

**SEGMENTASI *FORAMEN MENTALE* PADA *CITRA DENTAL*  
*PANORAMIC RADIOGRAPH* MENGGUNAKAN  
*THRESHOLDING FUZZY TYPE II***

**SKRIPSI**

Oleh:

**DEVI ARUM SARI**

**NIM : 09650093**



**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM  
MALANG  
2013**

**HALAMAN PENGAJUAN**

**SEGMENTASI *FORAMEN MENTALE* PADA CITRA *DENTAL*  
*PANORAMIC RADIOGRAPH* MENGGUNAKAN  
*THRESHOLDING FUZZY TYPE II***

**SKRIPSI**

**Diajukan kepada:  
Fakultas Sains dan Teknologi  
Universitas Islam Negeri Maulana Malik Ibrahim Malang  
Untuk Memenuhi Salah Satu Persyaratan Dalam Memperoleh  
Gelar Sarjana Komputer (S.Kom)**

**Oleh:  
DEVI ARUM SARI  
NIM: 09650093**

**JURUSAN TEKNIK INFORMATIKA  
FAKULTAS SAINS DAN TEKNOLOGI  
UNIVERSITAS ISLAM NEGERI (UIN) MAULANA MALIK IBRAHIM  
MALANG  
2013**

**HALAMAN PERSETUJUAN**  
**SEGMENTASI FORAMEN MENTALE PADA CITRA DENTAL**  
**PANORAMIC RADIOGRAPH MENGGUNAKAN**  
**THRESHOLDING FUZZY TYPE II**

**SKRIPSI**

Oleh :

Nama : Devi Arum Sari  
NIM : 09650093  
Jurusan : Teknik Informatika  
Fakultas : Sains dan Teknologi

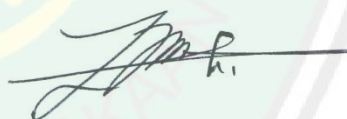
Telah Disetujui, 24 Oktober 2013

**Dosen Pembimbing I**



**Dr. Cahyo Crysdian**  
NIP. 19740424 200901 1 008

**Dosen Pembimbing II**



**Zainal Abidin, M.Kom**  
NIP. 19760613 200501 1 004

Mengetahui,

**Ketua Jurusan Teknik Informatika**



**Dr. Cahyo Crysdian**  
NIP. 19740424 200901 1 008

**HALAMAN PENGESAHAN**  
**SEGMENTASI FORAMEN MENTALE PADA CITRA DENTAL**  
**PANORAMIC RADIOGRAPH MENGGUNAKAN**  
**THRESHOLDING FUZZY TYPE II**

**SKRIPSI**

**Devi Arum Sari**

**NIM. 09650093**





Telah Dipertahankan Di Depan Dewan Penguji Skripsi  
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan  
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)

Tanggal, 14 November 2013

**Susunan Dewan Penguji :**

- |                           |   |
|---------------------------|---|
| <b>1. Penguji Utama</b>   | <b>: <u>Dr. Muhammad Faisal, M.T</u></b><br><b>19740510 200501 1 007</b>  |
| <b>2. Ketua Penguji</b>   | <b>: <u>Irwan Budi Santoso, M.Kom</u></b><br><b>19770103 201101 1 004</b> |
| <b>3. Sekretaris</b>      | <b>: <u>Dr. Cahyo Crysdiان</u></b><br><b>19740424 200901 1 008</b>        |
| <b>4. Anggota Penguji</b> | <b>: <u>Zainal Abidin, M.Kom</u></b><br><b>19760613 200501 1 004</b>      |

**Tanda Tangan**

(  )  
(  )  
(  )  
(  )

Mengetahui,

**Ketua Jurusan Teknik Informatika**



**Dr. Cahyo Crysdiان**

**NIP. 19740424 200901 1 008**

## MOTO

❁❁❁ Sesungguhnya Allah akan memberikan ❁❁❁  
❁❁❁ petunjuk kepada orang yang bersungguh-sungguh ❁❁❁

وَالَّذِينَ جَاهَدُوا فِينَا لَنَهْدِيَنَّهُمْ سُبُلَنَا وَإِنَّ اللَّهَ لَمَعَ الْمُحْسِنِينَ ﴿٦٩﴾  
“Dan orang-orang yang berjihad (bersungguh-sungguh) dalam  
memperoleh hidayahKu, niscaya akan Kami tunjuki jalan-jalanKu.  
dan Sesungguhnya Allah benar-benar beserta orang-orang yang  
berbuat baik” (QS. Al-Ankabuut: 69)

## PERSEMBAHAN

*Dengan memanjatkan puji dan syukur kepada Allah SWT, saya persembahkan sebuah karya ini kepada :*

*Kedua orang tua saya tercinta, bapak Edi Imam Santoso, S.P dan ibu Sofiana Mujiati S, Pd. Kakak-kakaku tersayang mas Rama Arya Kusuma, mas Raga Widi Santoso dan Mba Mariati, ponakan-ponakanku Bilqis, Azzam dan Affan, serta seluruh keluarga besarku.*

*Dosen pembimbing bapak Zainal Abidin, M.Kom dan Dr. Cahyo Crys dian, dan seluruh dosen jurusan Teknik Informatika UIN Maliki Malang, serta guru-guruku, terimakasih atas segala ilmu yang telah diberikan.*

*Sahabat-sahabat seperjuanganku, Reza Perdana, Novi Ulan Sully Rahayu, teman-teman tim penelitian osteoporosis Meidoasa, Ida, Yoan, Delliana, Shendy, Risma, Mita dan Alif, serta teman-teman jurusan Teknik Informatika UIN Maliki Malang.*

*Orang-orang yang aku sayangi yang tak mampu aku sebutkan satu-persatu, terimakasih.*

*Semoga kita semua selalu mendapat rahmat dari Allah SWT dan termasuk golongan orang-orang yang diberi petunjuk jalan yang lurus hingga waktu dimana kita bertemu dengan Dzat Yang Maha Pengasih dan Penyayang. AMIN.*

**SURAT PERNYATAAN**  
**ORISINALITAS PENELITIAN**

Saya yang bertandatangan di bawah ini:

Nama : Devi Arum Sari  
NIM : 09650093  
Fakultas/Jurusan : Sains dan Teknologi / Teknik Informatika  
Judul Penelitian : Segmentasi *Foramen Mentale* pada Citra *Dental Penoramic Radiograph* menggunakan *Thresholding Fuzzy Type II*

Menyatakan dengan sebenarnya bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri, bukan merupakan pengambil alihan data, tulisan atau pikiran orang lain yang saya akui sebagai hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka. Apabila di kemudian hari terbukti atau dapat dibuktikan skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 24 Oktober 2013

Yang Membuat Pernyataan,



Devi Arum Sari

09650093

## KATA PENGANTAR



Segala puji bagi Allah SWT, karena atas rahmat, hidayah serta karuniaNya, penulis dapat menyelesaikan skripsi yang berjudul “Segmentasi *Foramen Mentale* pada *Citra Dental Panoramic Radiograph* menggunakan *Thresholding Fuzzy Type II*” sebagai salah satu syarat untuk memperoleh gelar sarjana pada program studi Teknik Informatika jenjang Strata-1 Universitas Islam Negeri (UIN) Maulana Malik Ibrahim Malang.

Shalawat serta salam semoga senantiasa Allah limpahkan kepada Nabi Muhammad SAW, keluarga dan para sahabat yang telah membimbing umat dari gelapnya alam jahiliyah menuju cahaya Islam yang diridhoi oleh Allah SWT.

Penulis menyadari adanya banyak keterbatasan yang penulis miliki, sehingga ada banyak pihak yang telah memberikan bantuan baik moril maupun materil dalam menyelesaikan skripsi ini. Maka dari itu dengan segenap kerendahan hati patutlah penulis mengucapkan terima kasih kepada:

1. Prof. DR. H. Mudjia Rahardjo, M.Si, selaku rektor UIN Maulana Malik Ibrahim Malang, yang telah banyak memberikan pengetahuan dan pengalaman yang berharga.
2. Dr. Hj. Bayyinatul Muchtaromah., drh., M.Si selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Dr. Cahyo Crysdiyan, selaku Ketua Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim

Malang dan selaku dosen pembimbing I yang telah senantiasa meluangkan waktu untuk membimbing, dan mengarahkan penulis dalam pengerjaan skripsi.

4. Zainal Abidin, M.Kom selaku dosen pembimbing II yang telah meluangkan waktu untuk membimbing, memotivasi, mengarahkan dan memberi masukan serta kepercayaan dalam pengerjaan skripsi ini.
5. Seluruh Dosen Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang telah memberikan ilmu, bimbingan, pengetahuan dan pengalaman sebagai pedoman dan bekal bagi penulis.
6. Segenap sivitas akademik Jurusan Teknik Informatika, terimakasih atas bantuan serta dukungan kepada penulis.
7. Bapak dan Ibuku tercinta, Bapak Edi Imam Santoso, S.P dan Ibu Sofiana Mujiati S,Pd, kakak-kakakku tersayang serta seluruh keluarga besar yang senantiasa memberikan doa dan restunya kepada penulis dalam menuntut ilmu dan menyelesaikan skripsi ini.
8. Teman-teman tim penelitian osteoporosis dan sahabat-sahabat seperjuangan jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang yang tidak mampu penulis sebutkan satu-persatu, atas segala yang telah diberikan penulis ucapkan terimakasih yang sebesar-besarnya.

Penulis menyadari dalam skripsi ini masih banyak kekurangan dan jauh dari sempurna, untuk itu penulis selalu menerima segala kritik dan saran dari pembaca. Harapan penulis, semoga karya ini bermanfaat bagi kita semua.

Malang, 24 Oktober 2013

Penulis



## DAFTAR ISI

HALAMAN JUDUL .....	i
HALAMAN PENGANTAR .....	ii
HALAMAN PERSETUJUAN .....	iii
HALAMAN PENGESAHAN .....	iv
MOTO .....	v
PERSEMBAHAN .....	vi
HALAMAN PERNYATAAN .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI .....	xi
DAFTAR TABEL .....	xiii
DAFTAR GAMBAR .....	xiv
DAFTAR LAMPIRAN .....	xvii
ABSTRAK .....	xviii
ABSTRACT .....	xix
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	6
1.3 Tujuan Penelitian .....	6
1.4 Manfaat Penelitian .....	6
1.5 Batasan Masalah .....	6
1.6 Sistematika Penulisan .....	7

BAB II TINJAUAN PUSTAKA.....	8
2.1 <i>Foramen mentale</i> dan Osteoporosis .....	8
2.2 <i>Thresholding fuzzy type II</i> .....	11
BAB III PERANCANGAN DAN IMPLEMENTASI SISTEM.....	17
3.1 Perancangan Aplikasi .....	17
3.1.1 Desain Proses.....	18
3.1.2 Desain Antarmuka Aplikasi.....	44
3.2 Implementasi Aplikasi.....	47
3.2.1 Implementasi <i>Preprocessing</i> .....	48
3.2.2 Implementasi <i>Main processing</i> .....	51
3.2.3 Implementasi <i>Post Processing</i> .....	56
BAB IV HASIL DAN PEMBAHASAN .....	62
4.1 Langkah-langkah Uji Coba.....	62
4.2 Hasil Uji Coba .....	63
4.3 Pembahasan .....	64
4.4 Integrasi Metode <i>Thresholding Fuzzy Type II</i> dengan <i>Al-qur'an</i> .....	69
BAB V PENUTUP.....	71
5.1 Kesimpulan.....	71
5.2 Saran .....	72
DAFTAR PUSTAKA .....	73
LAMPIRAN.....	75

## DAFTAR TABEL

Tabel 2.1 Prosentase keakuratan metode <i>thresholding</i> .....	11
Tabel 2.2 Prosentase keakuratan metode <i>thresholding fuzzy type II</i> dengan fungsi keanggotaan yang berbeda .....	13
Tabel 2.3 Prosentase keakuratan metode <i>thresholding fuzzy type II</i> dengan nilai $\alpha$ yang berbeda .....	13
Tabel 3.1 Perhitungan histogram citra keabuan.....	25
Tabel 3.2 Data citra objek <i>foramen mentale</i> berdasarkan keadaan nilai $i$ .....	26
Tabel 3.3 Perhitungan batas atas dan bawah pada pergeseran fungsi keanggotaan.....	34
Tabel 3.4 Total <i>ultrafuzziness</i> pada masing-masing pergeseran fungsi keanggotaan.....	35
Tabel 3.5 Studi kasus pengurutan data dengan <i>buble sort descending</i> .....	40
Tabel 4.1 Akurasi, presisi dan <i>recall</i> citra RoI <i>foramen mentale</i> bagian 1.....	65
Tabel 4.2 Akurasi, presisi dan <i>recall</i> citra RoI <i>foramen mentale</i> bagian 2.....	67

## DAFTAR GAMBAR

Gambar 1.1 <i>Dual-photon absorptiometry</i> (DPA) dari tulang mandibula, ada 3 area penting yaitu (1) <i>ramus</i> ; (2) <i>body</i> mandibula, dan (3) <i>symphysis</i> .....	3
Gambar 1.2 (a) Lebar ramus pada tulang mandibula (Ra); (b) Tinggi <i>body</i> mandibula (pa, ph, pf) pada tiga tempat karakteristik (A, H, dan F); (c) Tinggi korpus mandibula pada bagian atas dan dan bawah mandibular kanal dan <i>foramen mentale</i> .....	3
Gambar 1.3 Pengukuran lebar mandibula.....	4
Gambar 2.1 Mandibula dan <i>foramen mentale</i> .....	10
Gambar 2.2 Letak <i>foramen mentale</i> pada citra DPR .....	10
Gambar 2.3 (a) citra sample percobaan; (b) metode <i>thresholding fuzzy type I</i> ; (c) metode <i>thresholding otsu</i> ; (d) metode <i>thresholding fuzzy type II</i> .....	12
Gambar 2.4 (a) sampel citra percobaan (b) citra <i>thresholding fuzzy type II</i> .....	14
Gambar 2.5 Pergeseran fungsi keanggotaan pada skala keabuan .....	15
Gambar 2.6 Pergeseran dua fungsi keanggotaan (objek dan <i>background</i> ).....	15
Gambar 3.1 Blok diagram perancangan aplikasi .....	18
Gambar 3.2 <i>Flowchart</i> masukan citra RoI <i>foramen mentale</i> .....	19
Gambar 3.3 RoI <i>foramen mentale</i> sebelah kanan dan kiri pada citra DPR.....	20
Gambar 3.4 <i>Flowchart</i> konversi citra RGB menjadi citra keabuan.....	21
Gambar 3.5 Posisi bit warna RGB .....	21
Gambar 3.6 Grafik histogram keabuan .....	23
Gambar 3.7 <i>Flowchart</i> histogram keabuan.....	24

Gambar 3.8 Matrik citra <i>grayscale</i> .....	24
Gambar 3.9 Fungsi keanggotaan objek dengan $a_O = 52$ , $b_O = 57$ dan $c_O = 73$ .....	27
Gambar 3.10 Fungsi keanggotaan <i>background</i> $a_B = 50$ , $b_B = 76$ dan $c_B = 104$ .....	27
Gambar 3.11 Pergeseran MFO dan MFB pada perancangan aplikasi .....	28
Gambar 3.12 Batasan kenggotaan <i>fuzzy type II</i> ( <i>Footprint Of Uncertainty</i> atau FOU) (Tizhoosh, 2005) .....	29
Gambar 3.13 <i>Flowchart thresholding fuzzy type II</i> .....	32
Gambar 3.14 Pergeseran dua fungsi keanggotaan dan nilai $T$ .....	36
Gambar 3.15 Pengisian permukaan topografi dan pembentukan dam .....	37
Gambar 3.16 <i>Flowchart</i> tahapan transformasi <i>watershed</i> .....	38
Gambar 3.17 <i>Studi kasus transformasi watershed berdasarkan 4 tetangga yang terhubung</i> , (a): Matrik citra asli; (b) Tahap <i>flooding</i> $h=0$ ; (c) Tahap <i>flooding</i> $h=1$ ; (d) Tahap <i>flooding</i> $h=2$ ; (e) Tahap <i>flooding</i> $h=3$ .....	39
Gambar 3.18 <i>Flowchart</i> algoritma <i>buble sort descending</i> (pengurutan gelembung menurun) .....	40
Gambar 3.19 <i>Klasifikasi piksel pada citra output aplikasi terhadap citra referensi</i> .....	42
Gambar 3.20 <i>Flowchart</i> perhitungan akurasi, presisi dan <i>recall</i> .....	42
Gambar 3.21 (a) Citra biner referensi, (b) Citra biner hasil segmentasi, (c) Kondisi piksel hasil klasifikasi citra biner (a) dan (b) .....	43
Gambar 3.22 Perancangan antarmuka <i>frame</i> histogram .....	44
Gambar 3.23 Perancangan antarmuka <i>frame</i> utama aplikasi .....	45
Gambar 3.24 Antarmuka proses konversi citra keabuan .....	48

Gambar 3.25 <i>Sourcecode</i> konversi keabuan pada kelas <i>SegFM_GUI.java</i> .....	49
Gambar 3.26 <i>Sourcecode</i> konversi keabuan pada kelas <i>ProsesCitra.java</i> .....	49
Gambar 3.27 Antarmuka proses histogram <i>grayscale</i> .....	50
Gambar 3.28 <i>Sourcecode</i> fungsi histogram pada kelas <i>ProsesCitra.java</i> .....	50
Gambar 3.29 <i>Sourcecode</i> eksekusi fungsi histogram pada <i>SegFM_GUI.java</i> .....	50
Gambar 3.30 Antarmuka proses <i>thresholding fuzzy type II</i> .....	51
Gambar 3.31 <i>Sourcecode</i> eksekusi fungsi <i>thresholding</i> pada kelas <i>SegFM_GUI.java</i> .....	53
Gambar 3.32 <i>Sourcecode thresholding fuzzy type II</i> pada kelas <i>ProsesCitra.java</i> .....	56
Gambar 3.33 Antarmuka proses transformasi <i>watershed</i> tanpa <i>thresholding</i> dan transformasi <i>watershed</i> dengan <i>thresholding fuzzy type II</i> .....	57
Gambar 3.34 <i>Sourcecode</i> memanggil fungsi <i>matlab</i> pada kelas <i>ProsesCitra.java</i> .....	57
Gambar 3.35 Antarmuka <i>sorting region</i> hasil transformasi <i>watershed</i> .....	58
Gambar 3.36 <i>Sourcecode bubble sort descending</i> pada <i>ProsesCitra.java</i> .....	59
Gambar 3.37 Antarmuka proses pengukuran akurasi, presisi dan <i>recall</i> (bagian 1 dan bagian 2).....	60
Gambar 3.38 <i>Sourcecode</i> mengukur akurasi, presisi, dan <i>recall</i> pada kelas <i>ProsesCitra.java</i> .....	61
Gambar 4.1 Hasil uji coba aplikasi pada satu citra.....	64

## DAFTAR LAMPIRAN

Lampiran 1: Tabel hasil uji coba identifikasi <i>foramen mentale</i> .....	75
Lampiran 2: Data citra uji coba penentuan fungsi keanggotaan <i>fuzzy</i> .....	86
Lampiran 3: Data <i>TP, TN, FP, dan FN</i> citra uji coba .....	88



## ABSTRACT

Sari, Devi Arum. 2013. **Segmentation for Foramen Mentale from Dental Panoramic Radiograph Image using Type II Fuzzy Thresholding.**

Informatics Department of Faculty of Science and Technology. The State of Islamic University Maulana Malik Ibrahim Malang.

Promotor : (I) Dr. Cahyo Crysdiyan, M.Kom

(II) Zainal Abidin, M.Kom

Foramen mentale is blood vessel and nerve canal located in mandibula corpus. The measurement of perpendicular line of foramen mentale and mandibula cortical is measurement of two indexes which used for immediate detection of osteoporosis. The research segments foramen mentale from Dental Panoramic Radiograph (DPR) image by extracting Region of Interest (RoI) from foramen mentale excluding tooth and cortical. Proposed method is type II fuzzy thresholding by assuming image histogram as fuzzy set which generate threshold value (T) as output. The research uses watershed transformation as additional segmentation method and analyzes the best region as foramen mentale object. Sample experiment using 20 DPR image group of osteoporosis, so there are 40 images RoI foramen mentale tested. The experiment result indicate 13 images successfully detected the foramen mentale. Result indicates that the application performance to segment foramen mentale as object is 32.5%, and the performance of type II fuzzy thresholding method has accuration, precision and recall for 93.39%, 70.53% and 0,57%

**Keywords:** Foramen Mentale, Osteoporosis, Dental Penoramic Radiograph, Type II Fuzzy Thresholding, Watershed Transformation, Accuration, Precision, Recall

## ABSTRAK

Sari, Devi Arum. 2013. **Segmentasi *Foramen Mentale* pada Citra Dental Panoramic Radiograph menggunakan *Thresholding Fuzzy Type II***. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang, Pembimbing: (I) Dr. Cahyo Crysdiyan (II) Zainal Abidin, M.Kom

Kata kunci: *Foramen Mentale*, Osteoporosis, *Dental Panoramic Radiograph*, *Thresholding Fuzzy Type II*, Transformasi *Watershed*, *Region*, Akurasi, Presisi, *Recall*

*Foramen mentale* merupakan suatu saluran keluar pembuluh darah dan saraf yang terletak pada korpus mandibula. Pengukuran garis tegak lurus pada *foramen mentale* dan kortikal mandibula mempengaruhi pengukuran dua indeks yang digunakan untuk mendeteksi dini osteoporosis. Penelitian ini mensegmentasi *foramen mentale* pada citra *Dental Panoramic Radiograph* (DPR) dengan mengambil *Region of Interest* (RoI) dari *foramen mentale* tanpa tampak gigi dan kortikal. Metode yang digunakan yaitu *thresholding fuzzy type II* dengan mengasumsikan histogram citra sebagai himpunan *fuzzy* sehingga didapatkan suatu *output* berupa nilai ambang *threshold* ( $T$ ). Peneliti menggunakan tambahan metode segmentasi yaitu metode transformasi *watershed* serta menganalisis *region* yang paling berarti sebagai objek *foramen mentale*. Sampel percobaan menggunakan 20 citra DPR kelompok osteoporosis, sehingga ada 40 citra RoI *foramen mentale* yang diuji. Hasil percobaan menunjukkan 13 citra berhasil mendeteksi *foramen mentale*. Aplikasi mampu mendeteksi *foramen mentale* dengan prosentase keberhasilan sebesar 32.5%, sedangkan kinerja metode *thresholding fuzzy type II* untuk menunjang aplikasi ini memiliki akurasi, presisi dan *recall* sebesar 93.39%, 70.53% dan 0.57%.

## **BAB I**

### **PENDAHULUAN**

#### **1.1 Latar Belakang**

Tulang merupakan rangka pembentuk dan penopang tubuh manusia. Tubuh manusia tersusun dari 206 tulang yang saling bersendian membentuk suatu sistem rangka (Soewolo, 2005). Tulang membentuk rongga yang melindungi struktur-struktur yang halus seperti otak, jantung, paru-paru dan alat-alat dalam rongga perut dan panggul, serta tulang menyediakan permukaan yang kuat untuk tempat melekatnya otot dan daging.

Susunan tulang terdiri atas sel-sel, matrik organik, dan mineral. Mineral terdiri atas kolagen dan bahan dasar yang mengandung monopolisakarida yang didalamnya terdapat kristaloid (kristal protein) tempat menyimpan kalsium dan fosfat. Kristaloid bagian dalam mengandung 97% kalsium tubuh, baik dalam bentuk anorganik maupun dalam bentuk garam (Syaifuddin, 2009:46).

Kalsium dalam tubuh berfungsi membentuk tulang, menghantarkan impuls kontraksi otot, koagulasi (pembekuan darah), dan membantu beberapa enzim pankreas. Jika kadar kalsium dalam darah menurun, maka tubuh akan mengambil persediaan kalsium pada tulang untuk proses metabolisme dalam darah.

Seseorang yang terus beraktifitas tanpa memperhatikan asupan kalsium yang cukup bagi tubuh dapat menjadi salah satu faktor dalam kerentanan tulang

keropos karena kepadatan mineral tulang yang berkurang. Jika kondisi ini terjadi secara terus-menerus maka akan berujung pada penyakit osteoporosis.

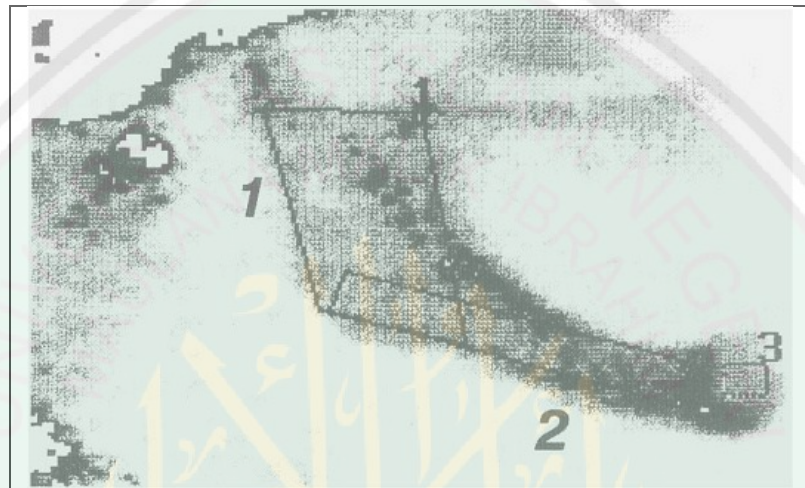
Osteoporosis merupakan suatu kondisi pengeroposan tulang secara merata dimana kekuatan tulang melemah karena pengikisan yang terjadi secara perlahan seiring aktivitas harian. Osteoporosis termasuk penyakit *Silent Epidemic Disease* yaitu penyakit yang menyerang secara diam-diam, tanpa danya tanda-tanda khusus sampai pasien mengalami patah tulang (Kemenkes, 2008).

Hasil analisa data risiko osteoporosis pada tahun 2005 yang dilakukan oleh Puslitbang Gizi Depkes RI dan sebuah perusahaan nutrisi dengan jumlah sampel 65.727 orang (22.799 laki-laki dan 42928 perempuan) pada 16 wilayah di Indonesia menunjukkan prosentase osteoporosis sebesar 10.3% dan osteopenia 41.7 %, dapat dikatakan dua dari lima penduduk Indonesia memiliki risiko untuk terkena osteoporosis (Kemenkes, 2008).

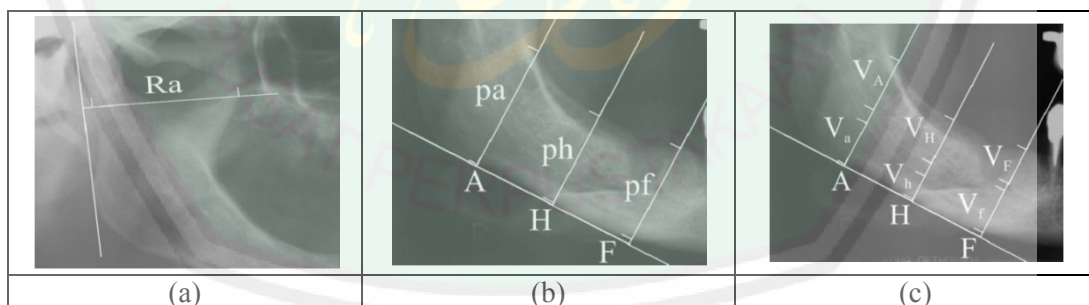
Osteoporosis dapat menyebabkan cacat seumur hidup dan juga mempengaruhi sistem pernapasan, karena pengeroposan tulang mengakibatkan ruas tulang belakang saling menekan akibat tidak mampu menyangga berat badan, masalah ini berdampak pada kesulitan bernapas karena rongga dada tidak dapat mengembang secara penuh. Hal ini menjadi salah satu faktor meningkatnya risiko kematian dini karena radang paru-paru dan penyakit paru-paru lainnya (Cosman, 2009).

Penelitian Horner (1996) mengenai osteoporosis berpengaruh terhadap kepadatan mineral tulang mandibula seperti ramus, *body* dan *symphysis*. Horner mengatakan osteoporosis dapat diprediksi melalui kepadatan mineral tulang

(BMD) melalui tulang mandibula secara lebih signifikan dari pada bagian tulang yang lain seperti ruas tulang belakang, tulang *femoral* leher, lengan atas dan lengan bawah dengan mengambil sample 40 orang wanita yang berumur 44 sampai 79 tahun.



Gambar 1.1 *Dual-photon absorptiometry* (DPA) dari tulang mandibula, ada 3 area penting yaitu (1) ramus; (2) *body* mandibula, dan (3) *symphysis* (sumber : Horner, 1996)



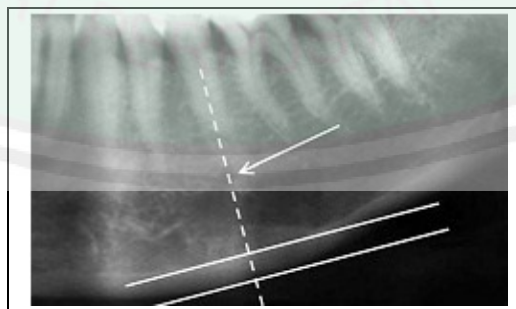
Gambar 1.2 (a) Lebar ramus pada tulang mandibula (Ra); (b) Tinggi body mandibula (pa, ph, pf) pada tiga tempat karakteristik (A, H, dan F); (c) Tinggi korpus mandibula pada bagian atas dan bawah mandibular kanal dan *foramen mentale* (sumber : Bozic dan Hren, 2005).

Bozic dan Hren (2005) memperkuat hasil penelitian dari Horner dengan menyatakan bahwa bagian yang sangat berpengaruh osteoporosis pada tulang mandibula berdasarkan kepadatan mineral tulang terletak pada bagian

ramus dan body mandibula yang dijadikan sebagai penentuan nilai enam indeks yang dimasukkan dalam perhitungan deteksi dini osteoporosis.

Penelitian Taguchi yang berjudul *use of dental panoramic radiographs in identifying younger postmenopausal women with osteoporosis* menyatakan dengan menggunakan citra panorama gigi (*Dental Panoramic Radiograph*) dapat terlihat ketebalan kortikel pada bagian body mandibula dan pengukuran dengan kelebaran tertentu dapat memprediksi osteoporosis, penelitian mengambil sample pada wanita *menopause* dengan usia 65 tahun. Pengukuran ketebalan kortikel ini mengambil garis lurus dengan posisi *foramen mentale* pada *body* mandibula (Taguchi, 2005).

Arifin (2005) melakukan penelitian pengukuran lebar kortikel mandibula menggunakan sistem komputasi. Sistem komputasi ini menerapkan algoritma *thresholding* menggunakan histogram berdasarkan kelas maksimal dan minimal dengan meng-klaster histogram membentuk dendogram hingga hanya menghasilkan dua kluster yaitu objek dan background.



Gambar 1.3 Pengukuran lebar mandibula (sumber : Taguchi, 2005).

Algoritma *thesholding* citra berkembang dengan beragamnya metode-metode *thresholding* yang dipublikasikan pada berbagai jurnal. Penelitian Arifin

(2007) pada proses pemisahan objek *foramen mentale* dan kortikel mandibula dengan *background* menggunakan metode *thresholding fuzzy type I*, sehingga model *fuzzy* yang digunakan terdiri dari indeks *fuzziness* (objek), dan indeks *non-fuzziness* (*background*).

Arifin (2010) mempublikasikan sebuah jurnal mengenai optimasi *thresholding* citra dengan menggunakan metode *thresholding fuzzy type II*, penelitian ini merupakan lanjutan penelitian sebelumnya dari Tizhoosh (2005) mengenai *thresholding* citra menggunakan metode *fuzzy type II*. Arifin melanjutkan penelitian ini dengan menerapkan optimasi ultrafuzziness pada algoritma *thresholding fuzzy type II*. Arifin menguji enambelas citra yang merupakan bagian dari citra DPR dengan membandingkan kinerja dari metode *thresholding fuzzy type II* dan metode *otsu* (Arifin, 2010).

Prosentase keakuratan masing-masing metode dalam mensegmentasi objek dan *background* pada enambelas citra yang diujikan menunjukkan *thresholding fuzzy type II* lebih akurat dibandingkan dengan metode *thresholding otsu*, dibuktikan dengan metode *otsu* rata-rata keakuratan 67.93%, sedangkan metode *fuzzy type II* memiliki keakuratan lebih tinggi mencapai 84.37%. Metode *Thresholding fuzzy type II* sangat cocok digunakan pada citra berskala keabuan yang memiliki kesamaran atau ketidakjelasan yang tinggi, dan kontras citra yang rendah (Arifin, 2010), sehingga peneliti akan menggunakan metode *thresholding fuzzy type II* yang merujuk pada jurnal Arifin (2010) untuk mengidentifikasi *foramen mentale* pada citra DPR.

Segmentasi bertujuan untuk membagi wilayah-wilayah pada citra. Pada aplikasi ini perlu adanya metode segmentasi untuk mengetahui *region foramen mentale* guna membantu pengukuran tingkat keberhasilan aplikasi dalam mengidentifikasi *foramen mentale* dan kemampuan metode *thresholding fuzzy type II* dalam menunjang aplikasi ini, sehingga peneliti menggunakan metode segmentasi transformasi *watershed*.

## 1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini sebagai berikut.

- a) Bagaimana kemampuan aplikasi dalam identifikasi *foramen mentale*?
- b) Bagaimana kinerja metode *thresholding fuzzy type II* dalam men-segmentasi *foramen mentale*?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini yaitu mengidentifikasi *foramen mentale* pada citra DPR menggunakan metode *thresholding fuzzy type II*.

## 1.4 Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk membantu mengidentifikasi *foramen mentale* secara otomatis pada citra DPR.

## 1.5 Batasan Masalah

Batasan masalah dari penelitian ini adalah:

- a) Citra DPR diperoleh dari proses *scanner*

- b) Data sampel yang digunakan merupakan *region of interest (RoI) foramen mentale* dengan memotong citra DPR sesuai kebutuhan
- c) Mengelompokkan piksel objek *foramen mentale* dan *background* menggunakan metode *thresholding fuzzy type II*

## 1.6 Sistematika Penulisan

### BAB I Pendahuluan

Bab I Pendahuluan berisi latar belakang, perumusan masalah, tujuan, batasan masalah, manfaat dan sistematika penulisan dalam penyusunan tugas akhir.

### BAB II Penelitian Terkait

Perkembangan penelitian terkait dari *foramen mentale* dan metode *thresholding fuzzy type II*.

### BAB III Perancangan dan Implementasi Sistem

Analisis kebutuhan serta desain perancangan dan implementasi aplikasi segmentasi *foramen mentale* dengan menggunakan *thresholding fuzzy type II* pada citra DPR.

### BAB IV Hasil dan Pembahasan

Hasil dan pembahasan dari pengujian aplikasi segmentasi *foramen mentale* dengan menggunakan *thresholding fuzzy type II* pada citra DPR.

### BAB V Penutup

Kesimpulan dan saran berdasarkan pembahasan hasil ujicoba.

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 *Foramen Mentale* dan Osteoporosis

Penelitian yang dilakukan oleh Klemetti dkk pada tahun 1993 mengenai keterkaitan *Panoramic Mandibular Index* (PMI) dan kepadatan mineral tulang atau *Bone Mineral Density* (BMD) yang dilakukan pada kelompok wanita *menopause*. Nilai kepadatan mineral tulang pada daerah leher *femoral* dan tulang belakang diukur dengan *Dual-energy X-ray Absorptiometry* (DXA) serta pengukuran secara kuantitatif pada area trabekula dan kortikal mandibula. Tinggi dan rendahnya rata-rata nilai PMI terikat dengan variabel nilai BMD (Klemetti dkk, 1993).

Pada tahun 1996 Horner melakukan sebuah penelitian yang lebih merujuk pada pendeteksian osteoporosis berdasarkan nilai BMD. Horner mengatakan osteoporosis dapat diprediksi melalui kepadatan mineral tulang (BMD) melalui tulang mandibula secara lebih signifikan dari pada bagian tulang yang lain seperti ruas tulang belakang, tulang *femoral* leher, lengan atas dan lengan bawah dengan mengambil sample 40 orang wanita yang berumur 44 sampai 79 tahun (Horner, 1996). BMD dari tulang mandibula dibagi menjadi 3 area yaitu BMD pada area ramus, *body* mandibula dan *symphysis* (Gambar 1.1).

Taguchi (1996) melakukan penelitian dengan bertujuan mengevaluasi kegunaan lebar inferior korteks mandibula pada DPR dalam mendiagnosis dini

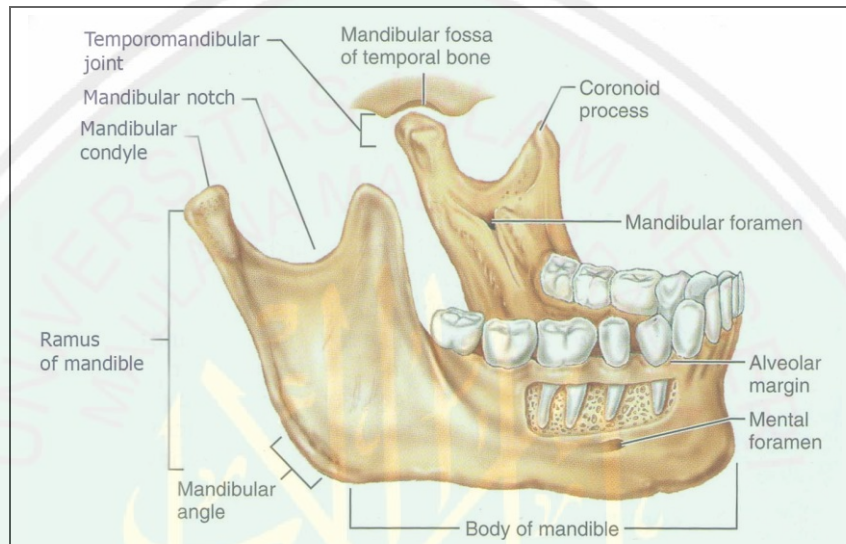
osteoporosis pada wanita *pascamenopause*. Penelitian ini menggunakan DXA dari tulang rahang atau mandibula dengan sampel 29 wanita sebelum *menopause* dan 95 wanita *pascamenopause*. Hasil penelitian menunjukkan bahwa DPR bisa digunakan dalam mendiagnosis osteoporosis (Taguchi, 1996).

Tinggi ramus dan *body* tulang mandibula memiliki hubungan yang cukup signifikan dengan resiko osteoporosis (Bozic dan Hren, 2005). Bozic dan Hren menggunakan enam indeks yang digolongkan menjadi dua. Pertama perhitungan tiga indeks berdasarkan perbandingan tinggi *body* mandibula dan lebar ramus, pengukurannya dimulai dengan membuat garis singgung pada bagian bawah luar tulang mandibula, selanjutnya membuat garis tegak lurus dengan garis tersebut pada tiga tempat yang berbeda, yaitu: (i) sudut dalam dari rahang (A) Pa, (ii) bersinggungan dengan *foramen mentale* (F) Pf, dan (iii) garis tegak lurus diantara titik A dan F (H) Ph (Gambar 1.1 b).

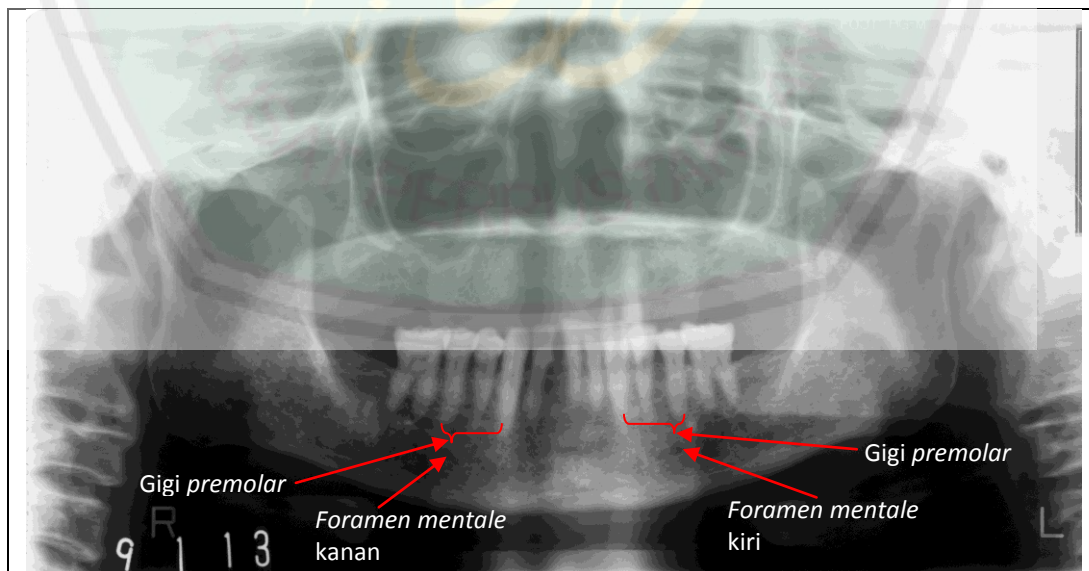
Kedua, perhitungan ketiga indeks berdasarkan perbandingan tinggi area *body* mandibula dengan batas area *mandibular canal* dan *foramen mentale* berdasarkan posisi masing-masing ketiga indeks. Pengukuran indeks diperoleh dari garis tegak lurus pada titik A, H, F dibagi menjadi dua garis berdasarkan batas *mandibular canal* dan *foramen mentale* (Gambar 1.1 b). Garis Pa dan Ph dibagi menjadi dua garis yaitu garis di atas *mandibular canal* VA dan VH dan garis di bawah *mandibular canal* Va dan Vh (Gambar 1.1 c). Garis Pf dibagi menjadi 2 garis dengan batas *foramen mentale* yaitu garis di atas *foramen mentale* VF dan garis di bawah *foramen mentale* Vf, dengan demikian keenam indeks yang dipaparkan oleh Bozic dan Hren dapat dirumuskan sebagai berikut.

$$I_1 = \frac{pa}{Ra}, I_2 = \frac{ph}{Ra}, I_3 = \frac{pf}{Ra} \quad (2.1)$$

$$I_4 = \frac{VA}{Va}, I_5 = \frac{VH}{Vh}, I_6 = \frac{VF}{Vf} \quad (2.2)$$



Gambar 2.1 Mandibula dan *foramen mentale* (sumber : Marieb, 2005)



Gambar 2.2 Letak *foramen mentale* pada citra DPR

Pada tulang mandibula terdapat dua *foramen mentale* yaitu pada bagian sisi kanan dan kiri mandibula, Gambar 2.1 menunjukkan letak *foramen mentale* pada tulang mandibula sisi kanan. *Foramen mentale* terletak di daerah bawah dan diantara gigi *premolar*, letak *foramen mentale* pada DPR lihat Gambar 2.2. *Foramen mentale* berfungsi sebagai saluran keluarnya pembuluh darah dan syaraf yaitu arteri dan *vena mental* serta *nervus mental*.

Pengukuran  $I_6$  dan  $I_3$  dapat dilakukan jika *foramen mentale* telah terdeteksi, sehingga perlu adanya aplikasi untuk mendeteksi *foramen mentale* secara otomatis pada citra DPR.

## 2.2 Thresholding Fuzzy Type II

Tizhoosh mempublikasikan sebuah jurnal pada tahun 2005 yang berjudul *Image Thresholding Using Type II Fuzzy Sets*. Pada penelitian ini menggunakan sampel percobaan sepuluh citra yang memiliki kerumitan seperti kontras yang rendah dan ketidakjelasan yang tinggi (Gambar 2.3).

Tabel 2.1 Prosentase keakuratan metode *thresholding*  
(sumber : Tizhoosh, Hamid R., 2005)

Citra	Type I	Otsu	Type II
Blocks	71.21	94.32	98.98
Zimba	86.31	97.87	99.52
Gearwheel	64.47	98.13	98.21
Shadow	75.75	90.64	94.39
Stones	39.96	95.96	96.99
Rice	99.98	94.34	99.65
Potatoes	98.96	98.01	99.77
Text	36.37	77.28	93.44
Ultrasonic	92.63	96.25	97.56
Newspaper	93.68	99.00	98.17
<b>Rata-rata</b>	<b>75.93</b>	<b>94.18</b>	<b>97.67</b>



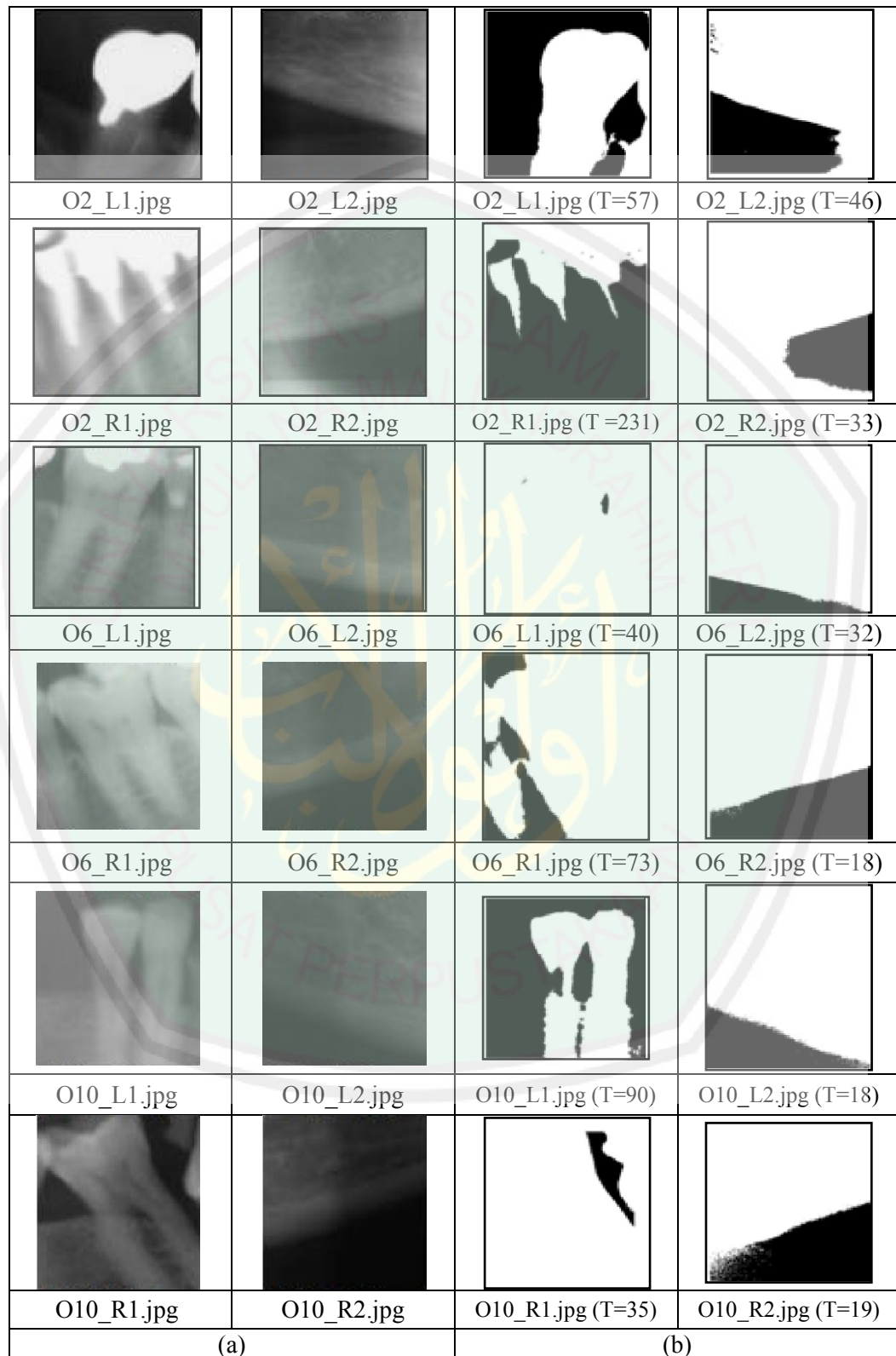
Hasil penelitian ini menunjukkan metode *Fuzzy Tipe II* memiliki prosentase ketepatan dalam mensegmentasi objek 97.67 %. Nilai prosentase ini lebih tinggi dibandingkan dengan prosentase metode yang lain seperti metode *otsu* dan *fuzzy type I* masing-masing memiliki prosentase 94.18% dan 75.93%.

Tabel 2.2 Prosentase keakuratan metode *thresholding fuzzy type II* dengan fungsi keanggotaan yang berbeda (sumber : Arifin, 2010)

Linguistic hedges parameter ( $\alpha$ )	Fungsi keanggotaan		
	Segitiga	Trapezioda	Z dan S
1	78.4	78.4	78.4
2	84.3	49.1	78.2
3	79.2	63.2	78.3
10	79.9	71.7	79.7
25	74.2	76.5	78.6
<b>Rata-rata</b>	<b>79.2</b>	<b>67.78</b>	<b>78.64</b>

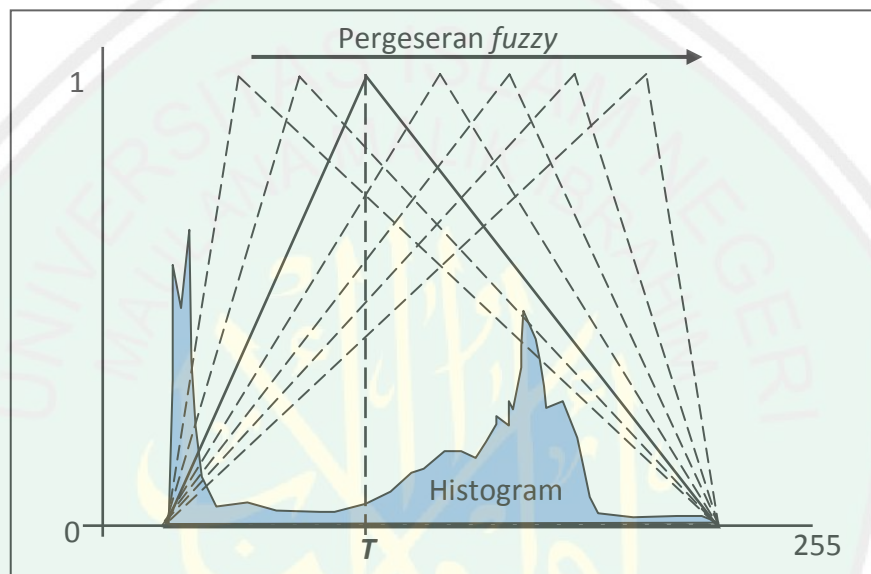
Tabel 2.3 Prosentase keakuratan metode *thresholding fuzzy type II* dengan nilai  $\alpha$  yang berbeda (sumber : Arifin, 2010)

Citra	Linguistics Hedges Parameter ( $\alpha$ )				
	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 10$	$\alpha = 25$
O1_L1.jpg	96.0281	95.9747	12.6846	63.2492	56.0278
O1_L2.jpg	66.9678	97.7448	97.7448	98.9319	99.9359
O1_R1.jpg	97.6456	14.5126	56.7947	54.5639	54.5639
O1_R2.jpg	58.2886	99.3362	99.0265	98.6893	99.3362
O2_L1.jpg	68.8385	79.2786	79.2786	55.4962	56.5033
O2_L2.jpg	70.7245	98.0103	97.5052	93.9255	92.9749
O2_R1.jpg	85.9177	38.7314	39.3951	40.4236	43.0756
O2_R2.jpg	76.3992	97.8470	98.3002	99.3744	96.7850
O6_L1.jpg	92.0044	92.1463	71.2799	64.8422	62.9120
O6_L2.jpg	87.4985	99.2432	99.2432	99.8749	58.2138
O6_R1.jpg	92.7032	87.3657	64.7766	64.2075	63.6154
O6_R2.jpg	70.8908	98.9288	99.2249	99.4919	99.7284
O10_L1.jpg	71.7590	69.2245	69.9661	81.8863	71.5820
O10_L2.jpg	77.9495	98.1262	98.1262	99.5163	66.4856
O10_R1.jpg	78.5431	85.7788	85.9528	65.0653	66.4993
O10_R2.jpg	62.5153	97.8241	98.5867	99.8993	99.2599
<b>Rata-rata</b>	<b>78.4171</b>	<b>84.3796</b>	<b>79.2429</b>	<b>79.9649</b>	<b>74.2187</b>

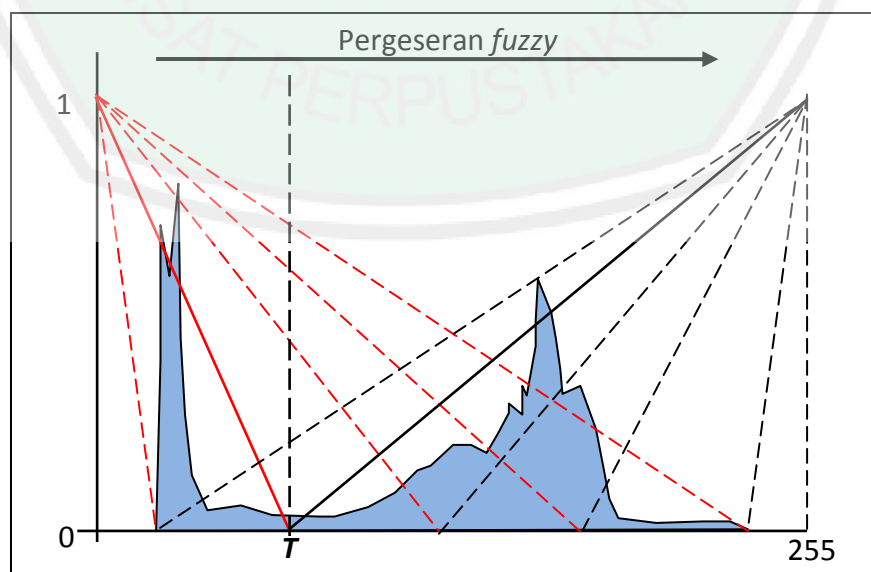


Gambar 2.4 (a) sampel citra percobaan (b) citra *thresholding fuzzy type II*  
(sumber : Arifin, 2010)

Arifin (2010) melanjutkan penelitian dari Tizhoosh (2005) dengan jurnal “*Image Thresholding using Ultrafuzziness based on Type II Fuzzy Sets*”. Arifin (2010) dan Tizhoos (2005) menganggap histogram citra sebagai himpunan *fuzzy type II* untuk mendapatkan *output* berupa nilai  $T$ .



Gambar 2.5 Pergeseran fungsi keanggotaan pada skala keabuan (sumber : Tizhoos, 2005)



Gambar 2.6 Pergeseran dua fungsi keanggotaan (objek dan *background*) (sumber : Arifin, 2010)

Letak perbedaan kedua penelitian ini pertama yaitu Arifin (2010) menambahkan natural number ( $e$ ) pada perhitungan *ultrafuzziness* untuk memberikan hasil yang optimal. Kedua, Tizhoos (2005) menggunakan satu fungsi keanggotaan (lihat Gambar 2.5) sedangkan Arifin (2010) menggunakan dua fungsi keanggotaan untuk objek dan *background* (lihat Gambar 2.6)

Hasil percobaan menunjukkan bahwa beberapa citra yang berbeda memiliki hasil *thresholding* yang optimal pada fungsi keanggotaan *fuzzy* yang berbeda pula. Pada citra DPR pemodelan fungsi *fuzzy* yang optimal yaitu pemodelan fungsi *fuzzy* berbentuk segitiga dengan nilai  $\alpha$  yaitu 2 (lihat Tabel 2.2 dan Tabel 2.3). Metode ini membutuhkan uji coba pada pemodelan fungsi keanggotaan yang digunakan untuk mendapatkan hasil *thresholding* yang terbaik.

## BAB III

### PERANCANGAN DAN IMPLEMENTASI SISTEM

#### 3.1 Perancangan Aplikasi

Penelitian ini bertujuan untuk membangun aplikasi segmentasi *foramen mentale* pada citra DPR menggunakan *thresholding fuzzy type II*. Input dari aplikasi berupa RoI *foramen mentale* (bagian kanan atau kiri) pada citra DPR, dan output berupa citra biner segmentasi *foramen mentale* setelah mengalami tahapan-tahapan pengolahan citra.

Perancangan aplikasi terdiri dari tiga tahapan utama yaitu tahapan *preprocessing*, *main processing* dan *post processing*. Berikut penjelasan singkat masing-masing tahapan perancangan aplikasi.

##### a) *Preprocessing*

Tahapan *preprocessing* yaitu memproses citra sesuai kebutuhan sebelum memasuki tahapan *main processing*, kebutuhan tersebut meliputi input citra RoI *foramen mentale*, konversi citra keabuan, dan histogram *grayscale*.

##### b) *Main processing*

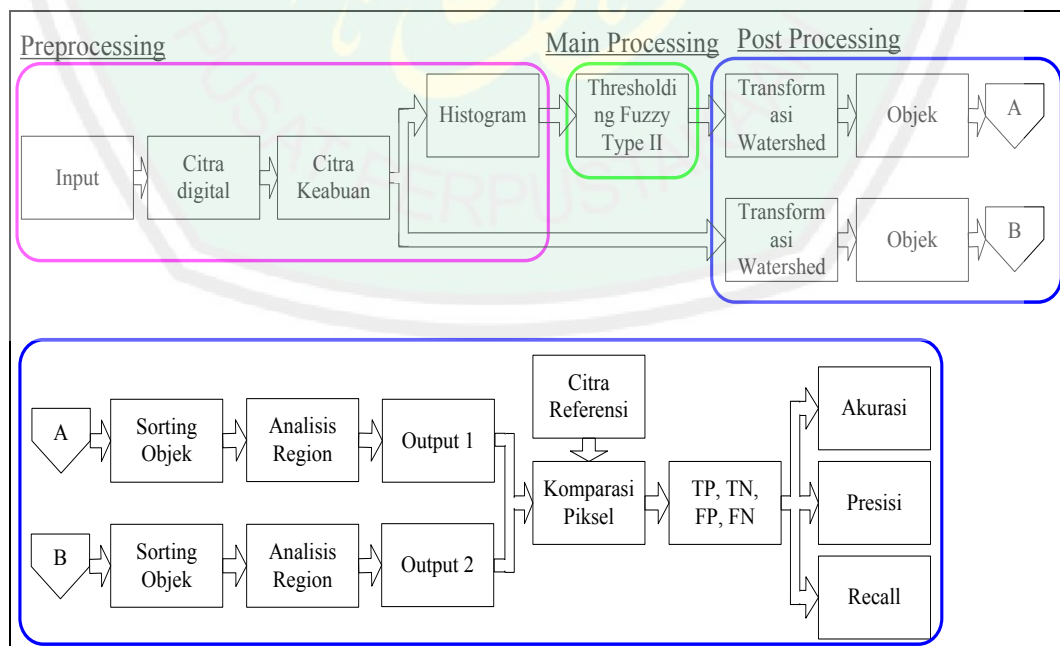
Tahapan *main processing* yaitu menerapkan metode *thresholding fuzzy type II*. Hal yang perlu diperhatikan sebelum memasuki perhitungan *fuzzy type II* yaitu menentukan fungsi *fuzzy* untuk *background* dan objek serta pergeseran dari kedua fungsi kenggotaan pada skala keabuan.

### c) *Post processing*

Tahapan *post precessing* membutuhkan penerapan metode segmentasi untuk mendapatkan bentuk *foramen mentale*. Metode segmentasi yang digunakan yaitu metode transformasi *watershed*, selanjutnya menganalisis *region* berarti dengan mengurutkan *region* hasil segmentasi transformasi *watershed*. Pengukuran akurasi, presisi dan *recall* berdasarkan citra referensi perlu dilakukan guna mengetahui tingkat keberhasilan metode *thresholding fuzzy type II*.

#### 3.1.1 Desain Proses

Desain proses aplikasi memiliki tiga tahapan utama yaitu tahapan pertama *preprocessing*, tahapan kedua *main processing*, dan tahapan ketiga *post precessing*. Desain proses perancangan aplikasi dalam bentuk blok diagram dapat dilihat pada Gambar 3.1.

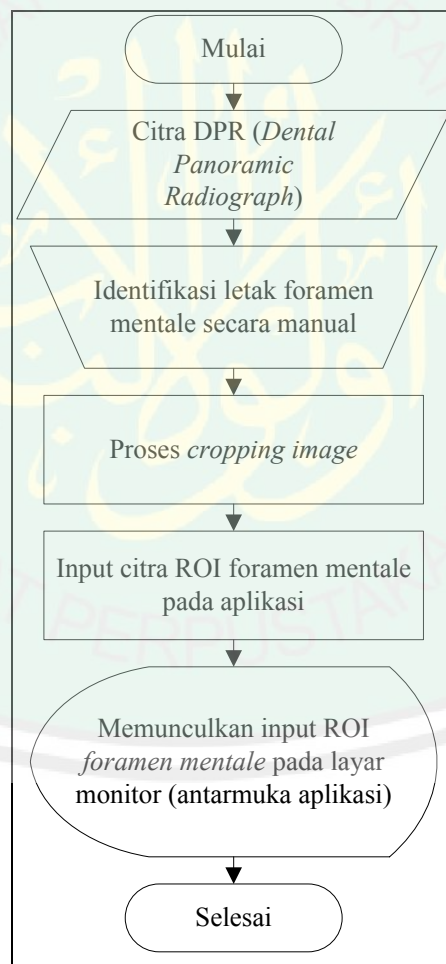


Gambar 3.1 Blok diagram perancangan aplikasi

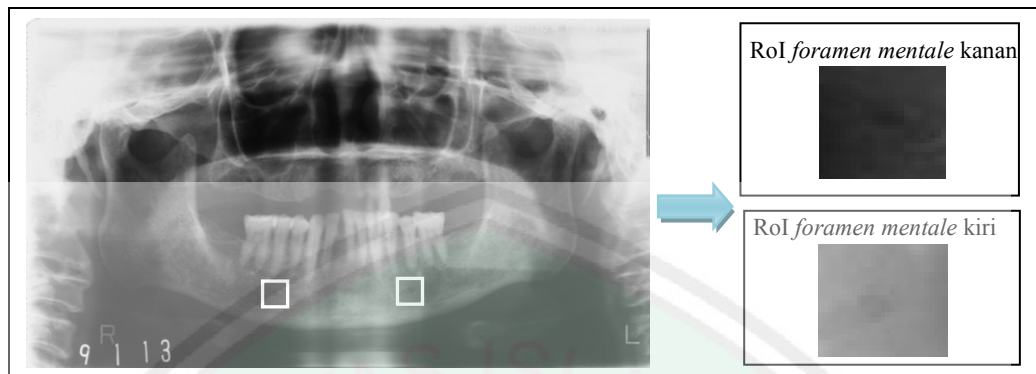
### 3.1.1.1 Preprocessing

#### a) Proses Masukan Citra ROI

Masukan dari aplikasi ini yaitu citra ROI *foramen mentale*. ROI *foramen mentale* merupakan area penting pada *foramen mentale* yaitu area letak *foramen mentale* tanpa tampak gigi dan kortikal mandibula (lihat Gambar 3.3). ROI *foramen mentale* diperoleh dengan melakukan identifikasi dan pemotongan area secara manual.



Gambar 3.2 Flowchart masukan citra ROI *foramen mentale*



Gambar 3.3 RoI *foramen mentale* sebelah kanan dan kiri pada citra DPR

Proses pemotongan citra dilakukan pada area yang diperkirakan sebagai RoI *foramen mentale* dengan aplikasi tambahan yang mendukung *cropping image*, setelah itu memasukkan citra RoI *foramen mentale* pada aplikasi dan menampilkan pada antarmuka aplikasi (lihat Gambar 3.2).

#### b) Konversi Citra Keabuan

Konversi menjadi citra keabuan bertujuan untuk menyederhanakan tiga layer (lapisan) warna pada citra RGB (*Red*, *Green* dan *Blue*) menjadi 1 lapisan intensitas pada keabuan (*grayscale*). Citra keabuan hanya memiliki satu nilai pada setiap pikselnya dengan 256 kombinasi warna keabuan dari hitam hingga putih (dari nilai 0 hingga 255). Tahapan desain proses konversi keabuan ditunjukkan pada Gambar 3.4. Langkah-langkah konversi citra keabuan sebagai berikut.

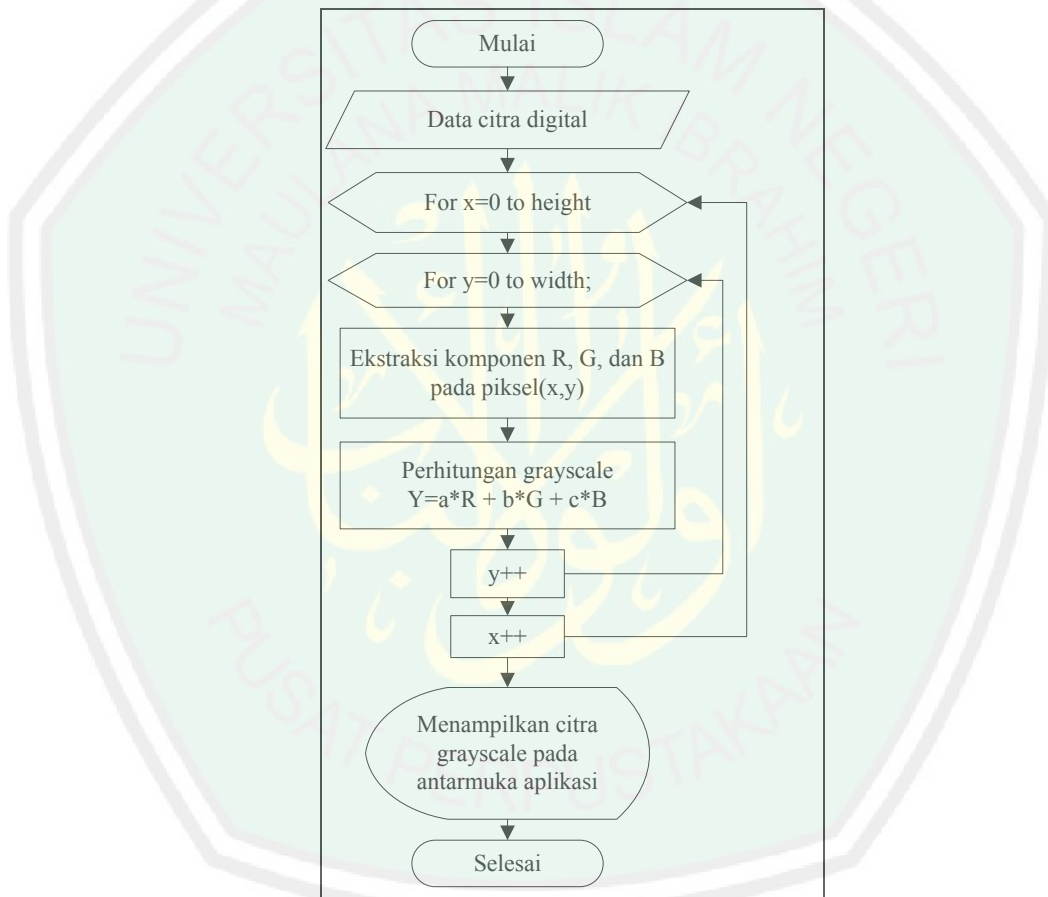
- (a) Menentukan nilai *red*, *green* dan *blue* masing-masing piksel pada suatu citra.
- (b) Menghitung nilai keabuan dengan mencari nilai rata-rata dari ketiga nilai *red*, *green* dan *blue*. Persamaan yang digunakan yaitu :

$$Y = a.R + b.G + c.B \quad (3.1)$$

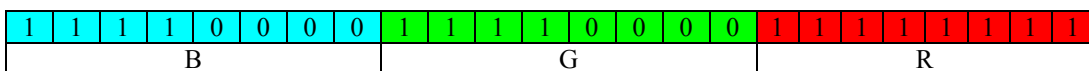
dimana  $R$ ,  $G$  dan  $B$  merupakan intensitas pada lapisan *Red*, *Green*, dan *Blue* sementara  $a$ ,  $b$ , dan  $c$  merupakan konstanta konversi yang memenuhi :

$$a + b + c = 1 \tag{3.2}$$

Nilai yang digunakan untuk konversi nilai adalah  $a=0.299$ ,  $b=0.587$ , dan  $c=0.114$ .



Gambar 3.4 *Flowchart* konversi citra RGB menjadi citra keabuan



Gambar 3.5 Posisi bit warna RGB

Perhitungan citra keabuan dengan studi kasus suatu piksel memiliki nilai RGB 24 bit yaitu 15790335 atau  $111100001111000011111111_2$ , Gambar 3.5

menunjukkan posisi bit dari masing-masing lapisan warna. Nilai  $R$ ,  $G$ , dan  $B$  didapatkan dengan melakukan pergeseran bit, dalam bahasa pemrograman java telah menyediakan fungsi pergeseran bit, sehingga didapatkan nilai sebagai berikut :

$$\text{Red} = 11111111 = 255$$

$$\text{Green} = 11110000 = 240$$

$$\text{Blue} = 11110000 = 240$$

Menghitung nilai keabuan dengan persamaan 3.1 :

$$Y = (0.299) \cdot 255 + (0.587) \cdot 240 + (0.114) \cdot 240$$

$$Y = 76.245 + 138.72 + 27.36 = 242.325$$

$$Y = 242.3$$

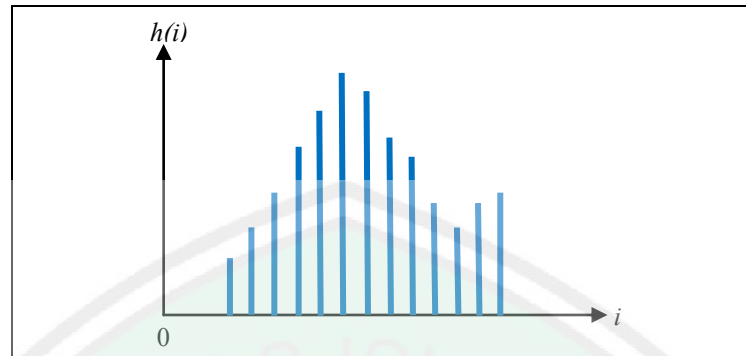
Perhitungan diatas menghasilkan nilai 242.325, jika dibulatkan maka akan menjadi 242, dengan mengaplikasikan prosedur perhitungan diatas pada semua piksel maka kita akan mendapatkan citra dengan format keabuan (*grayscale*).

### c) Histogram

Histogram mengelompokkan piksel berdasarkan derajat keabuannya mulai dari 0 hingga 255 dan menghitung nilai histogram dengan cara jumlah piksel pada derajat keabuan tertentu dibagi dengan jumlah keseluruhan piksel suatu citra, secara matematis dihitung dengan persamaan 3.3.

$$h_i = \frac{n_i}{n} \quad , i = 0, 1, \dots, L - 1 \quad (3.3)$$

dalam hal ini  $n_i$  sama dengan jumlah piksel yang memiliki derajat keabuan  $i$  dan  $n$  merupakan jumlah seluruh piksel di dalam citra.



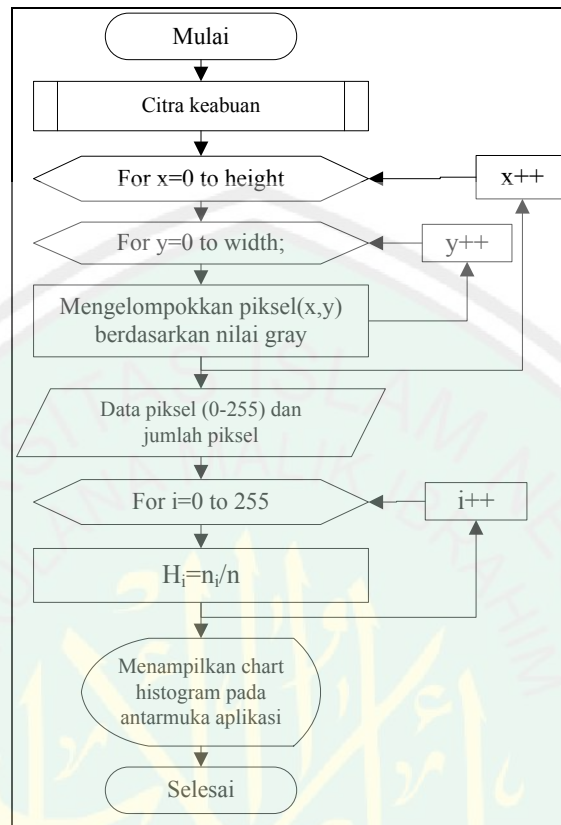
Gambar 3.6 Grafik histogram keabuan

Plot  $h_i$  dan  $i$  dinamakan *histogram* (Gambar 3.6), dengan grafis *histogram* ditampilkan menggunakan diagram batang. Perhatikan dari persamaan 3.3 bahwa  $n_i$  telah dinormalkan membaginya dengan  $n$ . Nilai  $h_i$  berada di diantara nilai 0 sampai 1.

$$\sum_{i=0}^{L-1} h_i = 1 \quad (3.4)$$

$$P(i \leq j) = \sum_{i=0}^j h_i, 0 \leq j \leq L - 1 \quad (3.5)$$

*Histogram* citra memberikan informasi yaitu nilai  $h_i$  menyatakan peluang (*probability*) pixel ( $P(i)$ ), dengan derajat keabuan  $i$ . Jumlah seluruh nilai  $h_i$  sama dengan 1 (lihat persamaan 3.4). Peluang suatu pixel memiliki derajat keabuan lebih kecil atau sama dengan derajat keabuan tertentu adalah jumlah  $h_i$  untuk  $0 \leq i \leq j$  (persamaan 3.5). Langkah-langkah perancangan aplikasi untuk perhitungan histogram ditunjukkan pada Gambar 3.7.



Gambar 3.7 Flowchart histogram keabuan

Studi kasus perhitungan histogram dengan menggunakan citra matrik ukuran  $9 \times 6$ , sehingga  $n$  citra adalah 54 (Gambar 3.8). Pixel dikelompokkan berdasarkan nilai  $i$  (lihat Tabel 3.1) dan perhitungan nilai  $h_i$  didapatkan dengan menggunakan persamaan 3.3, berikut contoh perhitungan nilai histogram pada  $i$  sama dengan 79,

$$h_{79} = \frac{n_{79}}{n} = \frac{1}{54} = 0.02$$

103	103	100	97	94	92	92	90	86
107	105	99	97	92	92	88	84	83
106	103	99	94	92	88	83	82	81
105	102	97	92	89	84	83	83	81
102	98	94	89	88	87	84	83	79
102	98	94	90	89	87	84	83	83

Gambar 3.8 Matrik citra *grayscale*

Tabel 3.1 Perhitungan histogram citra keabuan

$i$	$n_i$	$h_i$
79	1	0.02
81	2	0.04
82	1	0.02
83	7	0.13
84	4	0.07
86	1	0.02
87	2	0.04
88	3	0.06
89	3	0.06
90	2	0.04
92	6	0.11
94	4	0.07
97	3	0.06
98	2	0.04
99	2	0.04
100	1	0.02
102	3	0.06
103	3	0.06
105	2	0.04
106	1	0.02
107	1	0.02
<b>Jumlah</b>	<b>54</b>	<b>1.0</b>

Perhitungan nilai  $h_i$  dilakukan dari  $i$  samadengan 79 hingga  $i$  samadengan 107, sehingga jika nilai  $h_i$  dijumlahkan ( $i=79$  hingga  $i=107$ ) maka akan bernilai 1, sesuai dengan persamaan 3.4.

### 3.1.1.2 Main processing

Desain proses pada tahapan *main processing* yaitu menerapkan metode *thresholding fuzzy type II*. Citra masukan diasumsikan sebagai himpunan *fuzzy*. Himpunan *fuzzy* didapatkan berdasarkan nilai histogram citra. Ada dua tahapan penting dari desain proses *thresholding fuzzy type II* yaitu desain fungsi *fuzzy* dan algoritma *thresholding fuzzy type II*.

Fungsi keanggotaan *fuzzy* digunakan untuk mendapatkan himpunan keanggotaan batas atas dan bawah yang digunakan untuk menghitung nilai

*ultrafuzziness*. Nilai *ultrafuzziness* akan dibandingkan untuk mendapatkan nilai maksimal, jika nilai maksimal sudah didapatkan maka nilai ambang *thresholding*  $T$  dapat ditentukan.

a) Desain Fungsi Keanggotaan *Fuzzy Type II*

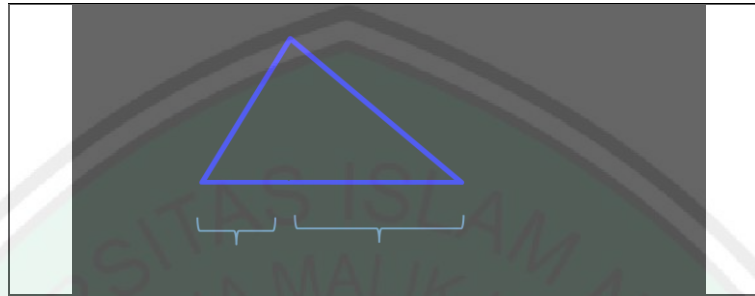
Tahap ini menentukan fungsi keanggotaan *fuzzy* yang tepat guna mendapatkan nilai ambang *thresholding* ( $T$ ) yang maksimal. Fungsi keanggotaan ditentukan berdasarkan data citra yang digunakan sebagai uji coba penentuan fungsi keanggotaan *fuzzy* (data citra pada Lampiran 2)

Tabel 3.2 Data citra objek *foramen mentale* berdasarkan posisi nilai  $i$

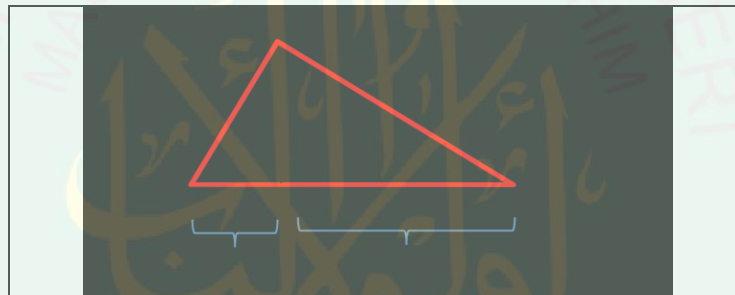
No	Nama Citra	Objek			Background		
		$i$ -terendah	$i$ - $h_i$ tertinggi	$i$ -tertinggi	$i$ -terendah	$i$ - $h_i$ tertinggi	$i$ -tertinggi
1	002-R	43	51	70	43	60	92
2	007-R	31	33	54	31	54	82
3	013-R	9	16	25	12	18	50
4	032-L	146	157	166	132	169	188
5	041-R	102	116	138	103	147	168
6	042-L	16	28	32	16	25	39
7	050-R	84	94	113	82	116	145
8	061-L	29	33	51	29	51	83
9	065-R	73	79	112	71	115	144
10	080-L	56	62	75	56	92	116
11	088-L	18	23	29	18	25	44
12	100-L	108	116	132	89	128	156
	MEDIAN	52	57	73	50	76	104

Peneliti mengidentifikasi *foramen mentale* pada beberapa citra DPR secara manual sebagai sampel percobaan untuk mengukur fungsi keanggotaan *fuzzy*. Langkah awal memisahkan *foramen mentale* dari *background* dengan menggunakan aplikasi yang mendukung proses *select object*. Data yang

didapatkan berupa nilai histogram dari citra *foramen mentale* tanpa *background* dan citra *background* tanpa *foramen mentale* (data citra pada Lampiran 2).



Gambar 3.9 Fungsi keanggotaan objek dengan  $a_O=52$ ,  $b_O=57$  dan  $c_O=73$

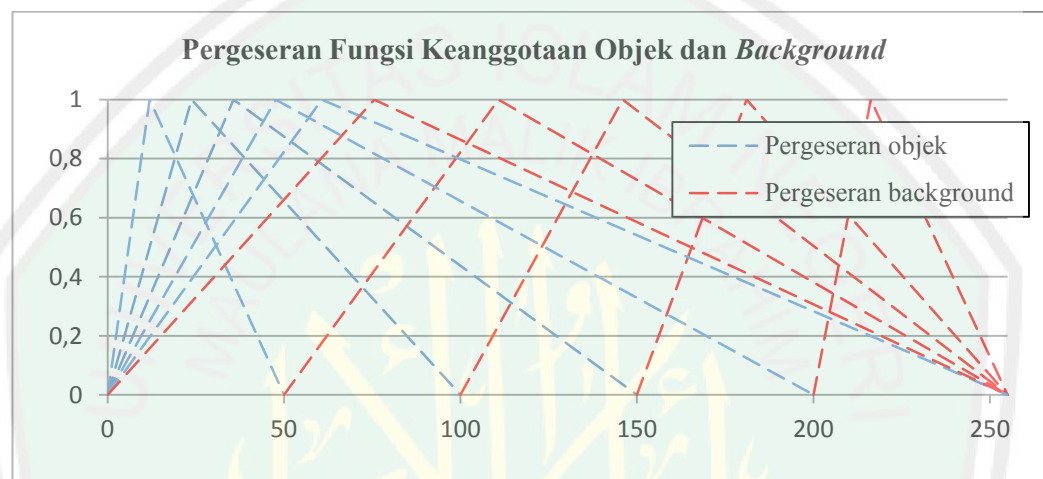


Gambar 3.10 Fungsi keanggotaan *background*  $a_B=50$ ,  $b_B=76$  dan  $c_B=104$

Hasil perhitung data objek, nilai median dari data *graylevel* terendah objek yaitu 52, median dari data *graylevel* tertinggi objek 73, dan median dari data *graylevel* objek yang memiliki  $h_i$  tertinggi yaitu 57. Data histogram objek *foramen mentale* menunjukkan hasil grafik yang menurun dari titik *graylevel* yang memiliki histogram tertinggi menuju titik *graylevel* 0 atau 255, sehingga fungsi keanggotaan *fuzzy* yang digunakan yaitu fungsi keanggotaan segitiga dengan nilai  $a_O$  yaitu 52,  $b_O$  yaitu 57 dan  $c_O$  yaitu 73 (Gambar 3.9).

Data *background* menunjukkan hasil perhitungan median dari data *graylevel* terendah yaitu 50, median dari data *graylevel* tertinggi yaitu 104 dan median dari data *graylevel* yang memiliki  $h_i$  tertinggi yaitu 76. Histogram

*background* juga menunjukkan hasil grafik yang menurun dari titik *graylevel* yang memiliki histogram tertinggi menuju titik 0 atau 255, sehingga fungsi keanggotaan *fuzzy* untuk *background* menggunakan fungsi keanggotaan segitiga dengan ukuran  $a_B$  yaitu 50,  $b_B$  yaitu 76 dan  $c_B$  yaitu 104 (Gambar 3.10).



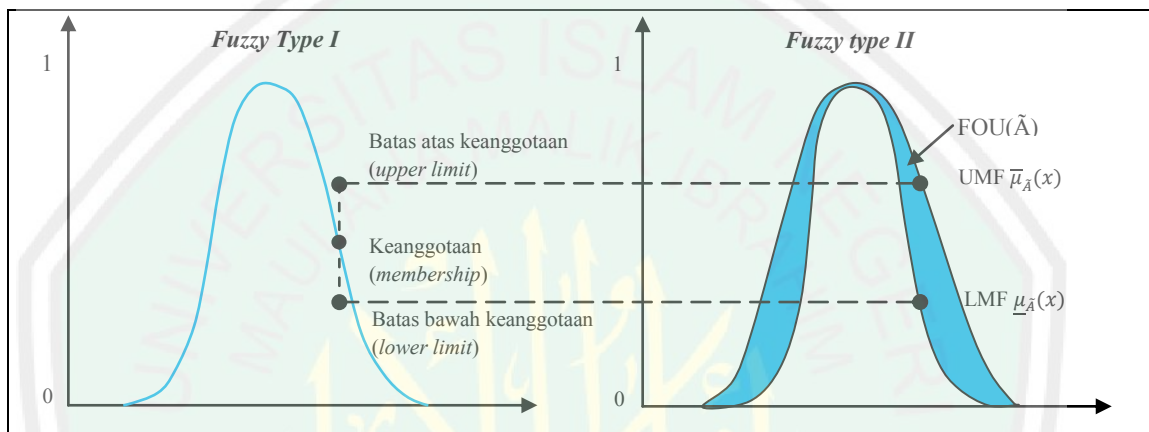
Gambar 3.11 Pergeseran MFO dan MFB pada perancangan aplikasi

Fungsi keanggotaan objek (MFO) (Gambar 3.9) dan *background* (MFB) (Gambar 3.10) jika digabungkan pada interval *graylevel* maka akan ditunjukkan sesuai pada Gambar 3.11, kedua fungsi keanggotaan mengalami pergeseran pada interval skala keabuan (Arifin, 2010). Interval nilai *grayscale* yaitu 0 sampai 255 sehingga pergeseran fungsi keanggotaan mengisi range antara 0 hingga 255.

#### b) *Thresholding Fuzzy Type II*

*Fuzzy type II* dapat dibayangkan sebagai fungsi keanggotaan *fuzzy type I* yang memiliki bayangan pada fungsi keanggotaannya (lihat daerah berarsir pada Gambar 3.12). Ketidakpastian pada fungsi keanggotaan *fuzzy type II* (daerah

berarsir pada Gambar 3.12) disebut *Footprint Of Uncertainty* (FOU), dengan demikian FOU juga dapat dikatakan batasan dua buah fungsi keanggotaan *fuzzy type II*, yaitu *Upper Membership Function* disingkat dengan UMF ( $\mu_U$ ) dan *Lower Membership Function* disingkat dengan LMF ( $\mu_L$ ).

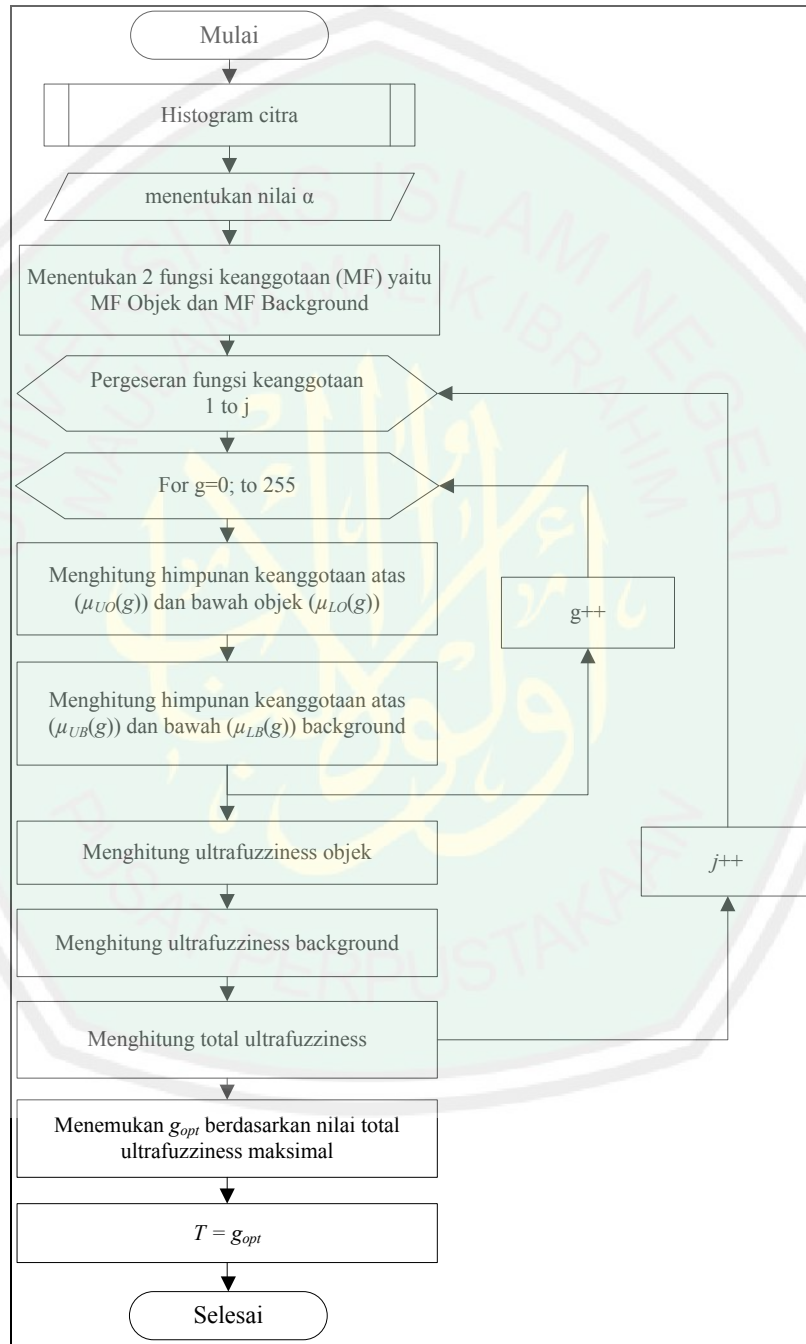


Gambar 3.12 Batasan keanggotaan *Fuzzy type II* (*Footprint Of Uncertainty* atau FOU) (Tizhoosh, 2005)

Penerapan metode *fuzzy* untuk mendapatkan nilai *output* berupa nilai ambang *thresholding* ( $T$ ) dengan menganggap histogram keabuan citra sebagai himpunan *fuzzy* (*flowchart thresholding fuzzy type II* lihat Gambar 3.13). Berikut penjelasan algoritma *thresholding fuzzy type II*.

- (1). Menentukan nilai  $\alpha$  untuk menghitung UMF dan UML dimana  $\alpha \in (1,2)$ , karena jika  $\alpha$  bernilai lebih dari 2 ( $\alpha \gg 2$ ) maka tidak berarti apa-apa pada data citra karena nilai *output* yang dihasilkan cenderung menghasilkan nilai ambang *thresholding* yang tidak baik. (Tizhoosh, 2005 dan Arifin, 2010).
- (2). Menentukan pergeseran kedua fungsi keanggotaan yaitu MFO dan MFB,  $j$  merupakan parameter yang menyatakan urutan pergeseran fungsi

keanggotaan, sehingga nilai  $j$  memenuhi kondisi  $j$  samadengan 1 hingga  $j$  samadengan  $L$  ( $j \in 1, 2, \dots L$ ).



Gambar 3.13 Flowchart thresholding fuzzy type II

(3). Menghitung nilai UMF dan LMF pada fungsi keanggotaan *background* dan objek pada masing-masing posisi pergeseran fungsi keanggotaan. Berikut perhitungan UMF dan LMF fungsi keanggotaan *background*

$$\mu_{UB}(g) = [\mu_B(g)]^{1/\alpha} \quad (3.6)$$

$$\mu_{LB}(g) = [\mu_B(g)]^\alpha \quad (3.7)$$

dalam hal ini  $\mu_{UB}(g)$  dan  $\mu_{LB}(g)$  menyatakan nilai batas atas dan batas bawah keanggotaan *background* (UMF dan LMF *background*) pada piksel ke  $g$  ( $g \in [0, L - 1]$ ), dan  $\mu_B(g)$  menyatakan nilai keanggotaan *background* pada piksel ke- $g$ . Sedangkan perhitungan UMF dan LMF fungsi keanggotaan objek sebagai berikut

$$\mu_{UO}(g) = [\mu_O(g)]^{1/\alpha} \quad (3.8)$$

$$\mu_{LO}(g) = [\mu_O(g)]^\alpha \quad (3.9)$$

dalam hal ini  $\mu_{UO}(g)$  dan  $\mu_{LO}(g)$  menyatakan nilai batas atas dan batas bawah keanggotaan objek (UMF dan LMF objek) pada piksel ke  $g$ , dan  $\mu_O(g)$  menyatakan nilai keanggotaan objek pada piksel ke- $g$ .

(4). Menghitung nilai *ultrafuzziness* objek dan *background*. *Ultrafuzziness background* dihitung dengan menggunakan persamaan 3.10, sedangkan *ultrafuzziness* objek menggunakan persamaan 3.11.

$$\tilde{\gamma}_B(\tilde{A}) = \frac{1}{MN} \sum_{g=0}^{L-1} h(g) \times e^{(\mu_{UB}(g) - \mu_{LB}(g))} \quad (3.10)$$

$$\tilde{\gamma}_O(\tilde{A}) = \frac{1}{MN} \sum_{g=0}^{L-1} h(g) \times e^{(\mu_{UO}(g) - \mu_{LO}(g))} \quad (3.11)$$

Pengukuran *ultrafuzziness background* ( $\tilde{\gamma}_B(\tilde{A})$ ) dan objek ( $\tilde{\gamma}_O(\tilde{A})$ ) pada suatu himpunan keanggotaan  $A$  (citra  $A$ ) yang memiliki lebar dan panjang citra yaitu  $M$  dan  $N$ , dengan  $L$  menyatakan jumlah derajat keabuan, sehingga  $g$  memiliki

nilai dari 0 hingga  $L-1$  ( $g \in [0, L-1]$ ), *histogram* piksel ke- $g$  dinyatakan dengan  $h(g)$  dan  $e$  merupakan *natural number* untuk mendapatkan perhitungan lebih akurat (Arifin, 2010).

- (5). Menghitung total *ultrafuzziness* ( $\tilde{\gamma}_{tot}$ ) dengan mengalikan *ultrafuzziness* objek dan *background* pada masing-masing posisi pergeseran fungsi keanggotaan ( $\tilde{\gamma}_{tot\ ke-1}, \tilde{\gamma}_{tot\ ke-2} \dots \tilde{\gamma}_{tot\ ke-j}$ ).

$$\tilde{\gamma}_{Tot}(\tilde{A}) = \tilde{\gamma}_B(\tilde{A}) \times \tilde{\gamma}_O(\tilde{A}) \quad (3.12)$$

- (6). Menentukan nilai  $g_{opt}$  berdasarkan nilai total *ultrafuzziness* yang tertinggi  
 (7). *Thresholding* citra dengan nilai  $T=g_{opt}$ . Jika nilai piksel citra lebih dari  $T$  maka nilai piksel menjadi 0 (hitam), sedangkan jika nilai piksel kurang atau samadengan  $T$  maka nilai piksel sama dengan 1 (putih).

Studi kasus perhitungan *thresholding fuzzy type II* terdapat nilai histogram citra dengan nilai  $i$  terendah 79 dan  $i$  tertinggi 107 (sesuai pada Tabel 3.1), sehingga banyak pergeseran :

$$j = (107 - 79) + 1 = 28$$

Selanjutnya menghitung batas atas dan bawah fungsi keanggotaan. Contoh perhitungan batas atas dan batas bawah pada fungsi objek, jika  $g$  berada pada titik 79 memiliki derajat keanggotaan 0.02 ( $\mu_o(g) = 0.02$ ) dengan nilai  $\alpha$  yang dimisalkan dengan 2, sehingga perhitungan UMF objek ( $\mu_{uo}(g)$ ) menggunakan persamaan 3.8, dengan langkah perhitungan :

$$\mu_{uo}(1) = [0.02]^{1/2} = \sqrt{0.02} = 0.1414$$

Perhitungan LMF objek ( $\mu_{Lo}(g)$ ) menggunakan persamaan 3.9, dengan langkah perhitungan :

$$\mu_{Lo}(1) = [0.02]^2 = 0.0004$$

perhitungan batas atas dan bawah fungsi keanggotaan dilakukan sepanjang nilai keabuan (Tabel 3.2).

Jika perhitungan himpunan keanggotaan telah selesai, dilanjutkan dengan perhitungan *ultrafuzziness* objek ( $\tilde{\gamma}_o(\tilde{A})$ ) dan *background* ( $\tilde{\gamma}_B(\tilde{A})$ ) sebagai berikut (menyesuaikan dengan contoh perhitungan langkah-langkah sebelumnya).

Langkah perhitungan *ultrafuzziness* objek ( $\tilde{\gamma}_o(\tilde{A})$ ) pada pergeseran fungsi keanggotaan ke-1 menggunakan persamaan 3.11 :

$$\begin{aligned} \tilde{\gamma}_o(\tilde{A}) &= \frac{1}{117 \times 129} \left( (h(0) \times e^{(\mu_{Uo}(0) - \mu_{Lo}(0))}) + (h(1) \times e^{(\mu_{Uo}(1) - \mu_{Lo}(1))}) \right. \\ &\quad \left. + \dots + (h(255) \times e^{(\mu_{Uo}(255) - \mu_{Lo}(255))}) \right) \\ &= \frac{1}{15093} \left( (0.0055 \times e^{(0.05 - 6.25 \times 10^{-6})}) \right. \\ &\quad \left. + (0.0129 \times e^{(0.04 - 2.56 \times 10^{-6})}) + \dots \right. \\ &\quad \left. + (0.0025 \times e^{(0.08 - 4.096 \times 10^{-5})}) \right) \\ &= \frac{1}{15093} \left( (5.782 \times 10^{-3}) + (13.4 \times 10^{-3}) + \dots + (2.708 \times 10^{-3}) \right) \\ &= \frac{1}{15093} \times 0.1217 \\ &= 8.063340621 \times 10^{-6} \end{aligned}$$

Tabel 3. 3 Perhitungan batas atas dan bawah pada pergeseran fungsi keanggotaan

Pergeseran fungsi keanggotaan		Gray-level (g)	Fungsi keanggotaan objek			Fungsi keanggotaan background		
Pergeseran $j$	Titik $g_i$		$\mu_{AO}$ (g)	$\mu_{uO}$ (g)	$\mu_{LO}$ (g)	$\mu_{AB}$ (g)	$\mu_{uB}$ (g)	$\mu_{LB}$ (g)
Ke-1	79	0	0	0	0	0	0	0
		1	0	0	0	0.0024	0.154	$5.76 \times 10^{-6}$
		⋮	⋮	⋮	⋮	⋮	⋮	⋮
		⋮	⋮	⋮	⋮	⋮	⋮	⋮
		254	0	0	0	0.0024	0.154	$5.76 \times 10^{-6}$
		255	0	0	0	0.0868	0.295	$7.53 \times 10^{-3}$
Ke-2	80	0	0.0036	0.06	$1.296 \times 10^{-5}$	0	0	0
		1	0	0	0	0	0	0
		⋮	⋮	⋮	⋮	⋮	⋮	⋮
		⋮	⋮	⋮	⋮	⋮	⋮	⋮
		254	0	0	0	0.0128	0.113	$1.63 \times 10^{-4}$
		255	0	0	0	0.0428	0.206	$1.83 \times 10^{-3}$
Ke-256	107	0	0.0121	0.11	$1.46 \times 10^{-4}$	0	0	0
		1	0.0049	0.07	$2.41 \times 10^{-5}$	0	0	0
		⋮	⋮	⋮	⋮	⋮	⋮	⋮
		⋮	⋮	⋮	⋮	⋮	⋮	⋮
		254	0.0003			0	0	0
		255	0.0004	0.02	$1.6 \times 10^{-7}$	0	0	0

Keterangan : nilai  $\alpha$  yang ditentukan dimisalkan dengan 2

Langkah perhitungan *ultrafuzziness background* ( $\tilde{\gamma}_B(\tilde{A})$ ) pada pergeseran fungsi keanggotaan ke-1, menggunakan persamaan 3.10 :

$$\begin{aligned}
 \tilde{\gamma}_B(\tilde{A}) &= \frac{1}{117 \times 129} \left( (h(0) \times e^{(\mu_{UB}(0) - \mu_{LB}(0))}) + (h(1) \times e^{(\mu_{UB}(1) - \mu_{LB}(1))}) \right. \\
 &\quad \left. + \dots + (h(255) \times e^{(\mu_{UB}(255) - \mu_{LB}(255))}) \right) \\
 &= \frac{1}{15093} \left( (0.0055 \times e^{(0.059 - 1.225 \times 10^{-5})}) \right. \\
 &\quad \left. + (0.0129 \times e^{(0.154 - 5.76 \times 10^{-6})}) + \dots \right. \\
 &\quad \left. + (0.0025 \times e^{(0.295 - 7.53 \times 10^{-3})}) \right)
 \end{aligned}$$

$$\begin{aligned}
&= \frac{1}{15093} ((5.83 \times 10^{-3}) + (15.04 \times 10^{-3}) + \dots + (3.33 \times 10^{-3})) \\
&= \frac{1}{15093} \times 1.2258 \\
&= 8.121645796 \times 10^{-5}
\end{aligned}$$

Perhitungan *ultrafuzziness* objek dan *background* dilakukan pada setiap posisi pergeseran fungsi keanggotaan dari pergeseran ke-1 hingga ke- $j$ .

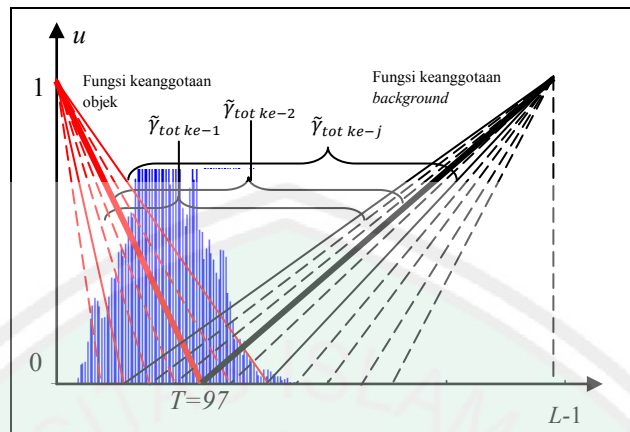
Perhitungan *ultrafuzziness* total menggunakan persamaan 3.12, contoh langkah perhitungan total *ultrafuzziness* pada pergeseran fungsi keanggotaan ke-1 sebagai berikut

$$\begin{aligned}
\tilde{\gamma}_{Total}(\tilde{A}) &= 8.063340621 \times 10^{-6} \times 8.121645796 \times 10^{-5} \\
&= 6.548759646 \times 10^{-10}
\end{aligned}$$

Tabel 3.4 Total *ultrafuzziness* pada masing-masing pergeseran fungsi keanggotaan

Pergeseran fungsi keanggotaan		<i>Ultrafuzziness</i>		Total <i>Ultrafuzziness</i> ( $\tilde{\gamma}_{tot}$ )
Pergeseran	Titik $g$	Objek ( $\tilde{\gamma}_o(\tilde{A})$ )	<i>Background</i> ( $\tilde{\gamma}_B(\tilde{A})$ )	
<b>Ke-1</b>	79	<b><math>8.063340621 \times 10^{-6}</math></b>	$8.121645796 \times 10^{-5}$	$6.548759646 \times 10^{-10}$
Ke-2	80	$1.727249527 \times 10^{-5}$	$1.857427471 \times 10^{-5}$	$3.208240721 \times 10^{-10}$
.	.	.	.	.
<b>Ke-13</b>	<b>97</b>	<b><math>9.68825718 \times 10^{-5}</math></b>	<b><math>1.190520379 \times 10^{-5}</math></b>	<b><math>11.53406761 \times 10^{-10}</math></b>
.	.	.	.	.
Ke-256	107	$1.331480052 \times 10^{-5}$	$8.57042747 \times 10^{-6}$	$11.41134363 \times 10^{-10}$

Keterangan : tulisan yang bercetak tebal merupakan fungsi keanggotaan yang memiliki total *ultrafuzziness* tertinggi ( $\tilde{\gamma}_{tot-maks}$ )



Gambar 3.14 Pergeseran dua fungsi keanggotaan dan nilai  $T$  (sumber : Arifin, 2010)

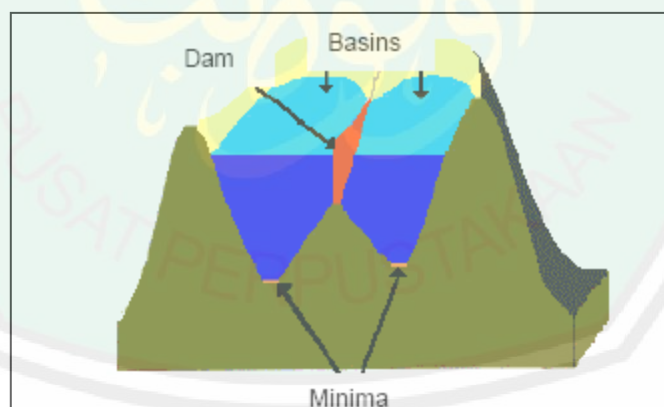
Langkah selanjutnya menentukan nilai  $g_{opt}$  berdasarkan nilai total *ultrafuzziness* tertinggi. Pada Tabel 3.4 menunjukkan hasil perhitungan *ultrafuzziness background*, objek dan total pada masing-masing pergeseran fungsi keanggotaan, sehingga jika diamati nilai tertinggi *ultrafuzziness* total yaitu  $11.53406761 \times 10^{-10}$  berada pada pergeseran ke-13 dengan titik  $g$  berada pada titik 97 ( $g_{opt}$ ), sehingga tahapan akhir didapatkan nilai  $T$  sama dengan 97 (Gambar 3.14).

### 3.1.1.3 Post processing

#### a) Transformasi *Watershed*

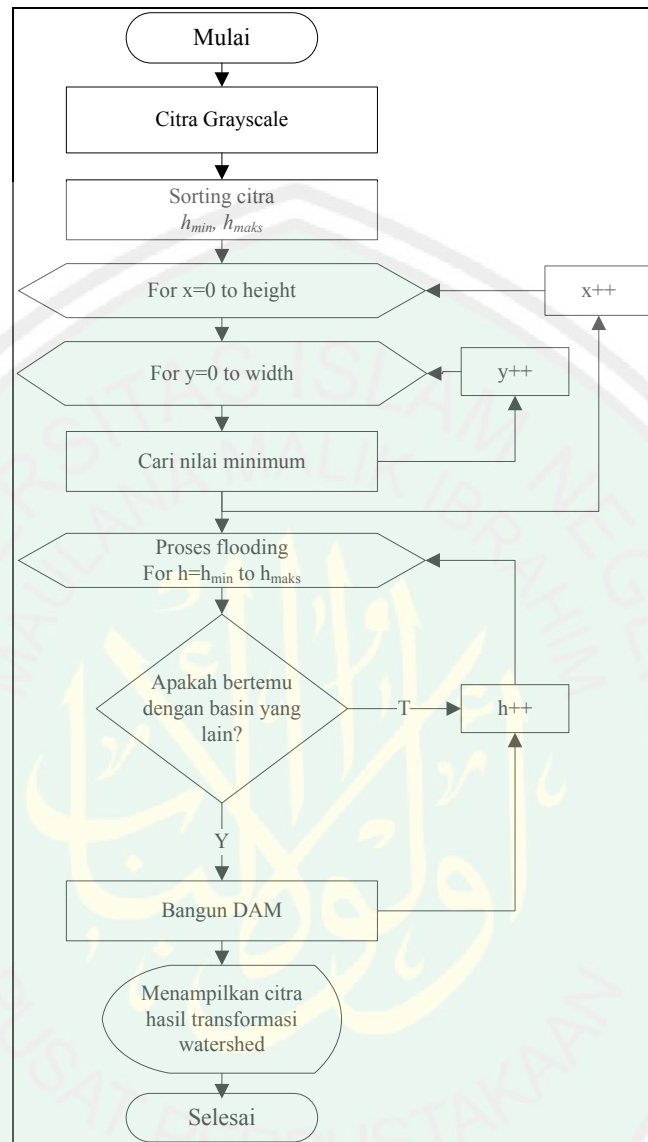
*Watershed* dalam ilmu geografi adalah pertemuan dua lereng yang membagi daerah yang dikeluarkan oleh sistem sungai yang berbeda. *Catchment basin* adalah daerah geografis yang keluar menuju sungai atau *reservoir*. Transformasi *watershed* mengaplikasikan ide ini terhadap pengolahan citra *grayscale* dengan cara yang dapat digunakan untuk menyelesaikan berbagai masalah segmentasi citra.

Suatu citra skala keabuan  $f(x,y)$  dianggap sebagai permukaan topografi  $S$ , di mana tiap skala keabuan dianggap sebagai kemiringan daerah (*terrain elevation*), dan daerah bukit bersesuaian dengan wilayah yang dimaksud, serta lembah yang menunjukkan suatu minimum (*basin*). Misalkan tiap minimum  $m_i(f)$  penuh dengan lubang-lubang dan permukaan topografi  $S$  digambarkan secara vertical ke dalam suatu danau. Diasumsikan dengan kecepatan konstan, air akan mengalir dan memenuhi permukaan. Selama proses pengisian ini, air akan datang dari dua atau lebih minima yang berbeda. Bendungan (*dam*) terbentuk apabila air yang memenuhi dari dua *catchment basin* akan bergabung menjadi satu. Air mengalir mencapai tingkat yang diinginkan dan berhenti mengalir ketika bagian atas dari *dam* terlihat. Gambar berikut memperlihatkan dua minima yang digambarkan sebagai dua basin. *Dam* ditunjukkan dengan warna terang di tengah antara dua basin.



Gambar 3.15 Pengisian permukaan topografi dan pembentukan dam (sumber : Murianto dan Harjoko, 2009).

Hasil dari transformasi *watershed* berupa garis yang membentuk area. Area hasil transformasi *watershed* diurutkan dari yang terbesar hingga terkecil menggunakan metode pengurutan gelembung menurun (*bubble sort descending*) (*flowchart* lihat Gambar 3.16).



Gambar 3.16 *Flowchart* tahapan transformasi watershed

Studi kasus langkah-langkah transformasi watershed ditunjukkan pada Gambar 3.17, menggunakan citra matrik 3x3, dengan tetangga piksel yang ditentukan yaitu sebanyak 4 tetangga yang terhubung, sehingga didapatkan nilai minima pada titik (2,0) dan (2,2) dengan nilai piksel yang sama yaitu 0 yang nantinya akan membentuk dua *catchment basin*, masing-masing area akan diberi label A dan B (sesuai dengan banyak *catchment basin*).

3 2 2	3 2 2	3 2 2	3 B B	B B B
3 1 1	3 1 1	3 W B	3 B B	W B B
0 1 0	A 1 B	A W B	A W B	A W B
(a)	(b)	(c)	(d)	(e)

Gambar 3. 17 Studi kasus transformasi watershed berdasarkan 4 tetangga yang terhubung, (a): Matrik citra asli; (b) Tahap flooding  $h=0$ ; (c) Tahap flooding  $h=1$ ; (d) Tahap flooding  $h=2$ ; (e) Tahap flooding  $h=3$   
(sumber : Roerdink dan Meijster, 2001)

*Sorting* piksel dilakukan untuk membantu tahapan proses *flooding*. Langkah awal menentukan nilai  $h_{min}$  dan  $h_{maks}$  piksel, sehingga didapatkan nilai  $h_{min}=0$  dan  $h_{maks}=3$ . Proses *flooding* dimulai dari flooding tahap 1 dengan  $h=0$ , hingga flooding tahap 4 dengan  $h=3$ .

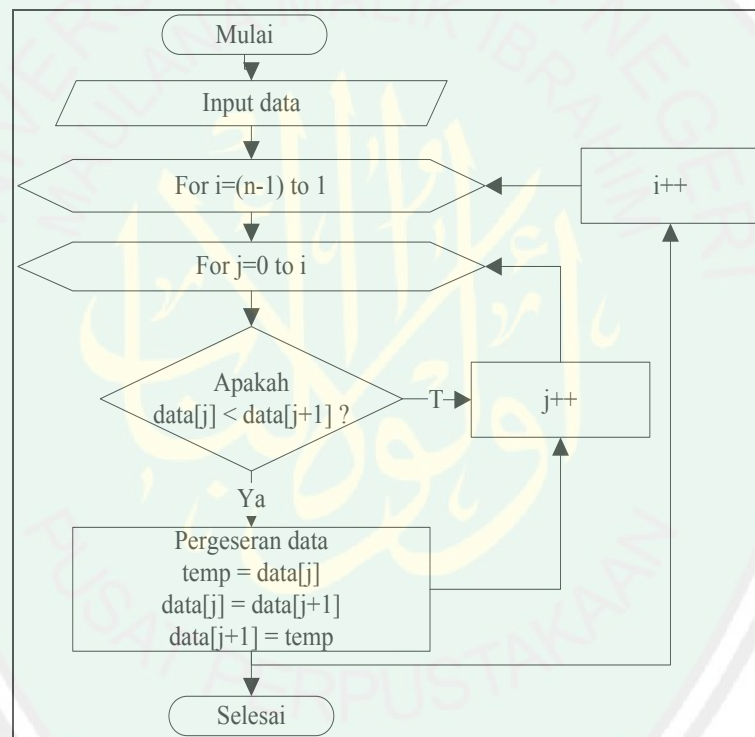
b) *Sorting Region Menggunakan Metode Buble Sort Descending*

Pada metode ini data dianggap sebagai gelembung udara didalam air (Sutanta, 2004). Metode ini bekerja dengan cara membandingkan elemen-elemen yang bersebelahan dan menukarnya jika dalam kondisi tertentu, sehingga jika pengurutan dilakukan secara menurun maka hasil akhir yaitu gelembung udara terbesar akan berada di permukaan air (*flowchart* lihat Gambar 3.18). Berikut langkah-langkah pengurutan gelembung menurun :

- (1). Menentukan nilai  $i$ , langkah awal nilai  $i$  samadengan indeks terbesar dari data, misalkan  $n$  merupakan banyak data jadi  $i$  samadengan  $n-1$ , sedangkan langkah selanjutnya  $i$  akan bernilai menurun dari  $n-1$ ,  $n-2$  hingga bernilai 1.
- (2). Membandingkan data yang bersebelahan mulai dari data ke- $j$  hingga data ke- $i$  ( $j \in 0, 1 \dots i$ ), jadi data ke-0 dibandingkan dengan data ke-1, data ke-1

dibandingkan dengan data-2 hingga data ke  $(i - 1)$  dibandingkan dengan data ke- $i$ .

- (3). Jika data ke- $j$  lebih kecil dari data ke  $(j+1)$ , maka lakukan pergeseran data ke- $j$  samadengan data ke  $j+1$  dan data ke  $(j+1)$  samadengan data ke- $j$ .
- (4). Kembali pada langkah (1),(2), dan (3) hingga  $i$  bernilai 1.



Gambar 3.18 *Flowchart* algoritma *bubble sort descending* (pengurutan gelembung menurun)

Tabel 3.5 Studi kasus pengurutan data dengan *bubble sort descending*

Langkah ke-	Nilai i	Data yang diproses (j=0 hingga j=i)	Hasil Proses					
1	5	j=0 hingga j=5	88	<u>60</u>	71	74	87	<b>54</b>
2	4	J=0 hingga j=4	88	<u>71</u>	74	87	<b>60</b>	<b>54</b>
3	3	j=0 hingga j=3	88	<u>74</u>	87	<b>71</b>	<b>60</b>	<b>54</b>
4	2	j=0 hingga j=2	88	<u>87</u>	<b>74</b>	<b>71</b>	<b>60</b>	<b>54</b>
5	1	j=0 hingga j=1	88	<b>87</b>	<b>74</b>	<b>71</b>	<b>60</b>	<b>54</b>

Studi kasus penerapan metode pengurutan pada enam elemen ( $n$  samadengan 6), yaitu sebagai berikut :

60 88 54 71 74 87

Langkah proses pengurutan dengan menggunakan *buble sort descending* dapat ditelusuri sebagaimana ditunjukkan pada Tabel 3.5. Elemen bercetak tebal merupakan elemen yang telah menempati posisi pengurutan dengan benar, dan elemen yang bergaris bawah merupakan elemen yang akan diurutkan pada langkah selanjutnya.

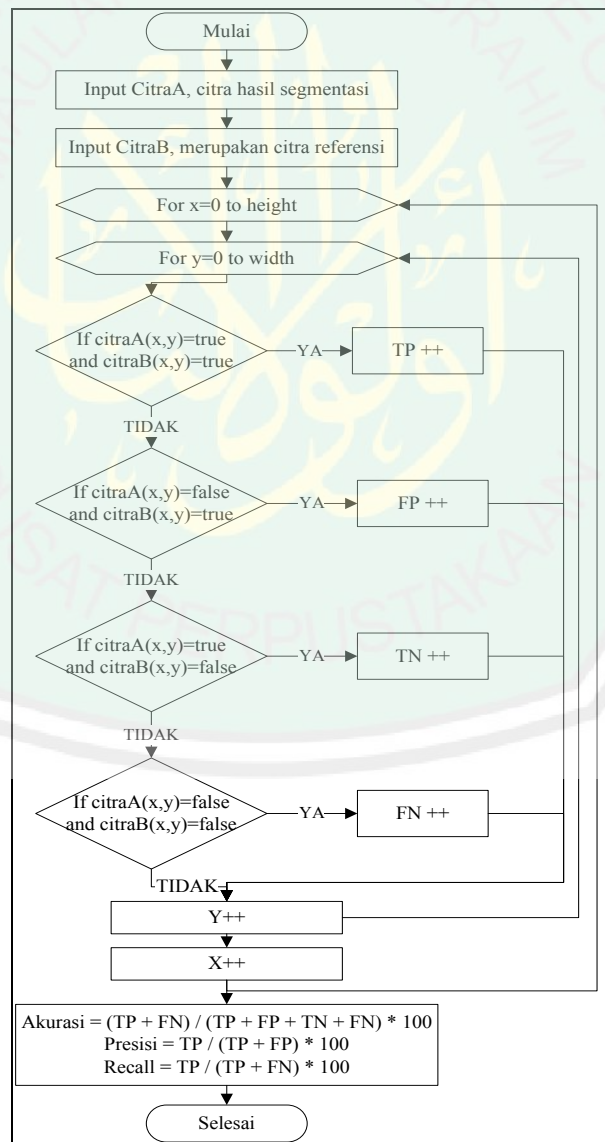
c) Perhitungan Akurasi, Presisi dan *Recall*

Akurasi, presisi dan *recall* merupakan analisis pengukuran kemiripan citra hasil segmentasi dengan citra referensi (*flowchart* lihat Gambar 3.20). Klasifikasi antara dua citra biner (citra biner hasil segmentasi dan citra biner referensi) ada empat kemungkinan kondisi piksel (lihat Gambar 3.19), sebagai berikut:

- (1). *True Positif* (TP), kondisi piksel objek *foramen mentale* diklasifikasikan sebagai piksel objek *foramen mentale*.
- (2). *False Positif* (FP), kondisi piksel bukan objek *foramen mentale* diklasifikasikan sebagai piksel objek *foramen mentale*.
- (3). *True Negative* (TN), kondisi piksel objek *foramen mentale* diklasifikasikan sebagai piksel bukan objek *foramen mentale*.
- (4). *False Negative* (FN), kondisi piksel bukan objek *foramen mentale* diklasifikasikan sebagai piksel bukan objek *foramen mentale*.

		<b>Citra Output Aplikasi</b>	
		Positive (1)	Negative (0)
<b>Citra Referensi</b>	True (1)	True Positive (TP)	True Negative (TN)
	False (0)	False Positive (FP)	False Negative (FN)

Gambar 3.19 Klasifikasi piksel pada citra output aplikasi terhadap citra referensi (sumber : Fawcett, 2003)



Gambar 3.20 Flowchart perhitungan akurasi, presisi dan recall

Akurasi merupakan ukuran ketepatan citra hasil segmentasi (objek dan *background*) terhadap citra referensi. Presisi merupakan ketepatan posisi objek hasil segmentasi terhadap objek pada citra referensi (objek *foramen mentale* pada citra referensi). *Recall* merupakan proporsi objek hasil segmentasi terhadap piksel kebenaran. Perhitungan akurasi, presisi dan *recall* menggunakan persamaan 3.13, 3.14 dan 3.15.

$$Akurasi = \frac{TP + FN}{TP + FP + TN + FN} \times 100 \% \quad (3.13)$$

$$Presisi = \frac{TP}{TP + FP} \times 100 \% \quad (3.14)$$

$$Recall = \frac{TP}{TP + FN} \times 100 \% \quad (3.15)$$

0 0 0 0 0	0 0 0 0 0	FN FN FN FN FN
0 0 1 0 0	0 1 1 1 0	FN FP TP FP FN
0 1 1 1 0	0 1 0 1 0	FN TP TN TP FN
0 0 1 0 0	0 1 1 1 0	FN FP TP FP FN
0 0 0 0 0	0 0 0 0 0	FN FN FN FN FN
(a)	(b)	(c)

Gambar 3.21 (a) Citra biner referensi, (b) Citra biner hasil segmentasi, (c) Kondisi piksel hasil klasifikasi citra biner (a) dan (b)

Studi kasus perhitungan akurasi, presisi, dan *recall*, citra biner referensi ditunjukkan pada Gambar 3.21(a) dan citra biner hasil segmentasi pada Gambar 3.21(b). Hasil klasifikasi piksel ditunjukkan pada Gambar 3.21(c), dengan demikian nilai TP : 4, FP : 4, TN : 1, dan FN : 16, sehingga perhitungan akurasi sebagai berikut.

$$Akurasi = \frac{4 + 16}{4 + 4 + 1 + 16} \times 100 \% = \frac{20}{25} \times 100\% = 80 \%$$

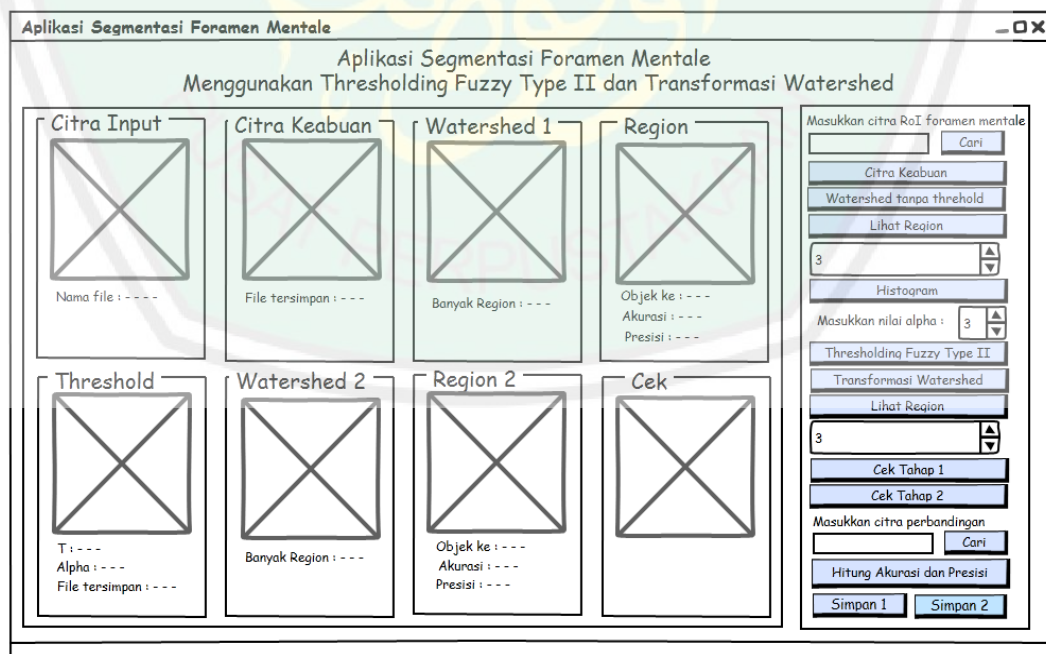
$$\text{Presisi} = \frac{4}{4 + 4} \times 100 \% = \frac{4}{8} \times 100\% = 50 \%$$

$$\text{Recall} = \frac{4}{4 + 16} \times 100 \% = \frac{4}{20} \times 100\% = 10 \%$$

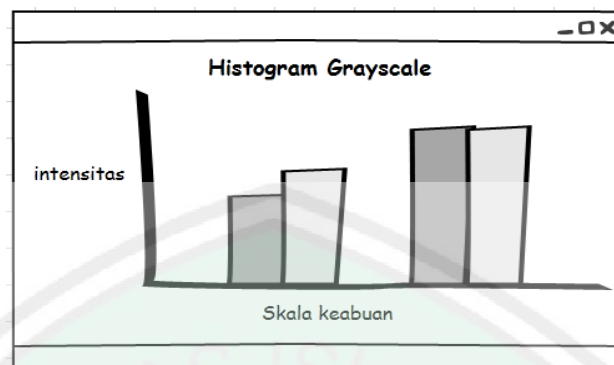
Hasil akhir perhitungan menunjukkan akurasi sebesar 80 %, presisi 50 % dan *recall* 10 %.

### 3.1.2 Desain Antarmuka Aplikasi

Perancangan antarmuka aplikasi bertujuan untuk mempermudah pengguna dalam menggunakan aplikasi. Peneliti merancang antarmuka meliputi *input* citra RoI *foramen mentale* hingga tahapan segmentasi *foramen mentale*. Pada rancangan aplikasi menggunakan dua *frame* yaitu *frame* utama dan *frame* histogram.



Gambar 3.22 Perancangan antarmuka *frame* histogram



Gambar 3.23 Perancangan antarmuka *frame* utama aplikasi

*Frame* histogram merupakan antarmuka untuk menampilkan grafik histogram pada skala keabuan (lihat Gambar 3.23), sedangkan *frame* utama merupakan antarmuka yang menyediakan berbagai proses tahapan segmentasi (lihat Gambar 3.22). Pada *frame* utama terdapat *panel* untuk menampilkan hasil proses citra pada masing-masing tahapan dan masing-masing *panel* terdapat *label* untuk menampilkan informasi citra. Ada empatbelas *button* pada *frame* utama yang memiliki aksi yang berbeda-beda, yaitu sebagai berikut.

- a) *Button1* (*button* dengan nama *cari*), merupakan *button* untuk membuka *directory* guna menginputkan citra masukan. Ada dua *button* untuk membuka *directory* yaitu *Button1* dan *Button11*, *Button1* untuk menginputkan citra ROI *foramen mentale* yang ditampilkan pada *panel* citra masukan, sedangkan *Button11* untuk menginputkan citra referensi yang pada tahapan selanjutnya digunakan untuk mengukur akurasi, presisi dan *recall*.
- b) *Button2* (*button* dengan nama *citra keabuan*), memproses perhitungan konversi citra keabuan dan menampilkan hasil konversi pada *panel* citra keabuan.

- c) *Button3* (*button* dengan nama *watershed tanpa threshold*), memproses citra keabuan menggunakan metode transformasi *watershed* dan akan menampilkan hasil segmentasi pada *Panel3* (*panel* dengan nama *watershed 1*) serta dilengkapi dengan *label* yang memberikan informasi jumlah *region* hasil segmentasi.
- d) *Button4* (*button* dengan nama *lihat region*), mengurutkan *region* hasil transformasi *watershed* dari terbesar hingga terkecil, serta menyediakan *ComboBox* untuk memilih *region*.
- e) *Button5* (*button* dengan nama *histogram*), menghitung histogram citra keabuan serta menampilkan diagram histogram pada frame histogram (Gambar 3.20).
- f) *Button6* (*button* dengan nama *thresholding fuzzy type II*), mengeksekusi perhitungan *thresholding fuzzy type II* serta menampilkan citra biner hasil perhitungan.
- g) *Button7* (*button* dengan nama transformasi *watershed*), mensegmentasi citra hasil *thresholding* dengan menggunakan metode transformasi *watershed* serta memberikan informasi jumlah *region* hasil transformasi *watershed* pada *label*.
- h) *Button8* (*button* dengan nama *lihat region*), mengurutkan *region* hasil transformasi *watershed* dengan menggunakan *thresholding fuzzy type II* dari terbesar hingga terkecil, serta menyediakan *ComboBox* untuk memilih *region*.
- i) *Button9* (*button* dengan nama *cek tahap 1*) mengecek posisi *region* hasil transformasi *watershed* tanpa *thresholding* pada citra asli.

- j) *Button10* (*button* dengan nama *cek tahap 1*) mengecek posisi *region* hasil transformasi *watershed* dengan *thresholding fuzzy type II* pada citra asli.
- k) *Button12* (*button* dengan nama *hitung akurasi presisi dan recall*), mengukur nilai akurasi, presisi dan *recall* pada *region* hasil transformasi *watershed* tanpa *thresholding* yang informasi hasil perhitungan ditampilkan pada *Panel4* dan *region* hasil transformasi *watershed* dengan *thresholding fuzzy type II* yang informasi hasil perhitungan ditampilkan pada *Panel7*.
- l) *Button13* (*button* dengan nama *simpan 1*), berfungsi menyimpan hasil *region* pada transformasi *watershed* tanpa *thresholding*.
- m) *Button14* (*button* dengan nama *simpan 2*), berfungsi menyimpan hasil *region* pada transformasi *watershed* dengan *thresholding fuzzy type II*.

### 3.2 Implementasi Aplikasi

Implementasi aplikasi terbagi menjadi tiga tahapan yaitu *preprocessing*, *main-processing* dan *post-processing*. Implementasi aplikasi dibangun dengan menggunakan bahasa pemrograman java, namun tahapan *post-processing* pada penerapan transformasi *watershed* menggunakan *script matlab* dengan bantuan *matlabcontrol-4.0.0.jar*. Rancang bangun aplikasi ini menggunakan dua kelas yaitu :

- a) Kelas *SegFM\_GUI.java*, mengatur antarmuka aplikasi dan berfungsi sebagai *main class* yaitu mengeksekusi *method* yang ada pada kelas *ProsesCitra.java*

- b) Kelas *ProsesCitra.java*, berisi berbagai *procedure* dan fungsi (dalam bahasa pemrograman java dikenal sebagai *method*) yang akan dieksekusi pada kelas *SegFM\_GUI.java*.

### 3.2.1 Implementasi *Preprocessing*

#### 3.2.1.1 Implementasi Konversi Citra Keabuan

Citra masukan berupa citra RGB sehingga perlu adanya fungsi dalam program untuk mengkonversi citra menjadi citra keabuan guna mempermudah perhitungan pada *main processing*. Antarmuka konversi citra keabuan ditunjukkan pada Gambar 3.24.



Gambar 3.24 Antarmuka proses konversi citra keabuan

Pada kelas *SFM\_GUI.java* *JButton* dengan nama *btn\_keabuan* pada kelas *SegFM\_GUI.java* mengeksekusi method *RGB2Gray(int value[][][])* menggunakan *sourcecode* pada Gambar 3.25. Fungsi *RGB2Gray(int value[][][])* berisi *sourcecode* konversi citra keabuan (*sourcecode* pada Gambar 3.26).

```

private void btn_keabuanActionPerformed(java.awt.event.ActionEvent
evt) {
// TODO add your handling code here:
    try {
//-----Memanggil method RGB2Gray() pada kelas ProsesCitra.java----
        gray2D=pc.RGB2Gray(citra1);
//-----Menampilkan hasil grayscale pada antarmuka aplikasi-----
        ImageIcon icon = new ImageIcon(pc.view_int2D(gray2D));
        lbl_img_keabuan.setIcon(icon);
        repaint();
//-----Mencetak citra grayscale-----
        String namafilebaru=pc.output_gray(gray2D, nama_tnp_eks);
        lbl_namagrayscale.setText(namafilebaru);
    } catch (Exception e) {
        System.out.println("Belum input citra");
    }
}

```

Gambar 3.25 Sourcecode konversi keabuan pada kelas *SegFM\_GUI.java*

```

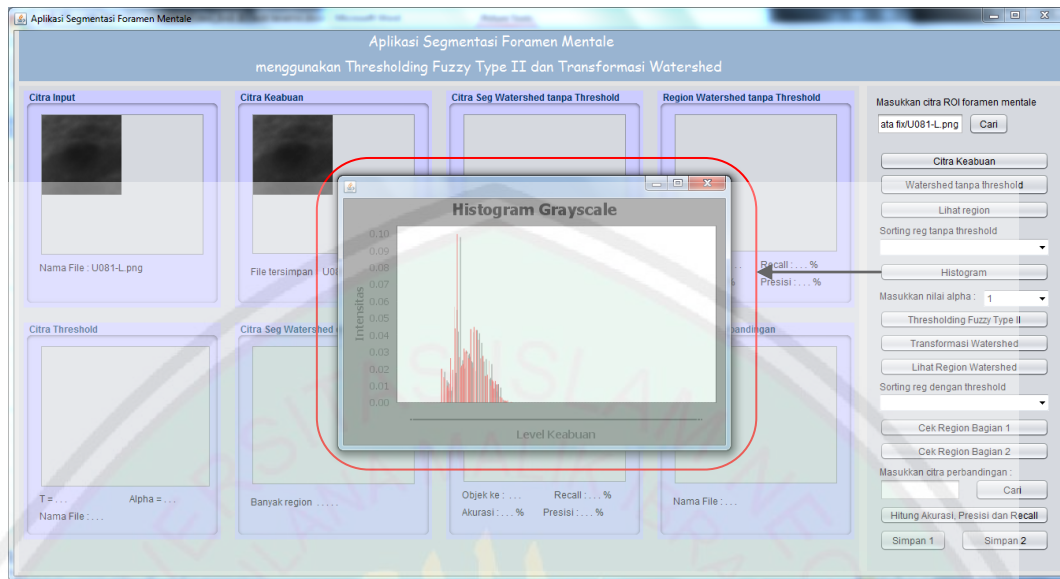
public int[][] RGB2Gray(short value[][][]){
    int height_val=value[0].length;
    int width_val = value[0][0].length;
    int size = height_val * width_val;
    int gray_int[] = new int[size];
    int gray_int2D[][] = new int[height_val][width_val];
    int total = 0;
    double r;
    double g;
    double b;
    for (int x=0; x<height_val; x++){
        for(int y=0; y<width_val; y++){
            r = (double) (0.299f * value[0][x][y]);
            g = (double) (0.587f * value[1][x][y]);
            b = (double) (0.114f * value[2][x][y]);
            gray_int[total] = (int) (r+g+b);
            gray_int2D[x][y]=(int) (r+g+b);
            total++;
        }
    }
    return gray_int2D;
}

```

Gambar 3.26 Sourcecode konversi keabuan pada kelas *ProsesCitra.java*

### 3.2.1.2 Implementasi Histogram Keabuan

Perhitungan *histogram* berada pada kelas *ProsesCitra.java* dengan nama fungsi *histogram(int gray[][])* (sourcecode pada Gambar 3.29) dan fungsi ini dieksekusi jika *button btn\_histogram* diaktifkan (sourcecode pada Gambar 3.28).



Gambar 3.27 Antarmuka proses *histogram grayscale*

```

public double[] histogram(int[][] gray) {
    int size = width * height;
    double[] n=new double[256];
    double jum_prob=0;
    //*****Mengelompokkan piksel*****
    for(int x=0; x< gray.length;x++){
        for(int y=0; y < gray[0].length; y++){
            int i= gray[x][y];
            double ni_temp = n[i];
            double ni = ni_temp+1;
            n[i]=ni;
        }
    }
    //*****Menghitung intensitas piksel*****
    for(int i=0; i< hist.length; i++){
        double ni= n[i];
        double hi= ni/ size;
        hist[i]=hi;
        jum_prob += hi;
    }
    return hist;
}

```

Gambar 3.28 *Sourcecode* fungsi histogram pada kelas *ProsesCitra.java*

```

private void btn_lihathistActionPerformed
(java.awt.event.ActionEvent evt) {
    // TODO add your handling code here:
    hist1D=pc.histogram(gray2D);
    pc.view_chart(hist1D);
}

```

Gambar 3.29 *Sourcecode* eksekusi fungsi histogram pada *SegFM\_GUI.java*

### 3.2.2 Implementasi *Main processing*

*Input* dari proses ini yaitu nilai *alpha* dan histogram citra *grayscale*. Nilai *alpha* diinputkan melalui *ComboBox* dengan nama variabel *choice\_alpha*. Gambar 3.30 menampilkan citra *biner* hasil *thresholding* pada *label* dengan nama variabel *lbl\_img\_threshold*.



Gambar 3.30 Antarmuka proses *thresholding fuzzy type II*

Proses *threshold* berada pada *Button btn\_threshold*, *btn\_threshold* merupakan implementasi algoritma *thresholding fuzzy type II*, sehingga *btn\_threshold* pada kelas *SegFM\_GUI.java* (*sourcecode* pada Gambar 3.31) mengeksekusi beberapa *method* pada kelas *ProsesCitra.java* yang terbagi menjadi delapan tahap (*sourcecode* pada Gambar 3.32).

```
private void btn_thresholdActionPerformed (java.awt.event.
ActionEvent evt) {
// TODO add your handling code here:
try {
//==Tahap 1 == Menentukan nilai alpha=====
String str_alpha = choice_alpha.getSelectedItem();
int alpha = Integer.parseInt(str_alpha);
```

```

pc.menentukan_nilai_alpha(alpha);
//==Tahap 2 == Menentukan pergeseran MF objek dan background=====
//----- Menentukan batas fungsi keanggotaan -----
int vO_min = 52;
int vO_max= 73;
int vO_hmax = 57;
double hO_max =1;
int vB_min = 50;
int vB_max= 104;
int vB_hmax = 76;
double hB_max =1;
pc.menentukan_MF_Objek(vO_min, vO_max, vO_hmax, hO_max);
pc.menentukan_MF_Background(vB_min, vB_max, vB_hmax,
hB_max);
//----- Menentukan batas pergeseran MF O dan MF B -----
pc.hitung_gmax_gmin(hist1D);
int g_min = pc.get_g_min();
int g_max = pc.get_g_max();
pc.menentukan_pergeseranMFO_MFB(g_min, g_max);
int g_mulai= pc.get_g_mulai();
int g_akhir = pc.get_g_akhir();
int juml_perg = g_akhir - (g_mulai-1);
//-----Inisial variabel untuk menyimpan nilai-----
double [][] ultrafuzziness_tot=new double[2][juml_perg];
double ultrafuzzi_B=0;
double ultrafuzzi_O=0;
double ultrafuzzi_total=0;
for(int j =1; j <= juml_perg; j++){
    int g_j = g_mulai+(j-1);
//== Tahap 3 = Himpunan Fuzzy -> batas atas dan batas bawah=====
    double himpunan_O[][]=pc.himpunan_fuzzy_MFO(j,hist1D);
    double himpunan_B[][]=pc.himpunan_fuzzy_MFB(j,hist1D);
//== Tahap 4 = Menghitung ultrafuzziness Objek =====
    ultrafuzzi_O=pc.fungsi_ultrafuzziness(himpunan_O,hist1D);
//== Tahap 5 = Menghitung ultrafuzziness Background =====
    ultrafuzzi_B=pc.fungsi_ultrafuzziness(himpunan_B,
hist1D);
//== Tahap 6 = Menghitung ultrafuzziness total =====
    ultrafuzzi_total=ultrafuzzi_O * ultrafuzzi_B;
    ultrafuzziness_tot[0][j-1]=g_j;
    ultrafuzziness_tot[1][(j-1)]=ultrafuzzi_total;
}
//== Tahap 7 = Menentukan nilai maksimal ultrafuzziness total ====
T = pc.ultrafuzziness2D_maks(ultrafuzziness_tot);
System.out.println("Nilai T "+T);
//== Tahap 8 = Thresholding citra dengan nilai T =====
biner = pc.Thresholding_Image(T, gray2D);
//----- Menampilkan pada antarmuka aplikasi -----
ImageIcon icon = new ImageIcon(pc.view_biner(biner));
lbl_img_threshold.setIcon(icon);
repaint();
//----- Mencetak citra biner hasil thresholding -----
String nama_output_threshold=pc.output_gambar_biner(biner,
nama_tnp_eks, "-03thresh");
lbl_alpha.setText("alpha="+alpha);
lbl_T.setText("T="+T);

```

```

    } catch (Exception e) {
        JOptionPane.showMessageDialog(null, "Histogram citra input
        tidak termasuk dalam membership function.", null, 1);
        System.out.println("ERROR "+e);
    }
}

```

Gambar 3.31 *Sourcecode* eksekusi fungsi *thresholding* pada kelas *SegFM\_GUI.java*

```

//=====
//==== START >> PROSES THRESHOLDING FUZZY TYPE II =====
//=====
public void menentukan_MF_Objek(int vOmin,int vOmax,int vOhmax,
double hOmax){
    this.vO_min=vOmin;
    this.vO_max = vOmax;
    this.vO_hmax=vOhmax;
    this.hO_max=hOmax;
}

public void menentukan_MF_Background(int vBmin,int vBmax,int
vBhmax, double hBmax){
    this.vB_min=vBmin;
    this.vB_max = vBmax;
    this.vB_hmax=vBhmax;
    this.hB_max=hBmax;
}

public void menentukan_nilai_alpha(int alph){
    this.alpha=alph;
}

public void hitung_gmax_gmin(double [] hist){
    int byk_data=0;
    for(int i=0; i < hist.length; i++){
        if(hist[i] != 0){
            byk_data++;
        }
    }
    double [][] hist_new = new double[2][byk_data];
    int index=0;
    for(int i=0; i < hist.length; i++){
        if(hist[i] != 0){
            hist_new[0][index]=i;
            hist_new[1][index]=hist[i];
            index++;
        }
    }
    this.g_min = (int) hist_new[0][0];
    this.g_max = (int) hist_new[0][byk_data-1];
}

public int get_g_min(){
    return g_min;
}

public int get_g_max(){
    return g_max;
}

```

```

}
public void menentukan_pergeseranMFO_MFB(int gmin, int gmax){
    g_mulai=gmin;
    g_akhir=gmax;
}

public int get_g_mulai(){
    return g_mulai;
}

public int get_g_akhir(){
    return g_akhir;
}

public double[][] himpunan_fuzzy_MFO(int j, double hist[]){
    double himpunan_O[][]=new double[3][256];
    int a = vO_min;
    int c =g_mulai+(j-1);
    int ac = c-a;
    double ab = ((vO_hmax-vO_min)*ac) / (vO_max - vO_min);
    double b = a + ab;
    double mu=0;
    double mu_Up = 0;
    double mu_Low = 0;
    for(int g=0; g < 256; g++){
        double p=0;
        if( g >= a && g <= b){
            p=((g-a)*hO_max)/(b-a);
            if(hist[g] < p){
                mu=hist[g];
            }else if(hist[g]>=p){
                mu=p;
            }
        }else if(g > b && g <=c){
            p=((c-g)* hO_max)/(c-b);
            if(hist[g] < p){
                mu=hist[g];
            }else if(hist[g]>=p){
                mu=p;
            }
        }else{
            mu=0;
        }
        mu_Up = Math.pow(mu, (1/alpha));//1
        mu_Low = Math.pow(mu, alpha);//0
        himpunan_O[0][g]=mu;
        himpunan_O[1][g]=mu_Up;
        himpunan_O[2][g]=mu_Low;
    }
    return himpunan_O;
}
//***** Menghitung himpunan fuzzy Background *****
public double[][] himpunan_fuzzy_MFB(int j, double hist[]){
    double himpunan_B[][]= new double[3][256];
    int a = g_mulai+(j-1);
    int c =vB_max;

```

```

int ac = c-a;
double ab = ((vB_hmax-vB_min)*ac)/(vB_max - vB_min);
double b = a + ab;
double mu=0;
double mu_Up = 0;
double mu_Low = 0;
for(int g=0; g < 256; g++){
    double p=0;
    if( g >= a && g <= b){
        p=((g-a)*hB_max)/(b-a);
        if(hist[g] < p){
            mu=hist[g];
        }else if(hist[g]>=p){
            mu=p;
        }
    }else if(g > b && g <=c){
        p=((c-g)* hB_max)/(c-b);
        if(hist[g] < p){
            mu=hist[g];
        }else if(hist[g]>=p){
            mu=p;
        }
    }else{
        mu=0;
    }
    mu_Up = Math.pow(mu, (1/alpha)); //1
    mu_Low = Math.pow(mu, alpha); //0
    himpunan_B[0][g]=mu;
    himpunan_B[1][g]=mu_Up;
    himpunan_B[2][g]=mu_Low;
}
return himpunan_B;
}
public double fungsi_ultrafuzziness(double [][]himpunan, double[]
hist){
    double ultrafuzziness=0;
    int M = height;
    int N = width;
    int L=256;
    double Sigma=0;
    double n_g=0;
    for(int g=0; g <=(L-1); g++ ){
        double mu_Up=himpunan[1][g];
        double mu_Low=himpunan[2][g];
        n_g=hist[g]*(Math.pow(Math.E, (mu_Up-mu_Low)));
        Sigma += n_g;
    }
    ultrafuzziness=Sigma/(M*N);
    return ultrafuzziness;
}

public int ultrafuzziness2D_maks(double[][] total_ultrafuzzines){
    int T=0;
    double temp_maks=0;
    int g_opt=0;
    for(int i=0; i < total_ultrafuzzines[0].length; i++){

```

```

        double temp_ultr = total_ultrafuzzines[1][i];
        if(temp_ultr > temp_maks){
            temp_maks = temp_ultr;
            g_opt = (int) total_ultrafuzzines[0][i];
        }
    }
    T=g_opt;
    return T;
}
public int[][] Thresholding_Image(int T,int [][] gray){
    int[][] biner = new int [height][width];
    for(int x=0; x < height ; x++){
        for(int y=0; y < width; y++){
            if(gray[x][y] <= T){
                biner[x][y]=1;
            }else{
                biner[x][y]=0;
            }
        }
    }
    return biner;
}
//=====
//==== END >> PROSES THRESHOLDING FUZZY TYPE II =====
//=====

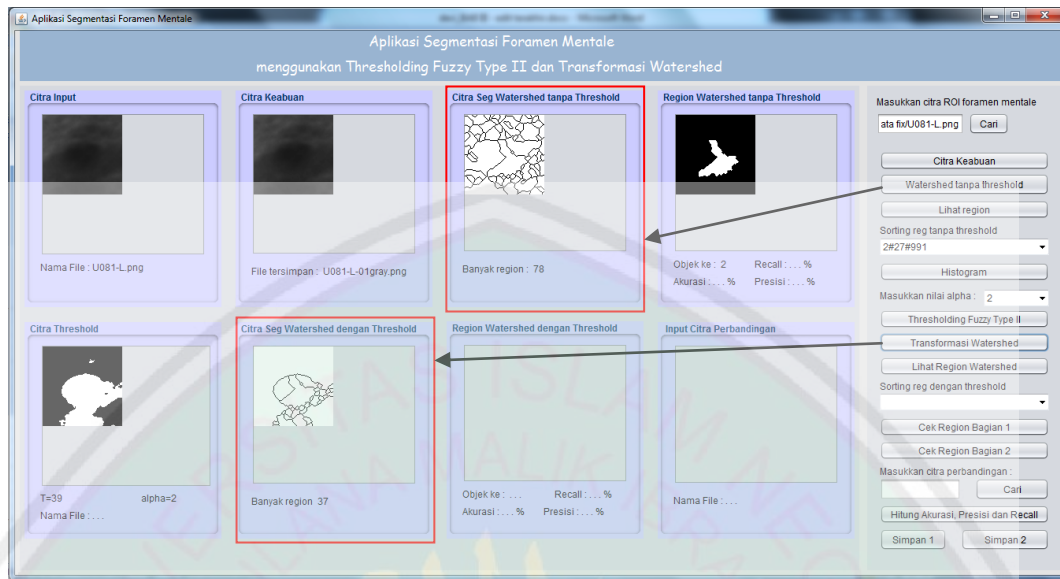
```

Gambar 3.32 Sourcecode thresholding fuzzy type II pada kelas *ProsesCitra.java*

### 3.2.3 Implementasi *Post Processing*

#### 3.2.3.1 Implementasi Transformasi *Watershed*

Implementasi Transformasi *watershed* menggunakan fungsi *matlab* namun dijalankan pada bahasa pemrograman *java* dengan menggunakan *matlabcontrol-4.0.0.jar*. Gambar 3.33 menunjukkan antarmuka hasil segmentasi menggunakan metode transformasi *watershed*. Sourcocode untuk menjalan *script matlab* pada kelas *ProsesCitra.java* ditunjukkan pada Gambar 3.34.



Gambar 3.33 Antarmuka proses transformasi *watershed* tanpa *thresholding* dan transformasi *watershed* dengan *thresholding fuzzy type II*

```

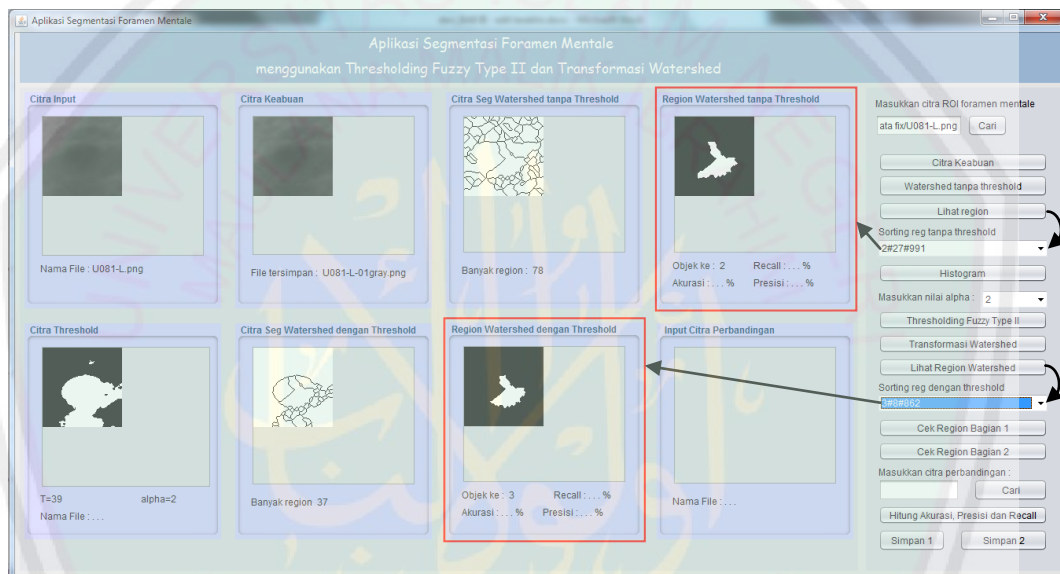
//***** Transformasi watershed dg menggunakan script MATLAB *****
public double[][] watershed(double[][] piksel) throws
MatlabConnectionException
{
    double[][] segmen = new double[height][width];
    try
    {
        MatlabProxyFactoryOptions option = new
        MatlabProxyFactoryOptions.Builder().setUsePreviouslyControll
        edSession(true).build();
        MatlabProxyFactory factory= new MatlabProxyFactory(option);
        MatlabProxy proxy = factory.getProxy();
        MatlabTypeConverter processor = new
        MatlabTypeConverter(proxy);
        processor.setNumericArray("A", new
        MatlabNumericArray(piksel, null));
        proxy.eval("w=watershed(A,4);");
        segmen = processor.getNumericArray("w").getRealArray2D();
        proxy.disconnect();
    }
    catch (MatlabInvocationException ex)
    {
        Logger.getLogger(ProsesCitra.class.getName()).log(Level.SEVERE,
        null, ex);
    }
    return segmen;
}
//=====

```

Gambar 3.34 Sourcecode memanggil fungsi *matlab* pada kelas *ProsesCitra.java*

### 3.2.3.2 Implementasi *Burble Sort Descending*

Tahapan ini mengurutkan *region* hasil transformasi *watershed* dari terbesar hingga terkecil dan menampilkan *region* pada antarmuka aplikasi. *Sourcecode* yang mendukung proses ini berada pada kelas *ProsesCitra.java* yang ditunjukkan pada Gambar 3.36.



Gambar 3.35 Antarmuka sorting *region* hasil transformasi *watershed*

```
//=====
//=== START >> PROSES SORTING REGION WATERSHED =====
//=====
//***** Mengelompokkan piksel *****
public int[][] pengelompokan_piksel(int maks, double[][] segmen)
{
    System.out.println("Cetak nilai maksimal "+maks);
    int[][] piksel_wtrshd = new int[2][maks+1];
    for(int x=0; x<segmen.length; x++)
    {
        for(int y=0; y < segmen[0].length; y++)
        {
            int idx= (int) (segmen[x][y]);
            if(idx != 0){
                int val = piksel_wtrshd[1][idx];
                int val_temp = val+1;
                piksel_wtrshd[1][idx]=val_temp;
            }
        }
    }
    return piksel_wtrshd;
}
```

```

}
//***** Mengambil nilai piksel dg mengurutkan juml piksel (desc)*
public int[] get_sort_piksel(int[][] value)
{
    int sort[]=new int[(value[0].length)-1];
    for(int i=0; i< sort.length; i++)
    {
        sort[i]=i+1;
    }
    for(int i=(sort.length-1); i>=0 ; i--)
    {
        int val_i=value[1][sort[i]];
        for(int p=0; p<i; p++)
        {
            int val_p = value[1][sort[p]];
            int val_q = value[1][sort[p+1]];
            if(val_p < val_q)
            {
                int sort_temp = sort[p];
                sort[p]=sort[p+1];
                sort[p+1]=sort_temp;
            }
        }
    }
    return sort;
}
//***** Menampilkan region pada antarmuka aplikasi *****
public Image set_piksel_img(int pxl, double[][]value)
{
    int display[]=new int [value.length * value[0].length];
    int idx =0;
    for(int x=0; x< value.length; x++)
    {
        for(int y=0; y< value[0].length; y++)
        {
            int i = (width * x) + y;
            if(value[x][y] == pxl)
            {
                display[i]=255;
            }else
            {
                display[i]=0;
            }
        }
    }
    ImageProducer ip = new MemoryImageSource(width, height,
    grayModel, display, 0, width);
    return Toolkit.getDefaultToolkit().createImage(ip);
}
//=====
//=== END >> PROSES SORTING REGION WATERSHED =====
//=====

```

Gambar 3.36 Sourcecode bubble sort descending pada ProsesCitra.java

### 3.2.3.3 Implementasi Akurasi, Presisi, dan Recall

Pengukuran ini melibatkan dua citra yaitu citra hasil segmentasi dan citra referensi. Citra referensi merupakan citra biner yang telah mengsegmentasi *foramen mentale* secara manual dengan aplikasi tambahan.



Gambar 3. 37 Antarmuka proses pengukuran akurasi, presisi dan *recall* (bagian 1 dan bagian 2)

```
//=====
//=== START >> HITUNG AKURASI, AKURASI, DAN RECALL =====
//=====
//***** Menghitung tn, tp, fn, dan fp *****
public void ReceiverOperatingCharecteristic(boolean citra_A[][],
boolean citra_B[][]) {
    tn=0; tp=0; fn=0; fp=0;
    for(int x=0; x < citra_A.length; x++){
        for(int y=0; y < citra_A[0].length; y++){
            if(citra_A[x][y]==true && citra_B[x][y]==false){
                //true-negative
                this.sigma_TN++;
            }else if(citra_A[x][y]==true && citra_B[x][y]==true){
                //true-positive
                this. sigma_TP++;
            }else if(citra_A[x][y]==false && citra_B[x][y]==true){
                //false-positive
                this. sigma_FP++;
            }else if(citra_A[x][y]==false && citra_B[x][y]==false){
                //true-negative
                this. sigma_FN++;
            }
        }
    }
}
```

```

    tn= sigma_TN; tp= sigma_TP; fp= sigma_FP; fn= sigma_FN;
}
//***** Mengukur akurasi citra *****
public double hitung_akurasi(){
    double akurasi=(double)((tp + fn)*100)/(tp+fp+fn+tn);
    int decimalPlace=2;
    BigDecimal bd1 = new BigDecimal(akurasi);
    bd1=bd1.setScale(decimalPlace, BigDecimal.ROUND_UP);
    akurasi=bd1.doubleValue();
    return akurasi;
}
//***** Mengukur Presisi Citra *****
public double hitung_presisi(){
    double presisi= (double) (tp* 100 )/ (tp + fp);
    int decimalPlace=2;
    BigDecimal bd1 = new BigDecimal(presisi);
    bd1=bd1.setScale(decimalPlace, BigDecimal.ROUND_UP);
    presisi=bd1.doubleValue();
    return presisi;
}
//***** Mengukur Recall Citra *****
public double hitung_recall(){
    double recall=(double)(tp*100)/(tp+fn);
    System.out.println(recall);
    int decimalPlace=2;
    BigDecimal bd1 = new BigDecimal(recall);
    bd1=bd1.setScale(decimalPlace, BigDecimal.ROUND_UP);
    recall=bd1.doubleValue();
    return recall;
}
//=====
//=== END >> HITUNG AKURASI, AKURASI, DAN RECALL =====
//=====

```

Gambar 3.38 Sourcecode mengukur akurasi, presisi, dan recall pada kelas *ProsesCitra.java*

*Method ReceiverOperatingCharecteristik(boolean[][] citra\_A, boolean[][] citra\_B)* pada kelas *ProsesCitra.java* berfungsi untuk menghitung nilai *true-positive* (tp), *true-negative* (tn), *false-positive* (fp), dan *false-negative* (fn). Ada tiga kriteria yang diukur pada proses ini yaitu akurasi, presisi dan *recall*. Pengukuran akurasi berada pada *method hitung\_akurasi()*, pengukuran presisi berada pada *method hitung\_presisi()*, dan pengukuran *recall* berada pada *method hitung\_recall()* (sourcecode Gambar 3.38).

## BAB IV

### HASIL DAN PEMBAHASAN

Menjelaskan rangkaian uji coba terhadap aplikasi segmentasi *foramen mentale* menggunakan *thresholding fuzzy type II* dan transformasi *watershed*. Ujicoba ditujukan untuk mengetahui sejauh mana keberhasilan dari metode yang digunakan pada aplikasi yang dibangun, selanjutnya menganalisa hasil uji coba sehingga mendapatkan kesimpulan hasil pengukuran akurasi, presisi, recall dan kesesuaian objek terhadap *foramen mentale*.

#### 4.1 Langkah-langkah Uji Coba

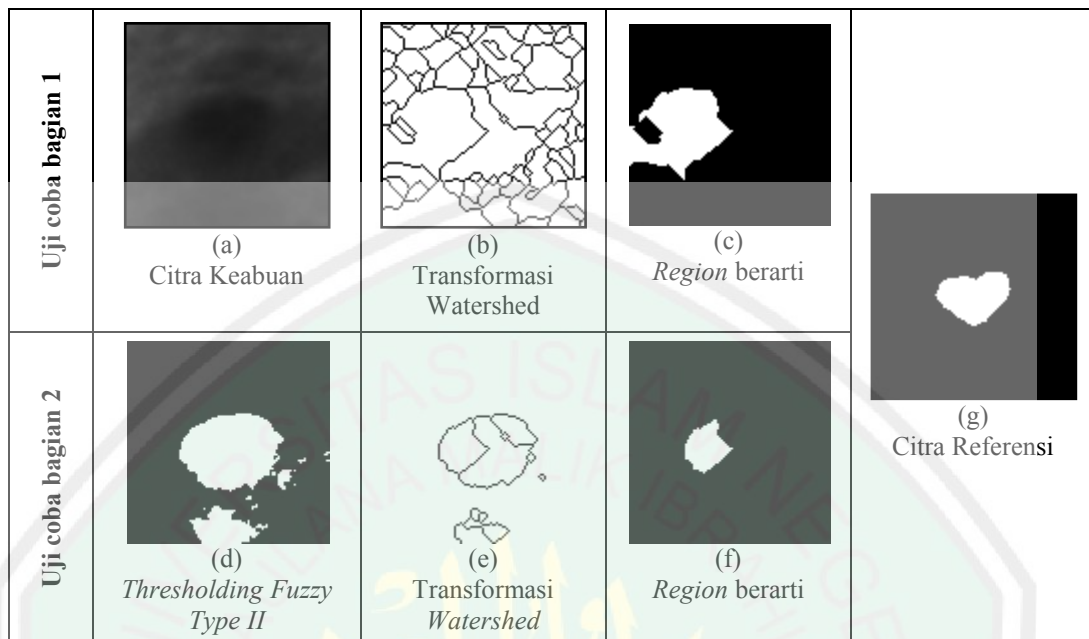
Langkah-langkah uji coba pada penelitian ini sebagai berikut.

- a) Menyiapkan data citra DPR sebanyak 20 citra yang termasuk dalam kelompok osteoporosis.
- b) *Cropping*. RoI *foramen mentale* didapatkan dengan memotong bagian penting citra DPR secara manual, sehingga mendapat dua bagian yaitu RoI foramen mentale kanan dan kiri, sehingga jumlah citra uji menjadi 40 citra RoI.
- c) Citra keabuan. Konversi citra masukan menjadi citra keabuan yang tersedia pada antarmuka perangkat lunak.
- d) Histogram. Perhitungan histogram yang dilakukan oleh perangkat lunak, yang digunakan untuk perhitungan himpunan *fuzzy type II*.

- e) *Thresholding*. Metode *thresholding* yang digunakan yaitu *thresholding* himpunan *fuzzy type II*.
- f) Segmentasi. Metode segmentasi yang digunakan yaitu transformasi *watershed*.
- g) *Sorting*. Mengurutkan objek berdasarkan jumlah piksel yang terbesar hingga terkecil.
- h) Analisa. Menganalisa *region* mana yang paling berarti (*foramen mentale*)
- i) Identifikasi *foramen mentale* terhadap citra hasil segmentasi
- j) *Input* citra referensi. Citra referensi merupakan citra biner yang telah mensegmentasi objek *foramen mentale* secara manual dengan bantuan aplikasi lain guna membantu pengukuran akurasi, presisi, dan *recall*.
- k) Akurasi. Mengukur akurasi citra hasil segmentasi terhadap citra referensi.
- l) Presisi. Menghitung presisi objek *foramen mentale* pada citra referensi.
- m) *Recall*. Mengukur proporsi objek terhadap piksel kebenaran.

#### 4.2 Hasil Uji Coba

Aplikasi akan mengidentifikasi foramen secara otomatis dan tahap akhir berupa pengukuran akurasi, presisi dan *recall* citra hasil segmentasi terhadap citra referensi. Proses uji coba akan melakukan pengujian satu persatu terhadap citra RoI *foramen mentale* sebanyak 40 citra RoI *foramen mentale* dengan kondisi kelompok osteoporosis (lihat Lampiran 1).



Gambar 4.1 Hasil uji coba aplikasi pada satu citra

Hasil uji coba pada Gambar 4.2 (a), (b) dan (c) merupakan hasil uji coba segmentasi tanpa perlakuan *thresholding fuzzy type II* (bagian 1), sedangkan Gambar 4.2 (d), (e) dan (f) merupakan hasil uji coba segmentasi dengan perlakuan *thresholding fuzzy type II* (bagian 2). Gambar 4.2 (g) merupakan citra referensi guna membantu pengukuran kemiripan citra berdasarkan akurasi, presisi dan recall.

#### 4.3 Pembahasan

Uji coba dilakukan satu persatu terhadap citra ROI *foramen mentale* dan mengidentifikasi *foramen mentale* secara manual. Tabel 4.1 menunjukkan hasil identifikasi foramen mentale berdasarkan citra hasil uji coba, sehingga rata-rata keberhasilan identifikasi foramen mentale sebesar 32.5 % (perhitungan 4.1).

Tabel 4.1 Hasil identifikasi *foramen mentale* terhadap citra hasil uji coba

No	Nama Citra	Keterangan Identifikasi <i>Foramen Mentale</i>
1	U002-L	tidak sesuai
2	U002-R	sesuai
3	U004-L	tidak sesuai
4	U004-R	tidak sesuai
5	U008-L	sesuai
6	U008-R	tidak sesuai
7	U009-L	tidak sesuai
8	U009-R	tidak sesuai
9	U012-L	sesuai
10	U012-R	sesuai
11	U021-L	tidak sesuai
12	U021-R	tidak sesuai
13	U025-L	tidak sesuai
14	U025-R	tidak sesuai
15	U037-L	tidak sesuai
16	U037-R	tidak sesuai
17	U055-L	tidak sesuai
18	U055-R	sesuai
19	U058-L	tidak sesuai
20	U058-R	tidak sesuai
21	U061-L	tidak sesuai
22	U061-R	sesuai
23	U072-L	tidak sesuai
24	U072-R	tidak sesuai
25	U073-L	tidak sesuai
26	U073-R	sesuai
27	U075-L	sesuai
28	U075-R	sesuai
29	U077-L	sesuai
30	U077-R	tidak sesuai
31	U079-L	tidak sesuai
32	U079-R	sesuai
33	U081-L	sesuai
34	U081-R	tidak sesuai
35	U084-L	tidak sesuai
36	U084-R	tidak sesuai
37	U091-L	tidak sesuai
38	U091-R	tidak sesuai
39	U096-L	sesuai
40	U096-R	tidak sesuai

$$\begin{aligned}
 Mean_{kesesuaian\ data} &= \frac{\sum data\ sesuai}{\sum data} \times 100\% & (4.1) \\
 &= \frac{13}{40} \\
 &= 32.5\%
 \end{aligned}$$

Perhitungan akurasi, presisi dan *recall* terbagi menjadi dua bagian yaitu bagian pertama pengukuran pada penerapan segmentasi *watershed* tanpa perlakuan *thresholding fuzzy type II*, dan bagian kedua pengukuran pada penerapan segmentasi *watershed* dengan perlakuan *thresholding fuzzy type II*.

$$\begin{aligned}
 Mean_{Akurasi(bagian1)} &= \frac{\sum Akurasi_{bagian\ 1}}{\sum data} & (4.2) \\
 &= \frac{3815.47}{40} \\
 &= 95.39
 \end{aligned}$$

$$\begin{aligned}
 Mean_{Presisi(bagian1)} &= \frac{\sum Presisi_{bagian\ 1}}{\sum data} & (4.3) \\
 &= \frac{2611.04}{40} \\
 &= 65.28
 \end{aligned}$$

$$\begin{aligned}
 Mean_{Recall(bagian1)} &= \frac{\sum Recall_{bagian\ 1}}{\sum data} & (4.4) \\
 &= \frac{25.96}{40} \\
 &= 0.65
 \end{aligned}$$

Tabel 4.2 Akurasi, presisi dan *recall* citra RoI *foramen mentale* uji coba bagian 1 dan bagian 2

No	Nama Citra	Bagian 1 (tanpa perlakuan <i>thresholding</i> )			Bagian 2 (dengan perlakuan <i>thresholding</i> )		
		Akurasi (%)	Presisi (%)	<i>Recall</i> (%)	Akurasi (%)	Presisi (%)	<i>Recall</i> (%)
1	U002-L	97.57	37.61	0.48	98.1	84.85	0.28
2	U002-R	94.36	28.52	0.7	95.92	91.43	0.66
3	U004-L	96.13	43.59	0.54	96.4	56.05	0.53
4	U004-R	96.25	42.31	0.46	96.55	58.67	0.46
5	U008-L	98.54	58.74	0.38	98.61	67.28	0.38
6	U008-R	97.59	50	0.14	97.69	81.25	0.14
7	U009-L	94.88	55.94	0.35	95.07	86.49	0.34
8	U009-R	91.54	100	0.42	91.54	100	0.42
9	U012-L	94.71	45.84	0.24	95.16	100	0.44
10	U012-R	93.41	96	0.26	93.35	94.45	0.19
11	U021-L	96.36	63.52	0.49	96.32	100	0.17
12	U021-R	92.13	57.38	0.38	92.16	60.35	0.38
13	U025-L	97.69	51.57	0.33	97.78	60	0.33
14	U025-R	97.1	82.98	0.42	97.13	100	0.37
15	U037-L	97.11	55.36	0.32	97.14	60	0.28
16	U037-R	98.51	52.64	0.2	98.64	80	0.2
17	U055-L	96.8	100	0.29	96.8	100	0.29
18	U055-R	96.6	31.25	0.11	96.89	100	0.18
19	U058-L	95.59	75.25	0.78	95.79	94.94	0.77
20	U058-R	95.72	99.01	1.03	94.87	100	0.13
21	U061-L	93.91	61.8	0.58	93.91	61.8	0.58
22	U061-R	93.1	100	0.27	93.1	100	0.27
23	U072-L	96.53	61.67	0.38	96.67	80.44	0.38
24	U072-R	92.53	18.36	0.41	93.63	39.59	0.41
25	U073-L	96.72	76	0.39	96.76	100	0.31
26	U073-R	97.23	71.19	0.85	96.87	100	0.14
27	U075-L	91.6	49.02	0.27	91.76	94.12	0.18
28	U075-R	92.55	100	0.19	92.52	100	0.16
29	U077-L	96.57	77.41	1.67	96.16	100	0.76
30	U077-R	98.91	47.06	0.33	99.22	90.91	0.31
31	U079-L	97.12	65.66	1.33	97.5	86.77	1.2
32	U079-R	94.13	83.93	0.51	94.09	100	0.37
33	U081-L	84.67	17.62	2.8	94.35	63.62	2.46
34	U081-R	94.67	63.73	4.79	96.24	90.89	4.07
35	U084-L	97.11	90.67	0.7	97.06	100	0.58
36	U084-R	94	58.5	0.34	94.02	60.79	0.34
37	U091-L	96.92	100	0.77	96.86	100	0.71
38	U091-R	94.43	100	0.55	94.24	100	0.35
39	U096-L	95.66	40.91	0.28	96	89.66	0.27
40	U096-R	98.52	100	0.23	98.52	100	0.22

Tabel 4.2 menunjukkan hasil uji segmentasi *foramen mentale* dengan menggunakan metode transformasi *watershed* tanpa perlakuan tahapan *thresholding fuzzy type II* (bagian 1) dan dengan perlakuan tahapan *thresholding* (bagian 2), nilai akurasi, presisi dan *recall* citra didapatkan berdasarkan nilai *tp*, *tn*, *fp*, dan *fn* (Lampiran 3). Perhitungan rata-rata nilai akurasi dan presisi bagian 1 ditunjukkan pada persamaan 4.2, 4.3 dan 4.4. Sedangkan perhitungan rata-rata akurasi, presisi dan *recall* bagian 2 ditunjukkan pada persamaan 4.5, 4.6 dan 4.7.

$$Mean_{Akurasi(bagian\ 2)} = \frac{\sum Akurasi_{bagian\ 2}}{\sum data} \quad (4.5)$$

$$= \frac{3831.39}{40}$$

$$= 95.79$$

$$Mean_{Presisi(bagian\ 2)} = \frac{\sum Presisi_{bagian\ 2}}{\sum data} \quad (4.6)$$

$$= \frac{3434.35}{40}$$

$$= 85.86$$

$$Mean_{Recall(bagian\ 2)} = \frac{\sum Recall_{bagian\ 2}}{\sum data} \quad (4.7)$$

$$= \frac{21.01}{40}$$

$$= 0.53$$

Hasil perhitungan rata-rata akurasi segmentasi tanpa perlakuan *thresholding* sebesar 95.39 %, rata-rata presisi 65,28 % dan rata-rata *recall* 0.65%.

Hasil perhitungan rata-rata akurasi segmentasi dengan perlakuan thresholding sebesar 95.79%, rata-rata presisi 85.86% dan rata-rata *recall* 0.53%.

#### 4.4 Integrasi Metode *Thresholding Fuzzy Type II* dengan Al-qur'an

Metode *thresholding* dalam pengolahan citra yaitu mengelompokkan piksel menjadi dua kelompok yaitu kelompok piksel objek dan kelompok piksel *background*. Citra hasil *thresholding* disebut juga dengan citra biner karena citra yang hanya memiliki dua macam nilai piksel yaitu 1 dan 0. Konsep binerisasi ini berlandaskan pada makna sistem berpasangan, sesuai dengan firman Allah S.W.T dalam *Al-Qur'an* surat *Az-Zariyat* (51) ayat ke 49.

وَمِنْ كُلِّ شَيْءٍ خَلَقْنَا زَوْجَيْنِ لَعَلَّكُمْ تَذَكَّرُونَ ﴿٤٩﴾

“dan segala sesuatu Kami ciptakan berpasang-pasangan supaya kamu mengingat kebesaran Allah”.(QS. *Az-Zariyat*: 49).

Kandungan dalam ayat ini Allah menciptakan sesuatu secara berpasangan tidak hanya manusia, melainkan segala sesuatu yang tumbuh dari bumi dan berbagai partikel yang tidak terlihat oleh mata.

Fungsi penting keberadaan sistem berpasangan telah diterangkan oleh secara jelas dalam firmanNya dalam *Al-Qur'an* surat *Ar-Ruum* ayat 21.

وَمِنْ آيَاتِهِ أَنْ خَلَقَ لَكُمْ مِنْ أَنْفُسِكُمْ أَزْوَاجًا لِتَسْكُنُوا إِلَيْهَا وَجَعَلَ بَيْنَكُمْ مَوَدَّةً

وَرَحْمَةً إِنَّ فِي ذَلِكَ لَآيَاتٍ لِقَوْمٍ يَتَفَكَّرُونَ ﴿٢١﴾

*“dan di antara tanda-tanda kekuasaan-Nya ialah Dia menciptakan untukmu isteri-isteri dari jenismu sendiri, supaya kamu cenderung dan merasa tenteram kepadanya, dan dijadikan-Nya diantaramu rasa kasih dan sayang. Sesungguhnya pada yang demikian itu benar-benar terdapat tanda-tanda bagi kaum yang berfikir.”(QS. Ar-Ruum: 21).*

Firman Allah S.WT. ini menegaskan bahwa manusia diciptakan berpasangan agar mereka tentram, ketentraman ini tercermin salah satunya melalui keseimbangan. Sistem berpasangan pun memiliki fungsi penting untuk menjaga keseimbangan. Konsep *thresholding* menyeimbangkan antara nilai piksel 0 dan 1, piksel dikelompokkan berdasarkan nilai ambang *thresholding* (T) yang didapatkan.

يَتَأْتِيهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا ۗ إِنَّ  
 أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتَقْوَاهُ ۗ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ ﴿١٣﴾

*“Hai manusia, Sesungguhnya Kami menciptakan kamu dari seorang laki-laki dan seorang perempuan dan menjadikan kamu berbangsa - bangsa dan bersuku-suku supaya kamu saling kenal-mengenal. Sesungguhnya orang yang paling mulia diantara kamu disisi Allah ialah orang yang paling taqwa diantara kamu. Sesungguhnya Allah Maha mengetahui lagi Maha Mengenal.”(QS. Al-Hujuraat:13).*

Kandungan dari *Al-Qur'an* surat *Al-Hujurat* ayat 13 yaitu Allah SWT menjadikan manusia berbangsa-bangsa dan bersuku-suku agar saling kenal-mengenal. Jika dihubungkan dengan konsep *thresholding* bahwa konsep utama metode *thresholding* yaitu mengelompokkan piksel menjadi nilai 1 atau 0 dengan tujuan mempermudah pengenalan suatu objek.

## BAB V

### PENUTUP

#### 5.1 Kesimpulan

Aplikasi ini bertujuan untuk mendeteksi *foramen mentale* pada citra RoI. Kemampuan aplikasi dalam mendeteksi *foramen mentale* diukur dengan cara menganalisis objek *foramen mentale* pada citra hasil uji coba. Hasil uji menunjukkan tigabelas citra RoI dari empat puluh citra RoI yang diuji dinyatakan berhasil terdeteksi adanya objek *foramen mentale*, sehingga aplikasi ini mampu mendeteksi *foramen mentale* dengan prosentasi keberhasilan sebesar 32.5%.

Kinerja aplikasi segmentasi *foramen mentale* dengan perlakuan *thresholding fuzzy type II* memiliki tingkat akurasi dan presisi sebesar 93.39% dan 70.53%, sedangkan segmentasi tanpa perlakuan *thresholding fuzzy type II* memiliki akurasi dan presisi sebesar 92.98% dan 54.71%. Namun segmentasi dengan perlakuan *thresholding fuzzy type II* memiliki nilai *recall* yang lebih rendah (0.57%) dibandingkan dengan segmentasi tanpa perlakuan metode *thresholding* (0.62%). Dengan demikian *thresholding fuzzy type II* mampu meningkatkan nilai akurasi dan presisi dalam mensegmentasi *foramen mentale*, namun proporsi objek pada piksel kebenaran memiliki nilai yang rendah, karena cenderung menghasilkan objek yang lebih kecil dibandingkan dengan proses segmentasi tanpa perlakuan metode *thresholding*.

## 5.2 Saran

Metode ini mampu dikembangkan dengan mengkombinasikan metode *thresholding* dengan metode segmentasi. Metode segmentasi yang tepat juga mempengaruhi hasil segmentasi.




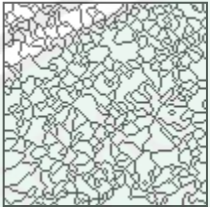

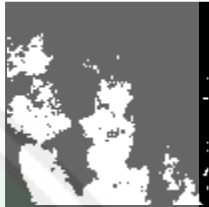
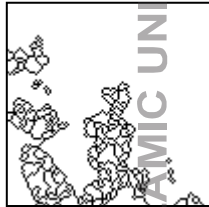
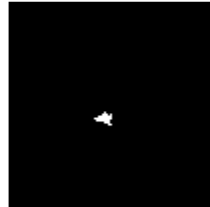

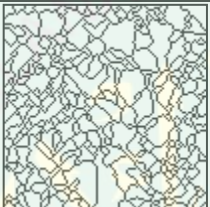


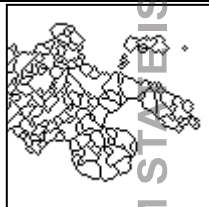


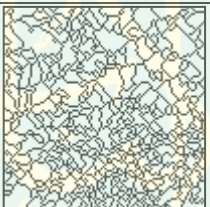



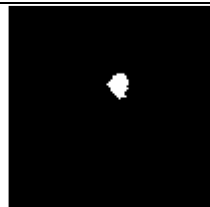
## DAFTAR PUSTAKA



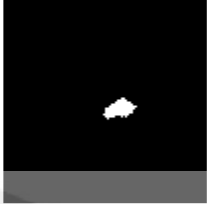


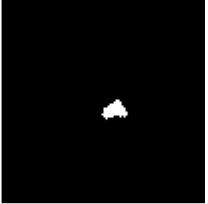
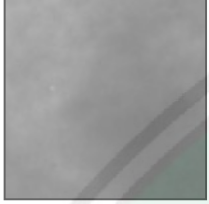
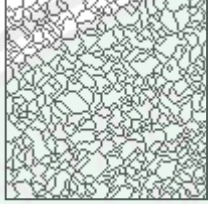


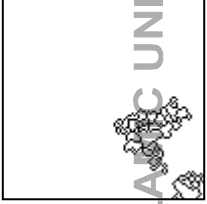

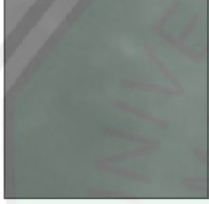
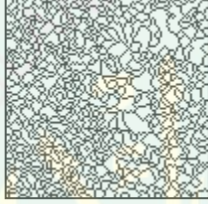

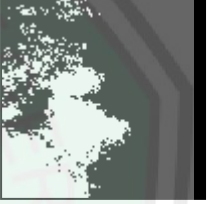

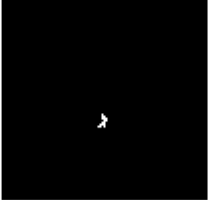

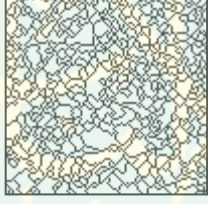


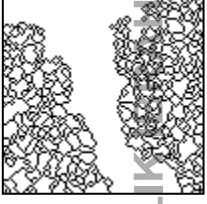
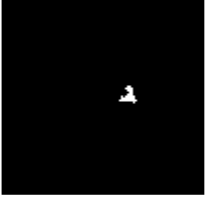
- Abadi, D. 2003. *Mengetahui Posisi Foramen Mental Melalui Radiograf. Skripsi Tidak Diterbitkan*. Medan: Fakultas Kedokteran Gigi Universitas Sumatra Barat.
- Arifin, A. Z. 2005. Computer-aided system for Measuring the Mandibular Cortical Width on Panoramic Radiographs in Osteoporosis Diagnosis. *Medical Imaging* , vol. 5747 h. 813-821.
- Arifin, A. Z. 2010. Image Thresholding using Ultrafuzziness Optimization Based on Type II Fuzzy Sets. *ITB J. ICT* , v. 4, no. 2, h. 79-94.
- Arifin, A. Z. 2007. Use of Fuzzy Neural Network in Diagnosing Postmenopausal Women with Osteoporosis Based on Dental Panoramic Radiograph. *Jurnal of Advanced Computational Intelligence and Intelligent Informatics* , vol. 18 no. 8 p. 1049-1058.
- Bozic, M., & Hren, N. I. 2005. Osteoporosis and Mandibles. *Dentomaxillofacial Radiology* , 178-184.
- Cosman, F. 2009. *OSTEOPOROSIS. Penj. Iriani Syahrir*. Yogyakarta: Penerbit B-First.
- Crysdian, C. dan Achmadi, S. 2008. Pengembangan Sistem Segmentasi Gambar Digital Menggunakan Transformasi Watershed. Seminar Nasional Teknoin Yogyakarta, ISBN: 978-979-3980-15-7.
- Fawcett, T. 2003. ROC graphs: Notes and practical considerations for data mining researchers. *Palo Alto, CA, Tech. Rep. HPL-2003-4. HPLab*.
- Gonzales, R. C., & Woods, R. E. 2001. *Digital Image Processing*. USA: Pearson Education.
- Horner, K. 1996. Mandibular Bone Mineral Density as a Predictor of Skeletal Osteoporosis. *The British Journal of Radiology* , vol. 69 p. 1019-1025.
- Kemenkes. 2008. *Surat Keputusan Menteri Kesehatan Republik Indonesia Nomor 1142/MENKES/SK/XII/2008 Tentang Pedoman Pengendalian Osteoporosis*.
- Klemetti, E. 1993. Panoramic Mandibular Index and Bone Mineral Densities in Postmenopausal Women. *Oral Surg Oral Med Oral Pathol* , v. 75, h. 774-779.


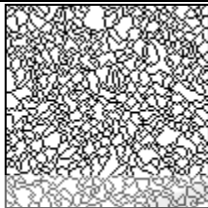
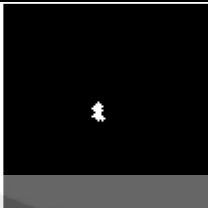

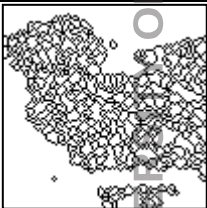
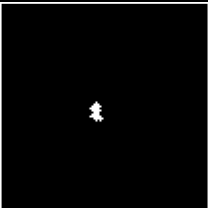
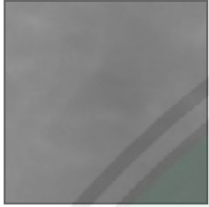
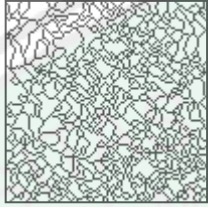



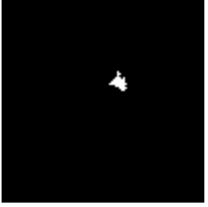

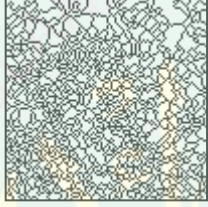

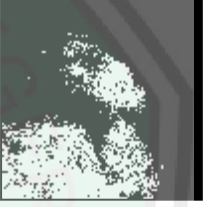
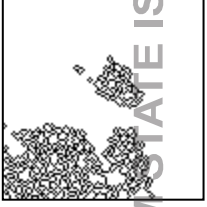

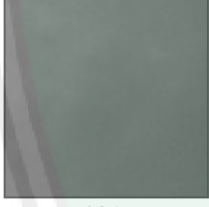
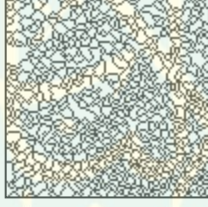


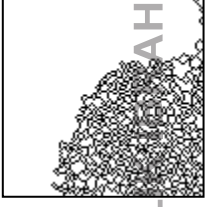
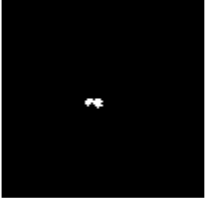
- Marieb, E. N. 2005. *Anatomy and Physiology*. San Fransisco: Person Education.
- Murianto Harjoko, Agus. 2009. Segmentasi Citra Menggunakan Watershed dan Intensitas Filtering sebagai Pre Processing. Seminar Nasional UPN Yogyakarta, ISSN: 1979-2328.
- Roerdink J. B. T. M. dan Meijter, A. 2001. The Watershed Transform: Definition, Algorithms and Parallization Strategies. *Fundamenta Informaticae IOS Press*, v. 41 h. 187-228.
- Soewolo. 2005. *Fisiologi Manusia*. Malang: UM Press.
- Syaifuddin. 2009. *Anatomi Tubuh Manusia, Edisi 2*. Jakarta: Penerbit Salemba Medika.
- Syaifuddin. 2009. *Fisiologi Tubuh Manusia, Edisi 2*. Jakarta: Penerbit Salemba Medika.
- Sutanta, E. 2004. *Algoritma: Teknik Penyelesaian Permasalahan untuk Komputasi*. Yogyakarta: Penerbit Graha Ilmu.
- Taguchi, A. 1996. Usefulness of panoramic radiography in the diagnosis of postmenopausal osteoporosis in womwn and mospology of inferior cortex of the mandible. *Dentomaxillofac Radiol* , v. 25, h. 263-267.
- Taguchi, A. 2005. Use of Dental Panoramic Radiograph in Identifying Younger Postmenopausal Women with Osteoporosis. *International Osteoporosis Foundation* , vol. 17 p. 387-394.
- Tizhoosh, H. R. 2005. Image Thresholding Using Type II Fuzzy Sets. *Pattern Recognition Society* , vol. 38 p. 2363-2372.


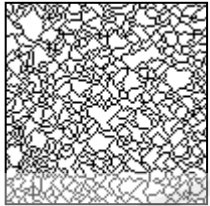


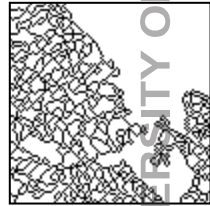

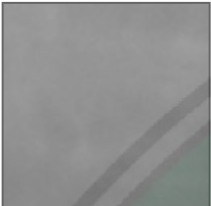
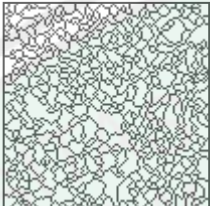





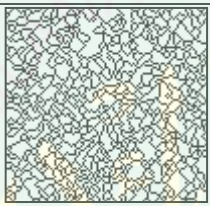


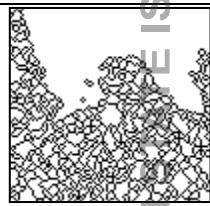


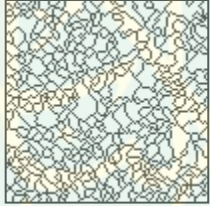


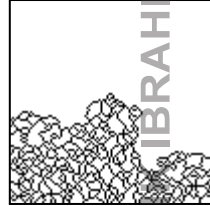
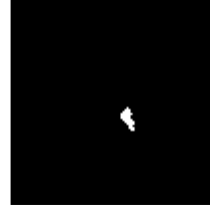
## LAMPIRAN

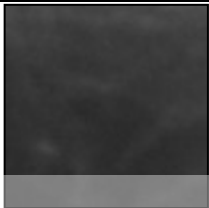
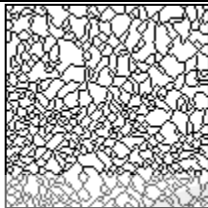
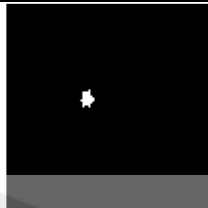


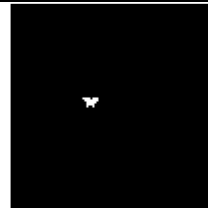
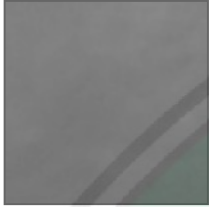
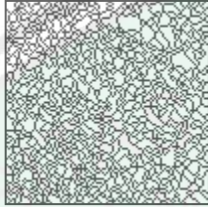


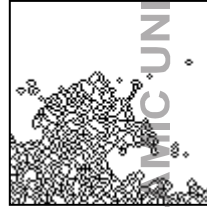


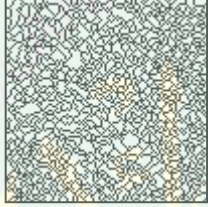





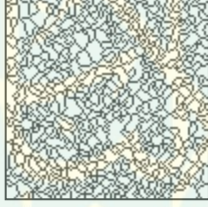

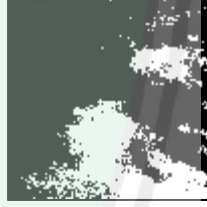
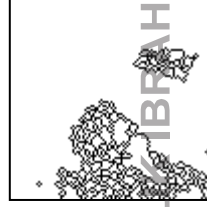

**Lampiran 1: Data citra hasil uji coba identifikasi *foramen mentale* kelompok osteoporosis**


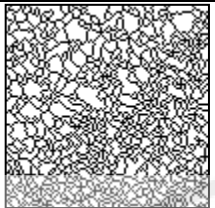
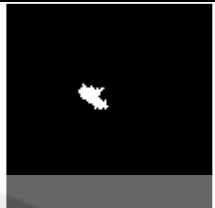

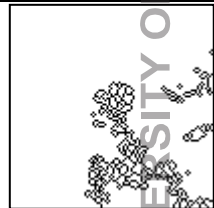
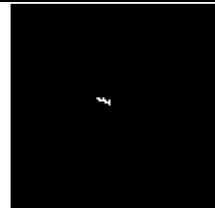

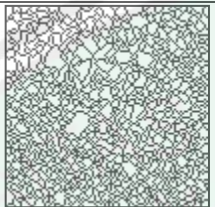

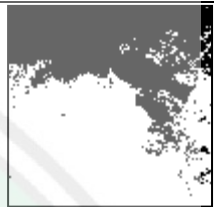
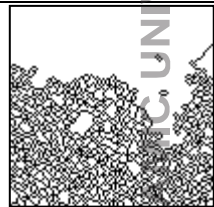
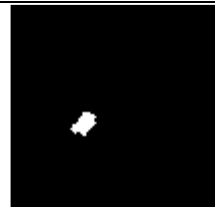

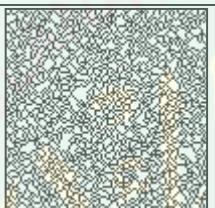

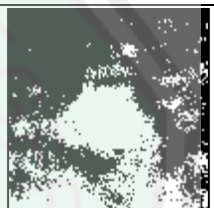
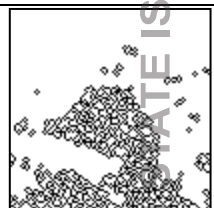
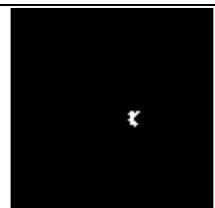

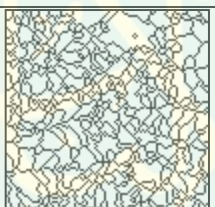




No	RoI <i>Foramen Mentale</i>	Bagian 1		Bagian 2		
		Transformasi <i>Watershed</i>	<i>Region yang Berarti</i>	<i>Thresholding Fuzzy Type II</i>	Trsnformasi <i>Watershed</i>	<i>Region yang Berarti</i>
1	 U002-L					
2	 U002-R					
3	 U004-L					


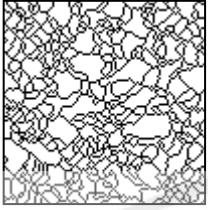




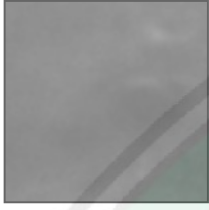
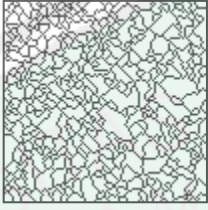


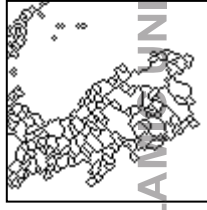




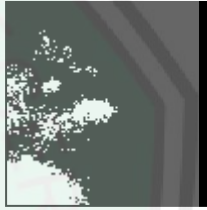



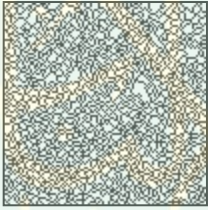

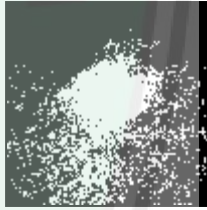
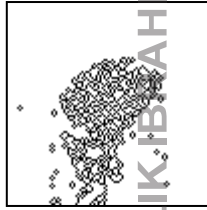
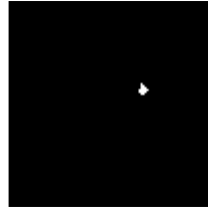
4	 <p>U004-R</p>					
5	 <p>U008-L</p>					
6	 <p>U008-R</p>					
7	 <p>U009-L</p>					

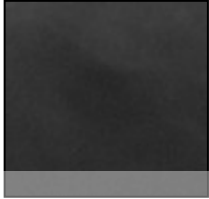
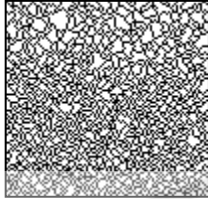
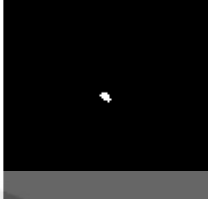


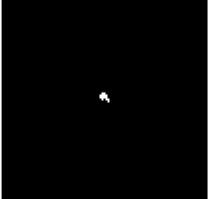

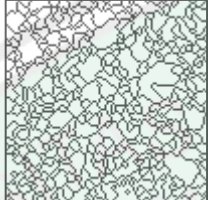

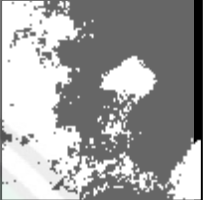

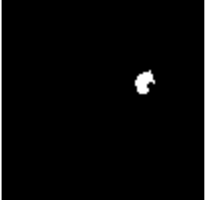

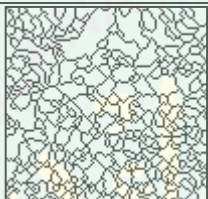

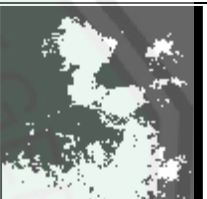



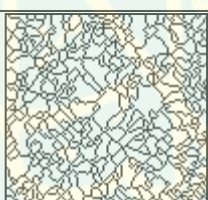

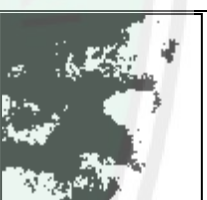

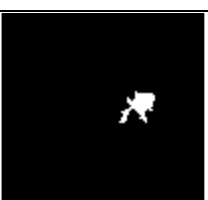
8	 <p data-bbox="353 517 448 544">U009-R</p>					
9	 <p data-bbox="353 761 448 788">U012-L</p>					
10	 <p data-bbox="353 1005 448 1032">U012-R</p>					
11	 <p data-bbox="353 1249 448 1276">U021-L</p>					


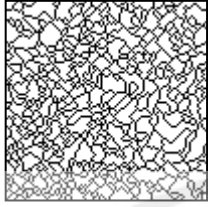
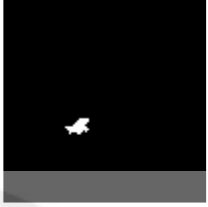
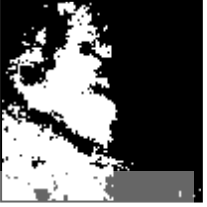


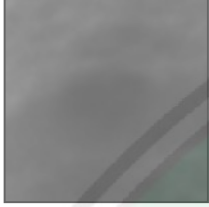



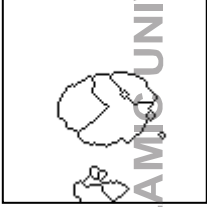








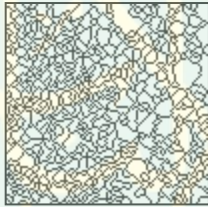

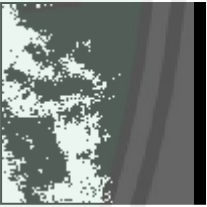
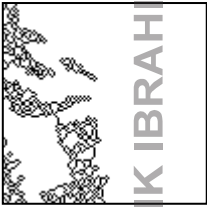

12	 U021-R					
13	 U025-L					
14	 U025-R					
15	 U037-L					


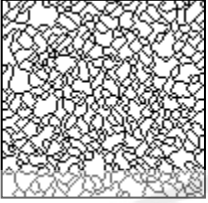
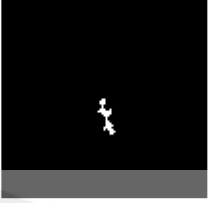
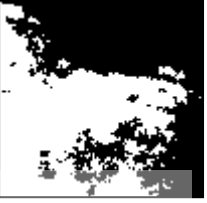
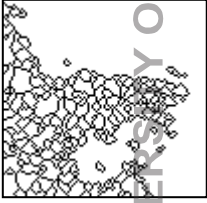
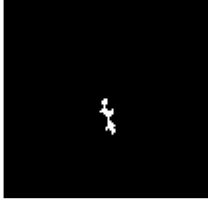

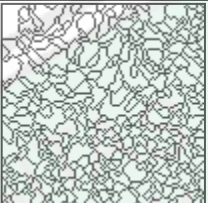

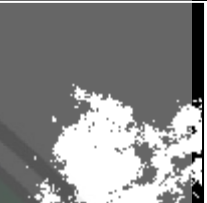
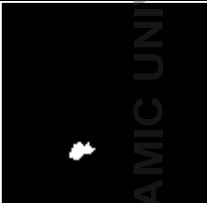
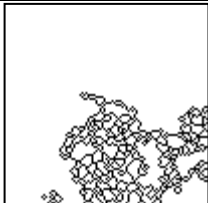

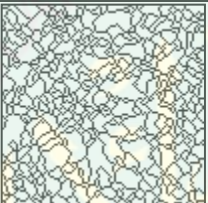

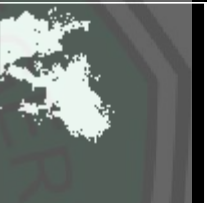

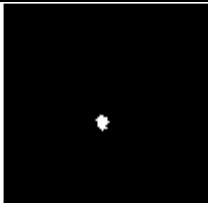

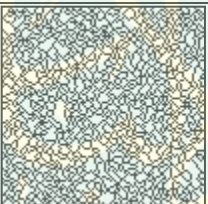


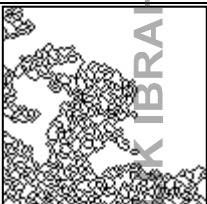
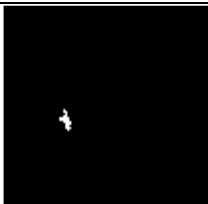
16	 <p>U037-R</p>					
17	 <p>U055-L</p>					
18	 <p>U055-R</p>					
19	 <p>U058-L</p>					

20	 <p data-bbox="353 517 450 539">U058-R</p>					
21	 <p data-bbox="353 756 450 778">U061-L</p>					
22	 <p data-bbox="353 995 450 1018">U061-R</p>					
23	 <p data-bbox="353 1235 450 1257">U072-L</p>					

24	 U072-R					
25	 U073-L					
26	 U073-R					
27	 U075-L					

28	 <p>U075-R</p>					
29	 <p>U077-L</p>					
30	 <p>U077-R</p>					
31	 <p>U079-L</p>					

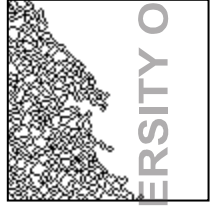
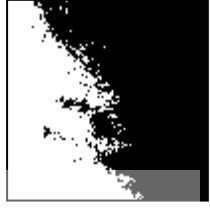
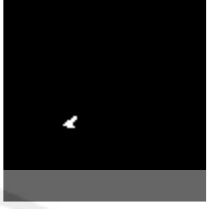
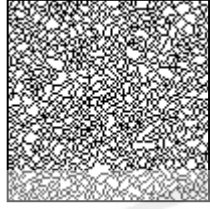
32	 <p>U079-R</p>					
33	 <p>U081-L</p>					
34	 <p>U081-R</p>					
35	 <p>U084-L</p>					

36	 <p>U084-R</p>					
37	 <p>U091-L</p>					
38	 <p>U091-R</p>					
39	 <p>U096-L</p>					








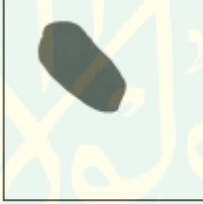
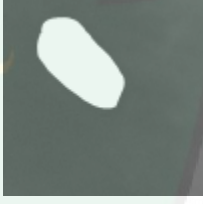



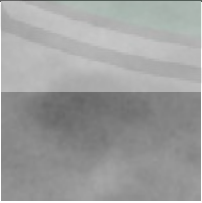



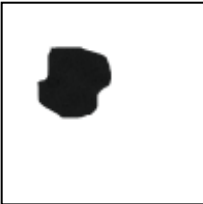

40

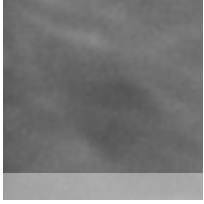


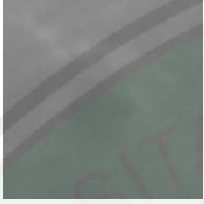

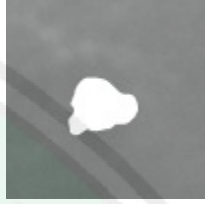


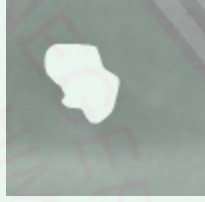

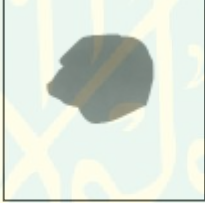
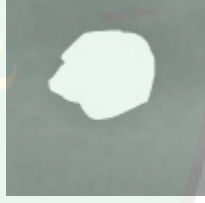


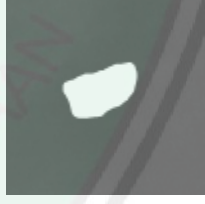

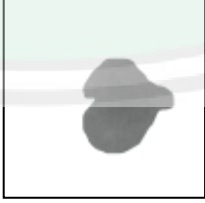
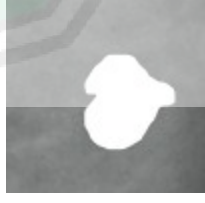


U096-R



Lampiran 2 : Data citra uji untuk penentuan fungsi keanggotaan *fuzzy*

No	Data Citra	Objek <i>Foramen Mentale</i> tanpa Background	Background tanpa Objek <i>Foramen Mentale</i>
1	 002-R.png		
2	 007-R.png		
3	 013-R.png		
4	 032-L.png		
5	 041-R.png		
6	 042-L.png		

7	 050-R.png		
8	 061-L.png		
9	 065-R.png		
10	 080-L.png		
11	 088-L.png		
12	 100-L.png		

Lampiran 3: Data *TP*, *TN*, *FP*, dan *FN* citra uji

No	Nama Citra	<i>T</i>	ROC Bagian 1				ROC Bagian 2			
			<i>TN</i>	<i>TP</i>	<i>FP</i>	<i>FN</i>	<i>TN</i>	<i>TP</i>	<i>FP</i>	<i>FN</i>
1	U002-L	79	170	47	78	9906	170	47	32	9952
2	U002-R	62	408	67	168	9558	408	67	16	9710
3	U004-L	39	321	51	66	9562	342	30	1	9627
4	U004-R	75	315	44	60	9581	315	44	31	9610
5	U008-L	62	121	37	26	9816	121	37	18	9824
6	U008-R	54	233	13	13	9942	233	13	10	9945
7	U009-L	116	487	33	26	9454	487	33	11	9469
8	U009-R	75	864	39	0	9298	864	39	0	9298
9	U012-L	39	503	22	26	9449	484	41	0	9475
10	U012-R	33	673	9	25	9293	658	24	1	9317
11	U021-L	56	341	47	27	9685	350	38	0	9712
12	U021-R	79	258	15	41	9686	262	11	30	9697
13	U025-L	79	210	33	31	10130	210	33	31	10130
14	U025-R	54	271	39	8	9282	271	39	3	9287
15	U037-L	92	270	31	25	9875	274	27	18	9882
16	U037-R	58	137	17	50	9997	134	20	19	10028
17	U055-L	39	327	28	0	9846	327	28	0	9846
18	U055-R	33	322	10	22	9746	322	10	15	9753
19	U058-L	36	425	76	25	9675	425	76	25	9675
20	U058-R	31	436	100	1	9664	467	69	0	9665
21	U061-L	58	582	55	34	9429	582	55	34	9429
22	U061-R	39	700	16	37	9247	701	15	11	9273
23	U072-L	54	353	12	60	9675	342	23	34	9701
24	U072-R	67	586	38	169	9307	586	38	100	9376
25	U073-L	75	304	54	60	9682	304	54	60	9682
26	U073-R	39	249	84	34	9834	249	84	22	9846
27	U075-L	46	831	25	26	9319	831	25	26	9319
28	U075-R	46	730	17	0	9050	730	17	0	9050
29	U077-L	58	296	161	47	9496	360	97	1	9542
30	U077-R	60	73	32	36	9859	73	32	19	9876
31	U079-L	62	223	130	68	9679	225	128	50	9697
32	U079-R	54	574	46	14	9266	590	30	0	9280
33	U081-L	39	437	241	1127	8396	437	241	222	9301
34	U081-R	60	281	462	263	9195	328	415	58	9400
35	U084-L	33	285	68	7	9740	285	68	7	9740

36	U084-R	84	572	31	22	9273	572	31	20	9275
37	U091-L	36	312	75	0	9713	318	69	0	9713
38	U091-R	51	595	20	16	9469	595	20	16	9469
39	U096-L	79	400	27	39	9634	402	25	3	9670
40	U096-R	75	197	20	12	9771	203	14	1	9782

