasi Parameter Model Segmentasi *Gaussian Mixture Model 5 Cluster*

<i>Cluster</i>	π	μ	σ
1	0,5325	0,6684	0,5591
2	0,4487	79,9858	68,2674
3	0,0188	94,4135	769,0575

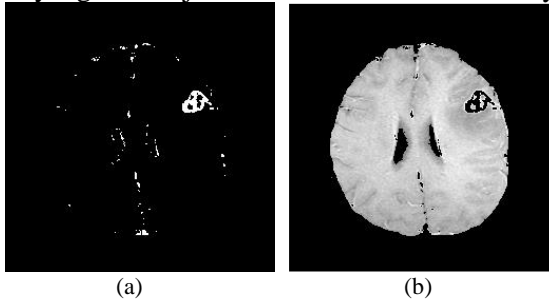
dan fungsi densitasnya dapat dituliskan pada persamaan (4.1).

$$\begin{aligned}
 f(y|\mu, \sigma, \pi) &= \sum_{j=1}^3 \pi_j f(y|\mu_j, \sigma_j) \\
 &= \pi_1 N(y|\mu_1, \sigma_1) + \pi_2 N(y|\mu_2, \sigma_2) + \pi_3 N(y|\mu_3, \sigma_3) \\
 &= 0,5325 \frac{1}{\sqrt{2\pi 0,5591}} \exp\left(-\frac{1}{2} \frac{(y_i - 0,6684)^2}{0,5591}\right) \\
 &\quad + 0,4487 \frac{1}{\sqrt{2\pi 68,2674}} \exp\left(-\frac{1}{2} \frac{(y_i - 79,9858)^2}{68,2674}\right) \\
 &\quad + 0,0188 \frac{1}{\sqrt{2\pi 769,0575}} \exp\left(-\frac{1}{2} \frac{(y_i - 94,4135)^2}{769,0575}\right)
 \end{aligned} \tag{4.1}$$

Melalui persamaan (4.1), dapat dilihat ketiga distribusi normal yang menyusun model segmentasi citra MRI. Dengan bantuan penilaian subjektif dari pihak medis, dapat ditentukan distribusi dari area tumor otak atau 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 atau Non-ROI (*Non-Region of Interest*). Untuk melihat dengan jelas area ROI dan Non-ROI, hasil segmentasi dapat disajikan pada Gambar 4.19.

Pada Gambar 4.19, 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.



Gambar 4.19 Citra Hasil Segmentasi GMM-3 Cluster (a) Area ROI (b) Area Non-ROI

Selanjutnya, untuk mendeteksi perubahan intensitas pada area ROI dilakukan *subjective analysis* dengan deteksi tepi. Tampilan *subjective analysis* dapat dilihat pada Gambar 4.20.



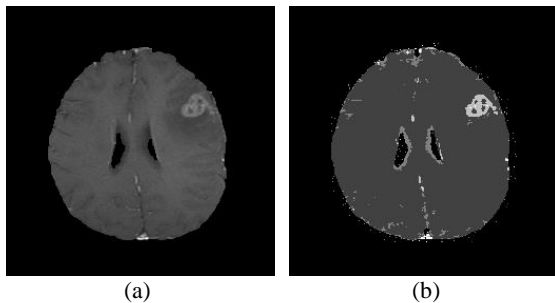
Gambar 4.20 *Subjective Analysis* GMM ROI Citra MRI Sequence ax T1 memp+C

Berdasarkan Gambar 4.20, dapat terlihat tepi-tepi yang menjadi area ROI tumor otak. *Subjective analysis* dilakukan dengan tujuan untuk membantu ahli medis memberi penilaian secara subjektif dalam menentukan area ROI tumor otak.

4.3.2 Segmentasi Citra dengan Metode Hybrid Gaussian Mixture Model – *Spatially Variant Finite Mixture Model* (GMM-SVFMM)

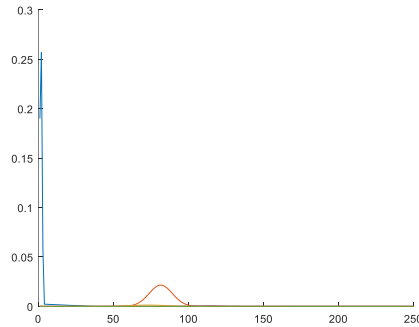
Metode lainnya yang akan digunakan dalam segmentasi citra MRI adalah *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model* (GMM-SVFMM). *Hybrid* GMM-SVFMM memiliki komponen fungsi distribusi yang sama dengan metode *Gaussian*

Mixture Model (GMM), yaitu berdistribusi Gaussian dengan asumsi yang sama bahwa data citra MRI terdiri dari beberapa komponen distribusi normal. *Syntax* pemrograman untuk segmentasi dengan metode *hybrid* GMM-SVFMM diperoleh dari penelitian sebelumnya oleh Nikou, Galatsanos, dan Likas (2007) dimana sebelumnya telah dilakukan modifikasi pada tahap inisialisasi parameter. Berdasarkan hasil validasi *cluster* yang dilakukan sebelumnya, untuk *hybrid* GMM-SVFMM jumlah *cluster* terbaik untuk segmentasi citra MRI adalah sebanyak 5 *cluster*. Algoritma yang digunakan untuk mengestimasi parameter *hybrid* GMM-SVFMM juga sama dengan pada metode GMM yaitu algoritma *Expectation-Maximization* (EM). Hasil segmentasi citra dengan metode GMM menggunakan algoritma EM ditampilkan pada Gambar 4.21.



Gambar 4.21 Citra Hasil Segmentasi dengan *Hybrid* GMM-SVFMM (a) Citra *Preprocessing* (b) Citra Hasil Segmentasi *Hybrid* GMM-SVFMM 5 *Cluster*

Berdasarkan Gambar 4.21 (a) dapat dilihat citra masukan untuk segmentasi dengan *hybrid* GMM-SVFMM dari hasil *preprocessing*, dan Gambar 4.21 (b) menunjukkan hasil segmentasi SVFMM dengan 5 *cluster* yang menghasilkan segmentasi area tumor otak dengan indikasi ROI berwarna abu-abu. Sementara itu, bentuk distribusi data dari hasil segmentasi SVFMM dengan 5 *cluster* dapat dilihat melalui Gambar 4.22.



Gambar 4.22 Histogram Hasil Segmentasi Citra menggunakan *Hybrid GMM-SVFMM-5 cluster*

Dapat dilihat pada histogram Gambar 4.22, diduga terdapat 5 puncak untuk data hasil segmentasi. Selanjutnya, estimasi parameter-parameter *hybrid GMM-SVFMM* dengan 5 komponen distribusi diperoleh dengan bantuan program ditampilkan pada Tabel 4.2.

Tabel 4.2 Estimasi Parameter Model Segmentasi *Hybrid GMM-SVFMM 5 Cluster*

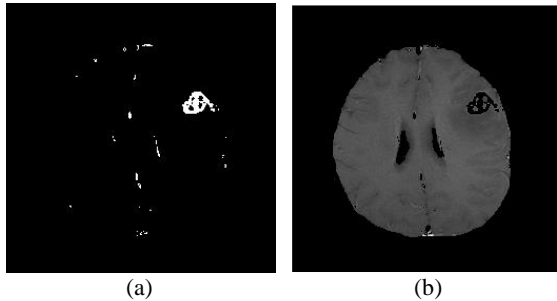
<i>Cluster</i>	$\boldsymbol{\pi}$	$\boldsymbol{\mu}$	$\boldsymbol{\Sigma}$
1	$\pi_{11}, \pi_{21}, \dots, \pi_{N1}$	0	0
2	$\pi_{12}, \pi_{22}, \dots, \pi_{N2}$	80,7825	54,4523
3	$\pi_{13}, \pi_{23}, \dots, \pi_{N3}$	62,5230	231,6030
4	$\pi_{14}, \pi_{24}, \dots, \pi_{N4}$	111,3844	66,0721
5	$\pi_{15}, \pi_{25}, \dots, \pi_{N5}$	111,3210	3976,1

dimana fungsi densitasnya dapat dituliskan pada persamaan (4.2).

Berdasarkan persamaan (4.2), dapat dilihat keenam distribusi normal yang menyusun model segmentasi citra MRI. Dengan bantuan penilaian subjektif dari pihak medis, dapat ditentukan distribusi dari area tumor otak atau ROI (*Region of Interest*), yaitu $N(y|111,3844;66,0721)$. Sementara itu, keempat komponen distribusi lainnya $N(y|0;0)$, $N(y|80,7825;54,4523)$, $N(y|62,523;231,6030)$, dan $N(y|111,3210;3976,1)$ merupakan bagian otak yang bukan tumor atau Non-ROI (*Non-Region of Interest*).

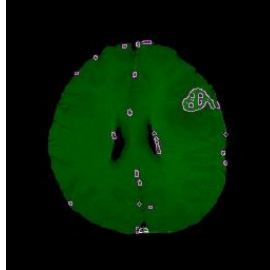
$$\begin{aligned}
f(\mathbf{y} | \boldsymbol{\mu}, \boldsymbol{\Sigma}, \boldsymbol{\pi}) &= \sum_{j=1}^5 \pi_{ij} f(y | \mu_j, \Sigma_j) \\
&= \pi_{i1} N(y | \mu_1, \sigma_1) + \pi_{i2} N(y | \mu_2, \sigma_2) \\
&\quad + \pi_{i3} N(y | \mu_3, \sigma_3) + \pi_{i4} N(y | \mu_4, \sigma_4) \\
&\quad + \pi_{i5} N(y | \mu_5, \sigma_5) \\
&= \pi_{i1} \frac{1}{\sqrt{2\pi}0} \exp\left(-\frac{1}{2} \frac{(y_i - 0)^2}{0}\right) \\
&\quad + \pi_{i2} \frac{1}{\sqrt{2\pi}54,4523} \exp\left(-\frac{1}{2} \frac{(y_i - 80,7825)^2}{54,4523}\right) \\
&\quad + \pi_{i3} \frac{1}{\sqrt{2\pi}231,6030} \exp\left(-\frac{1}{2} \frac{(y_i - 62,523)^2}{231,6030}\right) \\
&\quad + \pi_{i4} \frac{1}{\sqrt{2\pi}66,0721} \exp\left(-\frac{1}{2} \frac{(y_i - 111,3844)^2}{66,0721}\right) \quad (4.2) \\
&\quad + \pi_{i5} \frac{1}{\sqrt{2\pi}3976,1} \exp\left(-\frac{1}{2} \frac{(y_i - 111,3210)^2}{3976,1}\right)
\end{aligned}$$

Sementara itu untuk parameter label *cluster* π_{ij} tidak dituliskan satu-persatu estimasi parameternya karena terdapat $5 \times 65.536 = 327.680$ parameter proporsi. Untuk melihat dengan jelas area ROI dan Non-ROI, hasil segmentasi dapat disajikan pada Gambar 4.23.



Gambar 4.23 Citra Hasil Segmentasi *Hybrid GMM-SVFMM-5 Cluster* (a) Area ROI
(b) Area Non-ROI

Pada Gambar 4.23, menunjukkan area yang diduga menjadi ROI tumor otak (a) dan area otak yang bukan tumor atau Non-ROI (b). Hasil segmentasi dengan *Hybrid GMM-SVFMM* ini lebih baik bila dibandingkan dengan hasil segmentasi dengan *GMM*, hal ini diindikasikan dengan area tumor otak yang lebih *solid* dan *noise* yang lebih. Kemudian, untuk mendeteksi perubahan intensitas pada area ROI dilakukan *subjective analysis* dengan deteksi tepi.



Gambar 4.24 *Subjective Analysis Hybrid GMM-SVFMM ROI Citra MRI Sequence ax T1 memp+C*

Berdasarkan Gambar 4.24, dapat terlihat tepi-tepi yang menjadi area ROI tumor otak. Seperti pada segmentasi dengan *GMM*, *subjective analysis* dilakukan dengan tujuan untuk membantu ahli medis memberi penilaian secara subjektif dalam menentukan area ROI tumor otak.

4.4 Perbandingan Metode *Gaussian Mixture Model (GMM)* dan *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)*

Hasil segmentasi dengan metode GMM dan *Hybrid GMM-SVFMM* menunjukkan terdapat perbedaan area ROI tumor otak. Dalam mengevaluasi model segmentasi yang telah dilakukan dengan metode GMM dan *hybrid GMM-SVFMM*, akan dibandingkan nilai *correct classification ratio (CCR)*. Digunakannya CCR sebagai ukuran evaluasi hasil segmentasi ROI tumor otak, karena tumor otak pada pasien yang MRI-nya digunakan dalam penelitian ini telah ditangani di RSUD Dr. Soetomo Surabaya, sehingga dapat didefinisikan *target/ground truth* yang menjadi ROI tumor otak. Untuk melakukan perbandingan lebih jelas secara komputasi antara kedua metode, dapat dilihat pada Tabel 4.3.

Tabel 4.3 Perbandingan Metode GMM dan *Hybrid GMM-SVFMM* dengan Algoritma EM

Metode	Cluster	Banyak Parameter	Running Time	CCR	Tingkat Kesulitan
GMM	3	9	8,334 s	0,8735	Sederhana
<i>Hybrid GMM-SVFMM</i>	5	327.680	45,399 s	0,9876	Kompleks

Berdasarkan Tabel 4.3, dapat disimpulkan bahwa dengan menggunakan perangkat keras yang sama yaitu prosesor Intel (R) Core (TM) i5-4200U CPU @ 1.60GHz 1.6 GHz dengan RAM 6.00GB dan menggunakan algoritma EM, metode *hybrid GMM-SVFMM* menghasilkan segmentasi ROI tumor otak yang lebih baik dibandingkan dengan metode GMM. Hal ini diindikasikan dari nilai evaluasi ROI hasil segmentasi menggunakan CCR yaitu sebesar 0,8735 untuk GMM dan 0,9876 untuk *hybrid GMM-SVFMM*. Meskipun demikian, segmentasi dengan *hybrid GMM-SVFMM* cukup kompleks dari segi komputasi, karena memakan waktu *running program* yang lebih lama dibandingkan dengan GMM, yaitu 8,334 detik untuk metode GMM dan 45,399 detik untuk metode *hybrid GMM-SVFMM*. Disamping itu, banyaknya parameter yang diestimasi juga berkontribusi dalam kompleksitas segi komputasi, dimana metode

GMM hanya 9 parameter yang diestimasi sedangkan metode *hybrid* GMM-SVFMM membutuhkan 327.680 parameter yang diestimasi.

BAB V KESIMPULAN DAN SARAN

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

5.1 Kesimpulan

Berdasarkan implementasi, uji coba, analisis dan pembahasan yang telah dilakukan, terdapat beberapa kesimpulan yang didapatkan dari penelitian ini yaitu sebagai berikut. Setelah dilakukan *preprocessing*, pada data citra MRI *sequence ax T1 memp+C* terjadi perubahan kualitas citra MRI yang lebih baik. Hal ini diindikasikan dengan tampilan citra yang lebih jelas, serta bentuk histogram menjadi lebih baik. Selanjutnya pada validasi jumlah *cluster* terbaik berdasarkan nilai *Correct Classification Ratio* (CCR) dan penilaian subjektif dari ahli medis, untuk metode GMM dengan algoritma EM didapatkan jumlah *cluster* sebanyak 3 *cluster* dan untuk metode *hybrid* GMM-SVFMM dengan algoritma EM didapatkan jumlah *cluster* sebanyak 5 *cluster*. Evaluasi hasil segmentasi melalui nilai CCR juga menunjukkan bahwa hasil segmentasi dengan metode *hybrid* GMM-SVFMM lebih baik dibandingkan dengan metode GMM. Hal ini diindikasikan dengan nilai CCR untuk metode *hybrid* GMM-SVFMM sebesar 0,9876, dimana nilai ini lebih besar dibandingkan dengan nilai CCR untuk metode GMM yaitu sebesar 0,8735.

Apabila kedua metode dibandingkan dari segi komputasi, metode GMM lebih mudah diterapkan dibandingkan dengan metode *hybrid* GMM-SVFMM, karena algoritma GMM yang lebih sederhana dibandingkan dengan algoritma untuk metode *hybrid* GMM-SVFMM yang cukup kompleks. Segmentasi citra MRI untuk mendeteksi tumor otak dengan GMM membutuhkan *running time* selama 8,334 detik, sementara itu metode *hybrid* GMM-SVFMM membutuhkan *running time* selama 45,399 detik. Disamping itu, kompleksitas dari metode *hybrid* GMM-SVFMM juga disebabkan oleh banyaknya parameter yang perlu di estimasi yaitu sebanyak 327.680 parameter untuk *hybrid* GMM-SVFMM dan sebanyak 9 parameter untuk GMM.

5.2 Saran

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

1. Menggunakan pendekatan Bayesian pada estimasi parameternya sehingga informasi prior pada data citra dapat dipertimbangkan sepenuhnya.
2. Menggunakan metode mixture yang bentuk distribusinya mengikuti pola distribusi pada citra asli.
3. Menggunakan implementasi IDE lainnya yang lebih cepat dan ringan.
4. Menganalisa *sequence/slice* lainnya pada MRI kepala sehingga dapat memberikan visualisasi tumor yang lebih bervariasi.

DAFTAR PUSTAKA

- ABTA. (2017). American Brain Tumor Association [Online]. <http://www.abta.org/about-us/news/brain-tumor-statistics/>, Accessed 29 Januari 2018.
- Adipranata, R. (2005). *Kombinasi Metyode Morphological Gradient Dan Transformasi Watershed Pada Proses Segmentasi Citra Digital*. Surabaya: Universitas Kristen Petra.
- Anifa, A., Mukid, M. A., & Rusgiyono, A. (2012). Simulasi Stokastik Menggunakan Algoritma Gibbs Sampling. *Jurnal Gaussian, Volume 1, Nomor 1*, 21-30.
- Barbu, T. (2013). Variational Image Denoising Approach with Diffusion Porous Media Flow. *Abstract and Applied Analysis*.
- Besag, J. (1986). On the statistical analysis of dirty pictures., *Journal of the Royal Statistical Society, Series B, vol. 48, no. 3*, 259-302.
- Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.
- Blekas, K., Likas, A., Galatsanos, N. P., & Lagaris, I. E. (2005). A Spatially Constrained Mixture Model for Image Segmentation. *IEEE Transactions On Neural Networks, Vol. 16, No. 2*, 494-498.
- Bouchet, A., Pastore, J., & Ballariv, V. (2007). Segmentation of Medical Images using Fuzzy Mathematical Morphology. *JCS and T, Vol.7, No.3*, 256-262.
- Carlin, B. P., & Louis, T. A. (2000). *Bayes and Empirical Bayes Methods for Data Analysis (2 ed.)*. Florida: Chapman and Hall/CRC Press.
- Casella, G., & George, E. I. (1992). Explaining the Gibbs Sampler. *The American Statistician, Vol. 46, No. 3*, 167-174.
- Cucala, L., & Marin, J.-M. (2013). Bayesian Inference on A Mixture Model with Spatial Dependence. *Journal of Computational and Graphical Statistics*, 584-597.
- Dempster, A. P., Laird, N. M., & Rubin, D. P. (1977). Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society. Series B, Vol. 39, No. 1*.

- 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). *Bayesian Data Analysis, Third Edition*. New York: CRC Press.
- Gonzalez, R. C., & Wood, R. E. (2008). *Digital Image Processing, 3rd Edition*. New Jersey: Pearson Prentice Hall.
- Iriawan, N. (2001). Penaksiran Model Mixture Normal Univariabel: Suatu Pendekatan Metode Bayesian dengan MCMC. *Seminar Nasional dan Konferda VII Matematika Wilayah DIY dan Jawa Tengah*. Yogyakarta.
- Iriawan, N. (2011). Pemodelan Mixture of Mixture dalam Pemilihan Portofolio. *Prosiding Seminar Nasional Statistika Universitas Diponegoro*.
- Iriawan, N., Astutik, S., & Presetyo, D. D. (2010). Markov Chain Monte Carlo – Based Approaches for Autoregressive (CAR) Frailty. *International Journal of Computer Science and Network Security, Vol.10 No.12*.
- Ji, Z., Huang, Y., Sun, Q., & Cao, G. (2016). A Spatially Constrained Generative Asymmetric Gaussian Mixture Model for Image Segmentation. *Journal of Visual Communication and Image Representation*.
- Kato, Z., & Zerubia, J. (2011). Markov Random Fields in Image Segmentation. *Foundations and Trends in Signal Processing Vol. 5 No. 1-2*, 1-155.
- Lee, J. M., Yoon, U., Nam, S. H., Kim, J. H., Kim, I. Y., & Kim, S. I. (2003). Evaluation of Automated and Semi-automated Skull-stripping Algorithms Using Similarity Index and Segmentation Error. *Computers in Biology and Medicine, Vol. 33, No. 6*, 495-507.
- McLachlan, G. J., & Basford, K. E. (1988). *Mixture Models: Inference and Applications to Clustering*. New York: John Wiley and Sons.
- McLachlan, G. J., & Peel, D. (2000). *Finite Mixture Models, Wiley Series in Probability and Statistics*. New York: John Wiley & Sons Inc.

- Middleton, I., & Damber, R. I. (2004). Segmentation of Magnetic Resonance Images Using A Combination of Neural Networks and Active Contour Models. *Medical Engineering and Physics*, Vol.26, 71-86.
- Munir, R. (2004). *Pengolahan Citra Digital Dengan Pendekatan Algoritmik*. Bandung: Informatika.
- Murinto. (2012). *Studi dan Implementasi Klasifikasi Data Citra Satelit Menggunakan Model Markov Random Field*. Yogyakarta: Universitas Ahmad Dahlan.
- Nguyen, T. M. (2011). *Gaussian Mixture Model based Spatial Information Concept for Image Segmentation*. Ontario: University of Windsor.
- 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.
- Nikou, C., Galatsanos, N., & Likas, A. (2007). A Class-Adaptive Spatially Variant Mixture Model for Image Segmentation. *IEEE Transactions on Image Processing*, vol. 16, no. 4, 1121-1130.
- Notosiswoyo, M., & Suswati, S. (2004). Pemanfaatan Magnetic Resonance Imaging (MRI) Sebagai Sarana Diagnosa Pasien. *Media Litbang Kesehatan Volume XIV Nomor 3*.
- Ntzoufras, I. (2009). *Bayesian Modelling using WinBUGS*. New Jersey: John Wiley and Sons, Inc.
- Prasetyo, E. (2011). *Pengolahan Citra Digital dan Aplikasinya Menggunakan MATLAB*. Yogyakarta: Andi.
- Sanjay, G. S., & Hebert, T. J. (1988). Bayesian Pixel Classification Using Spatially Variant Finite Mixtures and The Generalized EM Algorithm. *IEEE Trans. Image Process*, vol.7, no.7, 1014-1028.
- 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: Institut Teknologi Sepuluh Nopember.

- Sonka, M., Hlavac, V., & Boyle, R. (1993). *Image Processing, Analysis, and Machine Vision.* New York: Chapman & Hall Computing - London.
- Stark, D. D. (1988). *Magnetic Resonance Imaging.* Toronto: The CV Mosby Company.
- Tan, P.-N., Steinbach, M., & Kumar, V. (2006). *Introduction to Data Mining.* Boston: Pearson.
- Wahjoepramono. (2006). *Tumor Otak.* Jakarta: Subur Jaringan Cetak Terpadu.
- Walsh, B. (2004). Markov Chain Monte Carlo and Gibbs Sampling. *Lecture Notes for EEB 581.*
- Xu, R., & Wunsch, D. C. (2008). *Clustering.* New Jersey: John Wiley & Sons, Inc.
- Zhang, Y., Brady, M., & Smith, S. (2001). Segmentation of Brain MR Images Through A Hidden Markov Random Field Model and The Expectation–Maximization Algorithm. *IEEE Trans. Med. Imag., vol. 20, no. 1, 45-57.*

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(prepare, [])
restore = mat2gray(prepare);
imshow(restore);
I2 = imcrop(restore, [75 50 360 400]);
imshow(I2)
    imwrite(restore, 'outputPrep.jpg', 'Quality', 100, '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(prepare)

```



```
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, 'Mode', 'lossy');
imwrite(substract, 'substract.jpg', 'Quality', 100, 'Mode', 'lossy')
imwrite(openingIn, 'openingInitial.jpg', 'Quality', 100, 'Mode', 'lossy');
imwrite(eDetection, 'edgeDetection.jpg', 'Quality', 100, '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
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 *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization* (EM)

```
function [mu, covar, p, z, u] =
gaussianMixDCASV(X, K, imageSize, groundTruth)
% [m, covar, w, z, u] = gaussianMixDCASV(X,
maxSegments,
%
%                               imageSize
%
%      | 2 |
%     --/  \--
%      1  x  3      <-- Neighbourhood indexing
%     --\  /--
%      | 4 |
% G.Sfikas 26 Nov 2007
% Updates 01 Feb 2007

D = size(X, 1);
N = size(X, 2);
NhooSize = 4;
NhooDirections = 2; %WARNING:Unreliable constant
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
% Initialization
[mu p junk covar] = EMSeg3(X, K);
p = p'*ones(1, N);
p = (p ./ (ones(K, 1) * (sum(p, 1)+eps)))';
beta = double(0.8 * ones(NhoodDirections, K));
u = double(ones(NhoodSize, K, N));
%% Hyperparameter initialization
z = inv(N) * ones(N, K);
% newWeight = inv(N) * ones(N, K);
wDiff2 = zeros(NhoodSize, K, N);

% disp('gaussian Mix: on labels. ');
% w = inv(K) * ones(size(w));
% EM iterations...
disp('EM iteration Likelihood Average
Logl LhoodIncrease RandIndex
RandIndexContextual');
disp('-----
-----
--');
likelihood = 0;
for iterations = 1:15
    if mod(iterations, 229) == 0
        fprintf(' ');
    end
    prev = likelihood;
    likelihood = 0;

    % 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));
```


Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
%logg(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
    %% Special -- reinitialize w to the posterior
if iterations == 1
%       im=zeros(imageSize);
%       for kk=1:K
%           im(:, :) = reshape(z(:, kk),
imageSize);
%           xx=filter2((1./9.)*ones(3,3),im);
%           z(:, kk)=xx(:);
%       end
%       w = z;
    %%%
%       if exist('groundTruth', 'var') &&
isempty(groundTruth) == 0
%           disp('gaussianMix: Cheat!
Initializing using ground truth!');
%           z = zeros(N, K);
%           for j = 1:K
%               z(groundTruth == j, j) = 1;
%           end
%           w = z;
    end
end
end
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```

% 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
    clear newCovar;
end
% Some convenient statistics, regarding w.
%Neighbour square differences: wDiff2
for j = 1:K
    for n = 1:NhoodSize
        [pos direction] = getNeighbourInfo(n);
        temp = translation(reshape(p(:, j),
imageSize), -pos);
        temp2 = translation(temp, +pos);
        wDiff2(n, j, :) = (temp(:) -
temp2(:)).^2;
        likelihood = likelihood + ...
            sum(logGaussianValue((temp(:) -
temp2(:))', 0, beta(direction, j)));
    end
end
llChangeRatio = (likelihood - prev) /
abs(prev);

```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
fprintf('%3d           %3.2f           %2.5f
%2.5f%%           %2.4f           %2.4f\n', ...
        iterations, likelihood, likelihood/N,
        llChangeRatio*100, currentRI, currentRICont);

% STEP 2: Maximize parameters: beta, v (freedom
degrees) and pi
%% Pi (weights) %%
for ii = 1:2
    for j = 1:K
        aQuad = 0; bQuad = 0;
        cQuad = -0.5 * z(:, j);
        for k = 1:NhoodSize
            [pos d] = getNeighbourInfo(k);
            temp = translation(reshape(p(:, j),
imageSize), -pos);
            aQuad = aQuad + inv(beta(d, j) + eps)
* squeeze(u(k, j, :));
            bQuad = bQuad - inv(beta(d, j) + eps)
* ...
                (squeeze(u(k, j, :)) .* temp(:));
        end
        newWeight(:, j) = solveQuad(aQuad, bQuad,
cQuad);
    end
    for n = 1:N
        p(n, :) = BIDProjection(newWeight(n, :));
    end
end
%     for n1 = 1:imageSize(1)
%     for n2 = 1:imageSize(2)
%         n = (n2-1)*imageSize(1) + n1;
%         for j = 1:K
%             aQuad = 0; bQuad = 0;
%             cQuad = -0.5 * z(n, j);
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
%           for k = 1:NhoodSize
%           [pos d] = getNeighbourInfo(k);
%           neighPos = [n1+pos(1)
n2+pos(2)];
%           if neighPos(1) < 1 ||
neighPos(1) > imageSize(1) || ...
%           neighPos(2) < 1 ||
neighPos(2) > imageSize(2)
%           continue;
%           end
%           neighPos = (neighPos(2) -
1)*imageSize(1) + neighPos(1);
%           aQuad = aQuad + inv(beta(d, j) +
eps) * squeeze(u(k, j, n));
%           bQuad = bQuad - inv(beta(d, j) +
eps) * ...
%           (squeeze(u(k, j, n)) .*
w(neighPos, j));
%           end
%           newWeight(j) = solveQuad(aQuad,
bQuad, cQuad);
%           end
%           w(n, :) = BIDProjection(newWeight);
%           end
%           end
%% Beta %%
for j = 1:K
    betaComponent = zeros(1, NhoodSize);
    for n = 1:NhoodSize
        betaComponent(n) = (sum(u(n, j, :) .*
wDiff2(n, j, :))) / ...

getTotalNeighbours(getNeighbourInfo(n),
imageSize);
    end
    for d = 1:size(beta, 1)
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
        beta(d, j) =
sum(betaComponent(getDirectionInfo(d)));
    end
    clear betaComponent;
end
end
p = p';
z = z';
return;

%%
function [pos direction] = getNeighbourInfo(n)
switch n
    case {1}
        pos = [0 -1];
        direction = 1;
    case {2}
        pos = [-1 0];
        direction = 2;
    case {3}
        pos = [0 1];
        direction = 1;
    case {4}
        pos = [1 0];
        direction = 2;
    otherwise
        disp('Error: Unknown neighbour');
end
return;

%%
function res = getDirectionInfo(d)
switch d
    case {1}
        res = [1 3];
    case {2}
        res = [2 4];
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
        otherwise
            disp('Error: Unknown direction');
    end
    return;

%%
function res = getTotalNeighbours(offset,
    imageSize)
res = prod(imageSize - abs(offset));
return;

%%
function res = solveQuad(a, b, c)
if a < 0
    a = -a;
    b = -b;
    c = -c;
end
determining = b.^2 - 4 * a .* c;
res = (-b + sqrt(determining)) ./ (2*a);
return;

%%
function res = gaussianValue(X, m, covar)
% Computes probability of dataset X (each _column_
% is one datum),
% given a gaussian pdf, Normal(m, covar).
%
% Example:
%         gaussianValue([4 3 2; 2 3 4], [3 3]',
eye(2))
%         will return [0.0585 0.1592 0.0585]'.
%
% G.Sfikas 7 feb 2007.
N = size(X, 2);
d = size(X, 1);
bigM = m * ones(1, N);
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
mah = sum((X-bigM)' *
(inv(covar)+eps*eye(size(X,1)))) .* (X-bigM)', 2);
res = (2*pi)^(-0.5*d) * det(covar)^(-0.5) * exp(-
0.5*mah);
return;

%%
function res = logGaussianValue(X, m, covar)
% Computes log-probability of dataset X (each
_ column_ is one datum),
% given a gaussian pdf, Normal(m, covar).
%
% See also:
%     gaussianValue, studentValue
%
% G.Sfikas 4 feb 2008.
N = size(X, 2);
d = size(X, 1);
bigM = m * ones(1, N);
mah = sum((X-bigM)' *
(inv(covar)+eps*eye(size(X,1)))) .* (X-bigM)', 2);
res = -(0.5*d)*log(2*pi) -0.5*logdet(covar+eps) -
0.5*mah;
return;

%%
function res = translation(rect, offset, filltype)
% res = translation(rect, offset, filltype)
%
% Translate (move) the matrix 'rect' _plus_ the
'offset'.
%
% rect                Input matrix.
% offset              Vector of
directions to move 'rect'.
% filltype            {'fill'}      Zero-padding.
```

Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
%           'replicate'       Padding using
border values.
%
% Examples
%     translation([1 2 3; 4 5 6; 7 8 9], [1 0])
will produce
%           [0 0 0;
%           1 2 3;
%           4 5 6]
%     while
%     translation([1 2 3; 4 5 6; 7 8 9], [1
0], 'replicate') will pad like
%           [1 2 3;
%           1 2 3;
%           4 5 6]
%
% G.Sfikas 15 Feb 2008
if nargin < 3
    filltype = 'fill';
end
T = maketform('affine', [eye(2) [offset(1)
offset(2)]]; 0 0 1]);
R = makesampler('nearest', filltype);
res = tformarray(rect, T, R, [1 2], [1 2],
size(rect), [], []);
return;

%%
function res = BIDProjection(x, w)
%
% BIDProjection(vector)
% Projects input vector onto constraint plane
defined by
% 1. sum(vector) = 1
% 2. Each variate of 'vector' >= 0
% vector must 1xK sized.
%
```


Lampiran 6 *Syntax* Segmentasi Metode *Hybrid Gaussian Mixture Model – Spatially Variant Finite Mixture Model (GMM-SVFMM)* dengan Algoritma *Expectation-Maximization (EM)* (lanjutan)

```
% If a 2nd argument 'weights' is defined, then the
ConjugateProjection
% function is called. See also there for details.
%
% ***CET VERSION EST UN WRAPPER QUI APPELERA LA
VERSION "MEX" DE CET FICHE.***
% G.Sfikas 22 Mar 2008
% based on code by K.Blekas
%
% Update 21 Apr 2008
if nargin == 1
%     res = MEX_BIDProjection(x, numel(x));
%     return;
    w = ones(size(x));
end
res = MEX_ConjugateProjection(x, w, numel(x));
return;
%%
```

Lampiran 7 *Syntax* Function *MEX Conjugate Projection C++*

```
// res = conjugateProjection(x, weights)
//
// Project Kx1 vector weights .* x to hyperplane
// defined by weights'*y = 1, subject
// to the constraint sum(y) = 1, y >= 0.
// The objective function is supposed to be of
the form
//     x'Ax + bx + const.
// A is diagonal, and 'weights' are its elements.
//
// Notes: 1. If weights = ones(1,K), then the
result should
//     coincide to K.Blekas'
BIDProjection(x).
// Notes: 2. In the CVPR08 submission, the
objective function was
```

Lampiran 7 Syntax Function MEX Conjugate Projection C++ (lanjutan)

```
//          in fact of the form  $x'Ax + bx +$   
clogx + const.  
//          In BLEKAS05, the objective function  
function was  
//           $x'x + bx + clogx + const.$   
// G.Sfikas 21 April 2008  
#define SQR(x) ((x)*(x))  
#include "mex.h"  
#include <stdlib.h>  
#include <math.h>  
#define MAXK 100  
#define MALAKA printf("HOHO\n");  
double K;  
  
void ConjProjection(double *result, double *a,  
double *weights);  
void findMaximum(double *maxValue, int *index,  
double x[]);  
  
void mexFunction(int nlhs, mxArray *plhs[], int  
nrhs, const mxArray *prhs[])  
{  
    double *x;  
    double *weights;  
    double *result;  
    double *sizeX;  
    /* Input*/  
    x = mxGetPr(prhs[0]);  
    weights = mxGetPr(prhs[1]);  
    sizeX = mxGetPr(prhs[2]);  
    if(x == NULL || sizeX == NULL) {  
        printf("ConjProjection: Syntax is  
MEX_ConjuageProjection(vector, weights, number of  
elements in vector)\n");  
        printf("ConjProjection: Example:  
MEX_ConjugateProjection([1 1.5 -2.1], [1 1 1],  
3)\n");  
    }  
}
```

Lampiran 7 Syntax Function MEX Conjugate Projection C++ (lanjutan)

```
K = *sizeX;
/* Output*/
plhs[0] = mxCreateDoubleMatrix(1, K, mxREAL);
result = mxGetPr(plhs[0]);
ConjProjection(result, x, weights);
}
void ConjProjection(double *result, double *a,
double *weights)
{
    double g[MAXK];
    double yInit[MAXK];
    double y[MAXK];
    int activeSet[MAXK];
    double sumW;
    double sumA;
    double sumL;
    int j;
    int negCount;
    double sumGINACT, sumGACTY;
    int iterations;
    double lambda[MAXK];
    double ggM; int ggExcl;
    double sumAbsX;

    /* Check if vector has too large variates */
    findMaximum(&ggM, &ggExcl, a);
    for(sumAbsX = 0, j = 0; j < K; j++)
        sumAbsX += fabs(a[j]);
    if(ggM > 0 && sumAbsX - ggM < 1e-5) {
        for(j = 0; j < K; j++)
            result[j] = 0;
        result[ggExcl] = 1;
        return;
    }

    for(sumW = 0, sumA = 0, j = 0; j < K; j++) {
        sumW += 1/weights[j];
        sumA += a[j];
    }
}
```

Lampiran 7 Syntax Function MEX Conjugate Projection C++ (lanjutan)

```
    }
    for(j = 0; j < K; j++)
        g[j] = -1/(sumW*weights[j]);
    //Initial projection to weights'y = 1.
    for(j = 0; j < K; j++) {
        y[j] = a[j] - g[j] + g[j]*sumA;
        yInit[j] = y[j];
    }
    // activeSet = zeros(K, 1);
    for(j = 0; j < K; j++)
        activeSet[j] = 0;
    ////////////////////////////////////
    // main loop
    ////////////////////////////////////
    for(iterations = 0; iterations < K; j++) {
        for(negCount = 0, j = 0; j < K; j++)
            if(y[j] < 0) {
                negCount++;
                activeSet[j] = 1;
            }
        if(negCount == 0) {
            for(j = 0; j < K; j++)
                result[j] = y[j];
            return;
        }
        for(sumGACTY = 0, sumGINACT = 0, j = 0; j
< K; j++) {
            if(activeSet[j] == 1)
                sumGACTY += g[j]*yInit[j];
            else
                sumGINACT += g[j];
            lambda[j] = 0;
        }
        // Iterate between recalculating Lagrange
        operators & proj to sum(y) = 1
        for(sumL = 0, j = 0; j < K; j++) {
            if(activeSet[j] == 0)
                lambda[j] = 0;
```


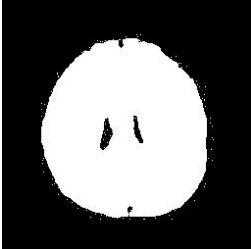
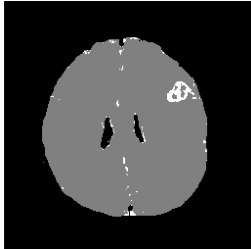

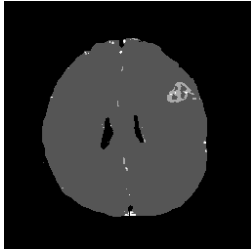
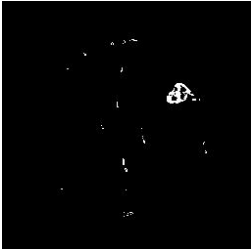
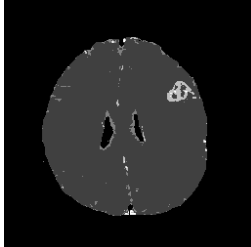

Lampiran 7 *Syntax Function MEX Conjugate Projection C++*
(lanjutan)

```
        else
            lambda[j] = -yInit[j] - sumGACTY /
sumGINACT;
        sumL += lambda[j];
    }
    for(j = 0; j < K; j++) {
        if(activeSet[j] == 0)
            y[j] = a[j] - g[j] + g[j]*sumA +
g[j]*sumL + lambda[j];
        else
            y[j] = 0;
    }
    }
    printf("conjugateProjection: Algorithm failed
after K = %f iterations!\n", K);
}

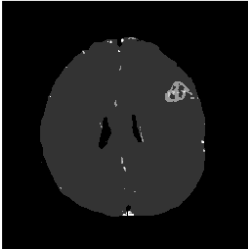

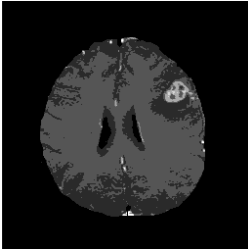

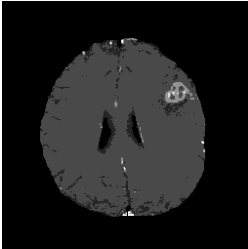
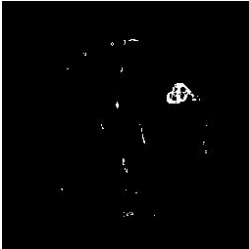
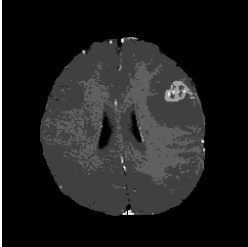
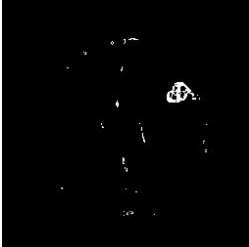
void findMaximum(double *maxValue, int *index,
double x[]) {
    int i;

    *maxValue = x[0]; *index = 0;
    for(i = 0; i < K; i++)
        if(x[i] > *maxValue) {
            *maxValue = x[i];
            *index = i;
        }
    return;
}
```

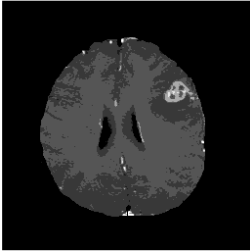
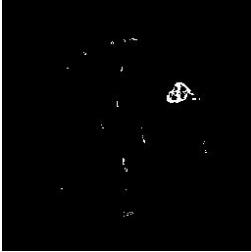
Lampiran 8 Hasil Segmentasi Citra *Sequence ax T1 memp+C* dengan Metode GMM

<i>Cluster</i>	Hasil Segmentasi	ROI
2		
3		
4		
5		

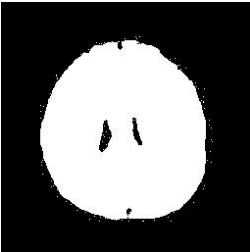
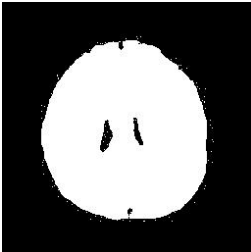
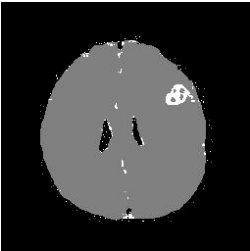
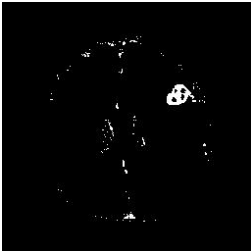
Lampiran 9 Hasil Segmentasi Citra *Sequence ax T1 memp+C* dengan Metode GMM (lanjutan)

<i>Cluster</i>	Hasil Segmentasi	ROI
6	 An axial T1-weighted MRI slice of a brain. The image shows a cross-section with a dark background. A small, bright, irregularly shaped region is highlighted in the upper right quadrant, representing the segmented area for cluster 6.	 A binary image showing the Region of Interest (ROI) for cluster 6. The background is black, and the segmented region from the previous image is shown in white.
7	 An axial T1-weighted MRI slice of a brain. The image shows a cross-section with a dark background. A small, bright, irregularly shaped region is highlighted in the upper right quadrant, representing the segmented area for cluster 7.	 A binary image showing the Region of Interest (ROI) for cluster 7. The background is black, and the segmented region from the previous image is shown in white.
8	 An axial T1-weighted MRI slice of a brain. The image shows a cross-section with a dark background. A small, bright, irregularly shaped region is highlighted in the upper right quadrant, representing the segmented area for cluster 8.	 A binary image showing the Region of Interest (ROI) for cluster 8. The background is black, and the segmented region from the previous image is shown in white.
9	 An axial T1-weighted MRI slice of a brain. The image shows a cross-section with a dark background. A small, bright, irregularly shaped region is highlighted in the upper right quadrant, representing the segmented area for cluster 9.	 A binary image showing the Region of Interest (ROI) for cluster 9. The background is black, and the segmented region from the previous image is shown in white.

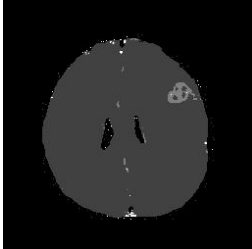

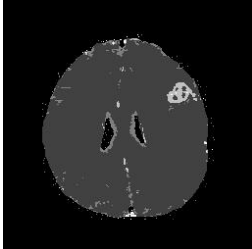

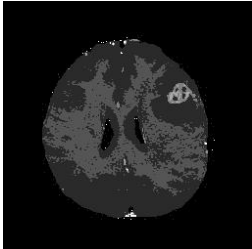
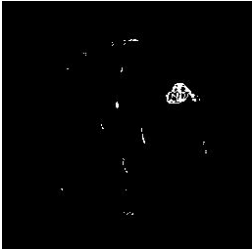
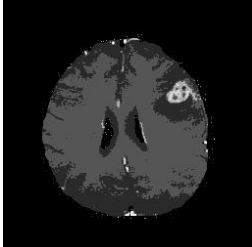
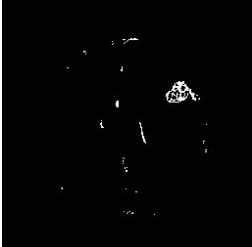
Lampiran 9 Hasil Segmentasi Citra *Sequence ax T1 memp+C* dengan Metode GMM (lanjutan)

<i>Cluster</i>	Hasil Segmentasi	ROI
10		

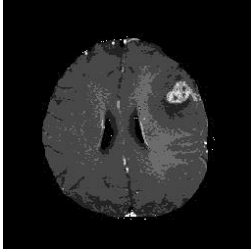
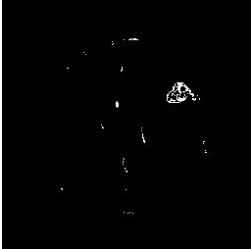
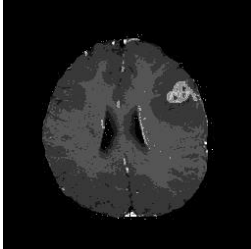
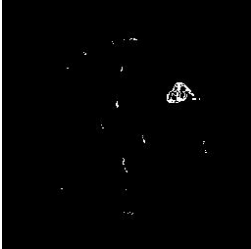
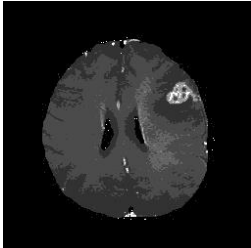
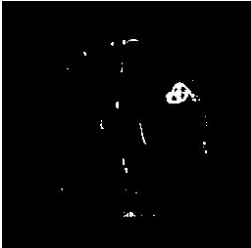
Lampiran 9 Hasil Segmentasi Citra *Sequence ax T1 memp+C* dengan Metode *Hybrid GMM-SVFMM*

<i>Cluster</i>	Hasil Segmentasi	ROI
2		
3		

Lampiran 10 Hasil Segmentasi Citra *Sequence ax T1 memp+C* dengan Metode *Hybrid GMM-SVFMM* (lanjutan)

Cluster	Hasil Segmentasi	ROI
4		
5		
6		
7		

Lampiran 10 Hasil Segmentasi Citra *Sequence ax T1 memp+C* dengan Metode *Hybrid GMM-SVFMM* (lanjutan)

<i>Cluster</i>	Hasil Segmentasi	ROI
8		
9		
10		

Lampiran 10 Nilai *Correct Classification Ratio* (CCR) Metode GMM dan *Hybrid* GMM-SVFMM

Jumlah Cluster	GMM	<i>Hybrid</i> GMM-SVFMM
2	-	-
3	0,8735	0,9783
4	0,7747	0,9537
5	0,8302	0,9876
6	0,7932	0,7839
7	0,679	0,787
8	0,8333	0,608
9	0,7932	0,6605
10	0,7315	0,5062

“Halaman ini sengaja dikosongkan”

Lampiran 11 Surat Pernyataan Pengambilan Data Sekunder

1

BIODATA PENULIS



Penulis bernama Sandra Firda Qonita dilahirkan di Kota Tangerang pada tanggal 13 Februari 1997 sebagai anak pertama dari tiga bersaudara. Penulis merupakan putri pertama dari tiga bersaudara, Ayah Wahjudi Swandaru dan Ibu Ulfa Atin. Penulis menempuh pendidikan formal di SDN Tangerang 6, SMP Negeri 1 Tangerang, dan SMA Negeri 1 Tangerang. Kemudian penulis diterima sebagai Mahasiswa jurusan Statistika FMIPA ITS melalui jalur Mandiri pada tahun 2014 dengan NRP 1314100119. Selama masa perkuliahan, penulis memiliki pengalaman dalam berpartisipasi di berbagai kepanitiaan diantaranya sebagai Anggota Divisi *Sponsorship* dalam *event* tahunan PRS (Pekan Raya Statistika) 2016 dan sebagai moderator *talkshow* dalam acara *Grand Final DAC (Data Analysis Competition)* 2017, yang merupakan kompetisi analisis data tingkat Asia Tenggara. Penulis pernah mengikuti program pertukaran mahasiswa selama satu semester di Dankook University, Korea Selatan, periode Musim Gugur 2016. Selama mengikuti program pertukaran, penulis aktif dalam organisasi sebagai anggota *Young Diplomats of Dankook University* 2016 dan mengikuti program *volunteer* Budaya dan Pendidikan yang diadakan oleh GTN Dankook University. Penulis juga pernah mengikuti pelatihan yaitu LKMM-Pra-TD dan *workshop* yaitu Mandiri Young Leaders 2017. Apabila pembaca ingin memberi kritik dan saran serta diskusi lebih lanjut mengenai Tugas Akhir ini, dapat menghubungi penulis melalui email sandrafirda@gmail.com.

“Halaman ini sengaja dikosongkan”