



**TUGAS AKHIR - EB184803**

**KLASIFIKASI TUMOR OTAK PADA CITRA *MAGNETIC RESONANCE IMAGE* (MRI) DENGAN MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE* (SVM)**

**Ainani Shabrina Febrianti**  
**0731154000020**

**DOSEN PEMBIMBING:**  
**Dr. Tri Arief Sardjono, S.T., M.T.**  
**Atar Fuady Babgei, S.T., M.Sc.**

**PROGRAM SARJANA**  
**DEPARTEMEN TEKNIK BIOMEDIK**  
**FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS**  
**INSTITUT TEKNOLOGI SEPULUH NOPEMBER**  
**SURABAYA**  
**2020**



**FINAL PROJECT - EB184803**

**CLASSIFICATION OF BRAIN TUMORS IN MAGNETIC RESONANCE  
IMAGE (MRI) USING SUPPORT VECTOR MACHINE (SVM) METHOD**

**Ainani Shabrina Febrianti  
0731154000020**

**SUPERVISOR:  
Dr. Tri Arief Sardjono, S.T., M.T.  
Atar Fuady Babgei, S.T., M.Sc.**

**UNDERGRADUATE PROGRAM  
BIOMEDICAL ENGINEERING DEPARTMENT  
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS  
TECHNOLOGY  
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY  
SURABAYA  
2020**

## PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa sebagian maupun keseluruhan isi tugas akhir saya yang berjudul “**Klasifikasi Tumor Otak pada Citra *Magnetic Resonance Image* (MRI) dengan Menggunakan Metode *Support Vector Machine* (SVM)**” merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya mandiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Januari 2020



Ainani Shabrina Febrianti

NRP. 0731154000020

**Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar  
Sarjana Teknik (ST)**

**di**

**Institut Teknologi Sepuluh Nopember**

**Oleh**

**Ainani Shabrina Febrianti**

**NRP. 0731154000020**

**Tanggal Ujian : 3 Januari 2020**


**Periode Wisuda : Maret 2020**

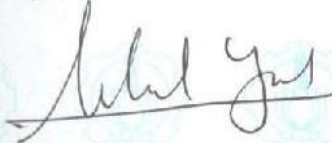
**Disetujui oleh :**

**Dosen Penguji :**

**Dosen Pembimbing :**

  
**1. Dr. Achmad Arifin, S.T., M.Eng.**  
**NIP. 19710314 199702 1001**

  
**1. Dr. Tri Arief Sardjono, S.T., M.T.**  
**NIP. 19700212 199512 1001**

  
**2. Muhammad Yazid, B.Eng., M.Eng.**  
**NIP. 19800420 201504 1001**

  
**2. Atar Fuady Babgei, S.T., M.Sc.**  
**NIP. 19891111 201812 1001**

  
**3. Muhammad Hilman Fatoni, S.T., M.T.**  
**NIP. 19910325 201504 1001**



**Kepala Departemen Teknik Biomedik,**

**Dr. Achmad Arifin, S.T., M.Eng.**

**NIP. 19710314 199702 1001**

# **KLASIFIKASI TUMOR OTAK PADA CITRA *MAGNETIC RESONANCE IMAGE* (MRI) DENGAN MENGGUNAKAN METODE *SUPPORT VECTOR MACHINE* (SVM)**

Nama mahasiswa : Ainani Shabrina Febrianti  
NRP : 07311540000020  
Pembimbing : 1. Dr. Tri Arief Sardjono, S.T., M.T.  
: 2. Atar Fuady Babgei, S.T., M.Sc.

## **Abstrak**

Di Indonesia, pasien yang terdiagnosis tumor otak semakin meningkat setiap tahunnya. Bukan hanya orang dewasa saja yang terserang penyakit tumor otak, tetapi anak-anak dengan usia yang masih muda juga bisa terserang penyakit tumor otak. Karena itu, perlu dilakukan deteksi gejala tumor sejak dini sebelum tumor otak berkembang menjadi tumor yang ganas. Tumor otak dapat dideteksi dengan cara melalui pemeriksaan *Magnetic Resonance Imaging* (MRI). Tumor otak merupakan penyakit berbahaya yang menyerang pada sistem syaraf pusat. Tumor otak dibedakan menjadi dua kelompok, yaitu tumor jinak dan tumor ganas. Untuk membedakan penyakit yang terdeteksi tumor otak dengan yang tidak terdeteksi tumor otak, perlu dilakukan proses klasifikasi. Proses klasifikasi ini menggunakan metode *Support Vector Machine* (SVM) untuk mengklasifikasikan tumor otak menjadi dua kelas yaitu ada tumor atau non tumor. Dalam tugas akhir ini, proses ini dilakukan dalam tiga langkah, *Pre-processing* yang mengubah ukuran citra, ekstraksi fitur yang mengambil informasi secara teksturnya, kemudian diikuti dengan melatih data dan dilakukan pengujian pada SVM. Hasil dari tugas akhir ini adalah untuk mengklasifikasikan citra dengan tumor atau non tumor. Serta menganalisis tingkat akurasi klasifikasi untuk kernel SVM yang berbeda seperti Linear, RBF dan Polynomial. Dari hasil pengujian klasifikasi dengan SVM, didapatkan tingkat akurasi yang cukup baik sebesar 0.76 dan memiliki *error* sebesar 0.24 dengan menggunakan kernel Linear dan RBF pada tipe C-SVM. Kesimpulan dari penelitian ini adalah metode SVM dapat digunakan untuk mengklasifikasikan tumor otak dengan tingkat akurasi yang cukup baik dengan menggunakan kernel Linear dan RBF. Dengan menggunakan SVM *classifier* untuk mengklasifikasi data yang bersifat non-linear dengan menggunakan fungsi kernel seperti RBF dapat menghasilkan hasil yang bagus ketika dataset yang digunakan lebih kecil.

Kata kunci : Ekstraksi Fitur, *Magnetic Resonance Imaging*, *Support Vector Machine*, *Pre-processing*, Tumor otak.

# **CLASSIFICATION OF BRAIN TUMORS IN MAGNETIC RESONANCE IMAGE (MRI) USING SUPPORT VECTOR MACHINE (SVM) METHOD**

By : Ainani Shabrina Febrianti  
Student Identity Number : 0731154000020  
Supervisor : 1. Dr. Tri Arief Sardjono, S.T., M.T.  
: 2. Atar Fuady Babgei, S.T., M.Sc.

## **Abstract**

In Indonesia, patients diagnosed with brain tumors are increasing every year. Not only adults who have a brain tumor, but children with a young age can also be stricken with a brain tumor. Therefore, it is necessary to detect tumor symptoms early before the brain tumor develops into a malignant tumor. Brain tumors can be detected by means of magnetic resonance imaging (MRI) examinations. Brain tumors are dangerous diseases that attack the central nervous system. Brain tumors are divided into two groups, namely benign and malignant tumors. To differentiate diseases detected by brain tumors from those not detected by brain tumors, it is necessary to do the classification process. This classification process uses the Support Vector Machine (SVM) method to classify brain tumors into two classes: tumor or non-tumor. In this final project, this process is carried out in three steps, Pre-processing which resizes the image, extracts features that take texture information, then is followed by training the data and testing on SVM. The result of this final project is to classify images with tumors or non-tumors. As well as analyzing the level of classification accuracy for different SVM kernels such as Linear, RBF and Polynomial. From the classification test results with SVM, obtained a fairly good level of accuracy of 0.76 and has an error of 0.24 by using the Linear kernel and RBF on type C-SVM. The conclusion from this study is the SVM method can be used to classify brain tumors with a fairly good degree of accuracy using the Linear kernel and RBF. Using the SVM classifier to classify data that is non-linear by using kernel functions such as RBF can produce good results when the dataset used is smaller.

Keywords : Brain tumor, Feature extraction, Magnetic Resonance Imaging, Pre-processing, Support Vector Machine.

## KATA PENGANTAR

Segala puji bagi Allah SWT atas segala rahmat-Nya, karena-Nya Tugas Akhir ini dapat berjalan lancar dan selesai tepat pada waktunya. Maksud dan tujuan dari penulisan Tugas Akhir ini adalah untuk memenuhi persyaratan kelulusan program Studi Strata I pada Jurusan Teknik Biomedik di Institut Teknologi Sepuluh Nopember. Menyadari penyusunan laporan ini tidak lepas dari bantuan berbagai pihak, maka pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua yang telah mendukung, memberi nasehat, saran, serta memberikan fasilitas untuk membantu penyelesaian Tugas Akhir ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. dan Atar Fuady Babgei, S.T., M.Sc., selaku dosen pembimbing yang senantiasa membimbing, mengarahkan, serta menginspirasi penulis selama pengerjaan penelitian ini.
3. Dr. Achmad Arifin, S.T., M.Eng. selaku kepala departemen yang telah memberi inspirasi, motivasi, serta bimbingan selama pengerjaan tugas akhir terlebih selama perkuliahan di ITS.
4. Ibu Nada Fitriyatul Hikmah, S.T., M.T. selaku dosen wali.
5. Seluruh staf pengajar Teknik Biomedik ITS yang telah membimbing dan memberikan materi perkuliahan kepada penulis.
6. Rekan-rekan Teknik Biomedik angkatan 2015 dan juga pihak-pihak lain yang turut membantu dalam penyelesaian Tugas Akhir ini.

Penulis merasa bahwa dalam menyusun laporan ini masih menemui beberapa kesulitan dan hambatan, disamping itu juga menyadari bahwa penulisan laporan ini masih jauh dari sempurna dan masih banyak kekurangan-kekurangan lainnya, maka dari itu penulis mengharapkan saran dan kritik yang membangun dari semua pihak.

Akhir kata semoga Tugas Akhir ini akan memberikan manfaat bagi penulis maupun bagi pihak-pihak yang sedang membutuhkan.

Surabaya, Desember 2019

Ainani Shabrina Febrianti

NRP. 07311540000020

## DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR.....	ii
LEMBAR PENGESAHAN .....	ii
Abstrak .....	vi
Abstract .....	vii
KATA PENGANTAR .....	viii
DAFTAR ISI.....	ix
DAFTAR GAMBAR .....	xii
DAFTAR TABEL.....	xiii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah.....	2
1.4 Tujuan dan Manfaat Penelitian.....	2
1.5 Kontribusi.....	3
BAB 2 TINJAUAN PUSTAKA .....	4
2.1 Tumor Otak .....	4
2.2 Pengolahan Citra .....	5
2.2.1 <i>Grayscale</i> .....	5
2.2.2 <i>Gaussian Filter</i> .....	6
2.2.3 <i>Thresholding</i> .....	6
2.2.4 Dilasi .....	7
2.2.5 Erosi .....	7
2.2.5 <i>Contour</i> .....	7
2.3 <i>Feature Extraction</i> .....	8
2.3.1 <i>Gray Level Co-occurrence Matrix (GLCM)</i> .....	8
2.4 Open CV.....	10



2.5	<i>Support Vector Machine (SVM)</i> .....	11
2.6	<i>Confusion Matrix</i> .....	14
2.7	Rasional .....	14
BAB 3 METODOLOGI.....		17
3.1	Desain Metode Penelitian.....	17
3.2	Dataset .....	18
3.3	<i>Image Pre-processing</i> .....	18
3.4	<i>Feature Extraction</i> .....	19
3.5	<i>Support Vector Machine (SVM)</i> .....	22
3.6	Metode Pengujian.....	23
BAB 4 PENGUJIAN SISTEM .....		25
4.1	<i>Pre-Processing</i> .....	25
4.1.1	<i>Gaussian Blur</i> .....	25
4.1.2	<i>Threshold</i> .....	26
4.1.3	<i>Contour</i> .....	26
4.1.4	<i>Automatic Cropping</i> .....	26
4.2	<i>Feature Extraction</i> .....	28
4.3	Klasifikasi.....	37
BAB 5 HASIL DAN PEMBAHASAN.....		41
5.1	<i>Pre-processing</i> .....	41
5.1.1	<i>Gaussian Blur</i> .....	42
5.1.2	<i>Thresholding</i> .....	42
5.1.3	<i>Contour</i> .....	42
5.1.4	<i>Automatic Cropping</i> .....	42
5.2	<i>Feature Extraction</i> .....	43
5.3	Klasifikasi.....	44
BAB 6 KESIMPULAN DAN SARAN .....		47
6.1	Kesimpulan.....	47
6.2	Saran .....	47

DAFTAR PUSTAKA .....	49
BIODATA PENULIS .....	50

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> Tumor Otak [1] .....	4
<b>Gambar 2. 2</b> Derajat Keabuan [3].....	5
<b>Gambar 2. 3</b> (a) contoh citra dengan 4 tingkat keabuan, (b) GLCM pada jarak 1 dan arah 0° .....	10
<b>Gambar 2. 4</b> SVM mencari Hyperplane yang memisahkan kedua class [8]. .....	11
<b>Gambar 2. 5</b> Dataset Nonlinear [8].....	13
<b>Gambar 2. 6</b> Fishbone Diagram.....	16
<b>Gambar 3. 1</b> Diagram Blok Metode Penelitian.....	17
<b>Gambar 3. 2</b> Diagram Blok proses Image Pre-processing.....	19
<b>Gambar 3. 3</b> (a) contoh citra dengan 4 tingkat keabuan, (b) Matrix Co-occurrence, (c) GLCM pada jarak 1 dan arah 0° .....	20
<b>Gambar 3. 4</b> Diagram Blok proses Feature Extraction.....	21
<b>Gambar 3. 5</b> Diagram blok metode Klasifikasi .....	22
<b>Gambar 3. 6</b> Flowchart learning SVM .....	24
<b>Gambar 4. 1</b> Hasil dari proses Gaussian Blur (b).....	25
<b>Gambar 4. 2</b> Hasil dari proses Thresholding (c).....	26
<b>Gambar 4. 3</b> Hasil dari proses Automatic Cropping (f) .....	27
<b>Gambar 4. 4</b> Hasil dari proses Contour (d).....	27
<b>Gambar 4. 5</b> Hasil dari proses ekstraksi fitur tumor otak.....	32
<b>Gambar 4. 6</b> Hasil dari ekstraksi fitur antara fitur Contrast dan fitur Correlation ....	33
<b>Gambar 4. 7</b> Hasil dari ekstraksi fitur antara fitur Energy dan fitur Dissimilarity ....	34
<b>Gambar 4. 8</b> Hasil dari ekstraksi fitur antara fitur ASM dan fitur Homogeneity .....	35
<b>Gambar 4. 9</b> Hasil dari ekstraksi fitur antara fitur Entropy dan fitur Dissimilarity ..	36
<b>Gambar 4. 10</b> Hasil klasifikasi SVM untuk mencari hyperplane antara kelas tumor dan non tumor dengan menggunakan C-SVM, (a) Gambar sebelum diberikan hyperplane, (b) Gambar setelah diberi hyperplane dengan menggunakan kernel linear, (c) Gambar setelah diberi hyperplane dengan menggunakan kernel RBF, (d) Gambar setelah diberikan hyperplane dengan menggunakan kernel polynomial.....	38
<b>Gambar 4. 11</b> Hasil klasifikasi SVM untuk mencari hyperplane antara kelas tumor dan non tumor dengan menggunakan Nu-SVM, (a) Gambar sebelum diberikan hyperplane, (b) Gambar setelah diberi hyperplane dengan menggunakan kernel linear, (c) Gambar setelah diberi hyperplane dengan menggunakan kernel RBF, (d) Gambar setelah diberikan hyperplane dengan menggunakan kernel polynomial.....	39

## DAFTAR TABEL

<b>Tabel 2. 1</b> Hasil dari Confusion Matrix.....	14
<b>Tabel 4. 1</b> Hasil dari proses ekstraksi fitur dengan GLCM.....	28
<b>Tabel 4. 2</b> Hasil Confusion Matrix .....	40
<b>Tabel 4. 3</b> Hasil dari proses klasifikasi dengan SVM .....	40
<b>Tabel 5. 1</b> Perbandingan hasil akurasi dari SVM dengan menggunakan pre-processing dan tidak menggunakan pre-processing .....	41

# BAB 1 PENDAHULUAN

## 1.1 Latar Belakang

Kasus tumor otak di dunia semakin meningkat setiap tahunnya. Di Indonesia, terhitung ada 300 pasien setiap tahunnya yang terdiagnosis tumor otak. Bukan hanya orang dewasa, tetapi tumor otak juga menyerang anak-anak dengan usia yang tergolong muda. Tidak sedikit masyarakat yang cenderung menganggap remeh gejala tumor otak yang ditimbulkan. Oleh karena itu, deteksi sejak dini menjadi sangat penting dan diperlukan sebelum tumor otak berkembang ke stadium yang lebih parah. Baik dari tenaga medis maupun masyarakat harus waspada terhadap berbagai gejala dan kelainan yang terjadi secara terus-menerus.

Tumor otak merupakan sebuah lesi yang terletak pada intracranial yang menempati ruang di dalam tengkorak. Tumor-tumor selalu bertumbuh sebagai sebuah massa yang berbentuk bola tetapi juga dapat tumbuh menyebar, masuk ke dalam jaringan. Penyakit berbahaya ini menyerang pada system saraf, di samping tumor spinal dan tumor saraf perifer. Tumor otak dibedakan menjadi dua kelompok berdasarkan perkembangannya, yaitu tumor jinak (tidak bersifat kanker) dan tumor ganas (bersifat kanker). Tumor yang tumbuh di otak dikenal dengan istilah tumor otak primer, sedangkan tumor yang tumbuh di bagian lain dari tubuh dan menyebar hingga ke otak disebut dengan tumor otak sekunder atau metastatik. Meskipun jinak, tumor otak tetap berbahaya sama seperti tumor yang ganas tergantung pada lokasi tumor, di mana tumor yang terletak pada bagian otak yang penting akan menimbulkan gejala yang serius seperti kematian atau kecacatan pada otak.

Tumor pada otak dapat dideteksi dengan cara melalui pemeriksaan *Magnetic Resonance Imaging* (MRI) yang menggunakan alat medis. Setelah pasien menjalani proses pemeriksaan, dokter spesialis radiologi akan menganalisis dan mengambil kesimpulan dari citra yang dihasilkan oleh alat medis tersebut. Salah satu kelebihan dari menggunakan MRI ini adalah dapat mengamati diferensiasi pada jaringan lunak seperti jaringan “white matter” dan “gray matter” yang dapat dibedakan oleh citra MRI secara jelas. Dengan mengolah citra yang dihasilkan oleh alat MRI dapat dikembangkan metode pendeteksian tumor otak yang mampu mendeteksi dan mengklasifikasikan tumor tersebut.

Pada tugas akhir ini, digunakan tiga tahapan yang akan membantu proses mengklasifikasi tumor otak. Tahapan pertama yaitu pre-processing, kemudian dilanjutkan dengan ekstraksi ciri sebagai proses untuk mengambil informasi pada citra otak yang dilakukan dengan metode pendekatan statistik. Hasil ekstraksi ciri akan dikenali dan diklasifikasikan dengan metode SVM (*Support Vector Machine*). Dengan

menggunakan metode dari ekstraksi ciri ini diharapkan dapat menjadi alat bantu untuk mengklasifikasikan tumor otak ke dalam dua kelompok yaitu terdapat tumor atau tidak terdapat tumor. Dengan menerapkan ekstraksi ciri yang dapat mengambil informasi dari citra serta dengan metode klasifikasi SVM yang dapat mencari *Hyperplane* pemisah yang paling baik antar kelas maka system deteksi tumor otak dapat diimplementasikan dengan menggunakan gabungan dua metode tersebut untuk menghasilkan pengklasifikasi yang lebih baik.

## **1.2 Rumusan Masalah**

Berdasarkan latar belakang yang sudah dijelaskan di atas, untuk mendeteksi keabnormalan pada gambar MRI secara otomatis sangat penting dalam aplikasi diagnosis karena data kuantitas dalam gambar MRI yang tinggi, serta melakukan segmentasi dan klasifikasi tumor yang membutuhkan waktu yang tidak sedikit dan terbilang sangat sulit. Maka diperlukan metode deteksi dan klasifikasi tumor otak secara otomatis untuk meningkatkan akurasi dan hasil serta mengurangi waktu yang diperlukan untuk klasifikasi tumor otak.

## **1.3 Batasan Masalah**

Batasan citra yang digunakan sebagai masukan merupakan citra otak hasil MRI dengan format .JPG. Dataset citra yang digunakan yaitu citra medis berwarna RGB. Kemudian dataset tersebut akan dijadikan *Grayscale*. Dataset citra yang digunakan berjumlah 100 gambar. Untuk proses pembelajaran, dataset tersebut akan dibagi menjadi data *training* dan data *test*. Perbandingan *training* dataset dengan *test* dataset adalah 75:25. Keluaran yang dihasilkan berupa klasifikasi tumor otak dengan tingkat keakuratan yang cukup baik dengan menggunakan metode SVM.

## **1.4 Tujuan dan Manfaat Penelitian**

Tujuan dari penelitian ini adalah dihasilkannya kesimpulan pengujian terkait dengan klasifikasi data MRI tumor otak menggunakan metode *Support Vector Machine* (SVM) untuk dapat membedakan lesi pada otak yang diduga tumor otak dengan bagian yang bukan tumor.

Manfaat dari penelitian ini adalah Membantu dokter untuk mengklasifikasikan jenis tumor pada otak dan dapat dengan mudah ditemukan serta dengan belajar memahami setiap proses dari *image processing* akan menambah pengetahuan di bidang *Image Processing* dan *machine learning*.

## **1.5 Kontribusi**

Kontribusi yang diharapkan dari penelitian ini meliputi kontribusi ilmiah dan kontribusi praktis. Kontribusi ilmiah yang diharapkan dari penelitian ini adalah dapat menambah wawasan mengenai *Machine Learning*, pencitraan pada MRI, peningkatan kualitas citra MRI, serta metode klasifikasi menggunakan SVM berupa publikasi karya ilmiah dari pengembangan metode yang digunakan pada penelitian ini sehingga kedepannya dapat digunakan sebagai referensi untuk mengklasifikasi citra MRI.

Kontribusi praktis yang diharapkan dalam penelitian ini adalah dapat memfasilitasi dokter dalam mengklasifikasi penyakit tumor otak dengan menggunakan metode SVM serta dapat membantu masyarakat umum untuk diberikan penanganan awal pada penyakit tumor otak dikarenakan dapat mendeteksi tumor otak secara akurat.

## BAB 2 TINJAUAN PUSTAKA

### 2.1 Tumor Otak

Menurut Smeltzer[1], tumor otak adalah sebuah lesi yang terletak pada intrakranial yang menempati ruang di dalam tengkorak yang selalu tumbuh sebagai massa yang berbentuk bola dan dapat menyebar masuk ke dalam jaringan akibat kompresi dan infiltrasi jaringan dengan perubahan fisik yang bervariasi. Tumor otak adalah neoplasma atau proses desak ruang (*space occupying lesion* atau *space taking lesion*) yang timbul di dalam rongga tengkorak baik di dalam kompartemen supratentorial maupun infratentorial mencakup tumor-tumor primer pada korteks, meninges, vaskuler, kelenjar hipofise, epifise, saraf otak, jaringan penyangga, serta tumor metastasis dari bagian tubuh lainnya.

Tumor otak dapat dibedakan menjadi dua, yaitu tumor jinak dan tumor ganas. Tumor jinak berasal dari sel di dalam atau di sekitar otak, tidak mengandung sel kanker, berkembangnya lambat dan memiliki batas yang jelas serta tidak menyebar ke jaringan lain. Sedangkan tumor ganas mengandung sel kanker dan seringnya tidak memiliki batas yang jelas. Tumor juga dibedakan berdasarkan asal pertumbuhannya, yakni tumor primer dan tumor sekunder. Tumor primer adalah sel kanker yang pertama kali tumbuh di dalam otak dan kemungkinan dapat menyebar ke bagian otak dan tulang belakang, namun jarang menyebar ke bagian tubuh lainnya. Sedangkan tumor sekunder, yang dikenal dengan metastik, merupakan sel kanker yang berasal dari bagian tubuh lain lalu menyebar hingga ke otak. Tumor sekunder lebih banyak



**Gambar 2. 1** Tumor Otak [1]



menyerang manusia dibandingkan tumor primer (National Brain Tumor Society, 2013). Klasifikasi tumor otak menurut Smeltzer[1], yaitu :

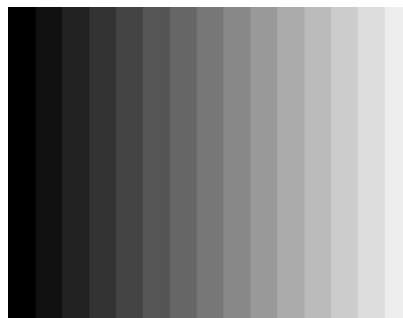
- Tumor yang muncul dari pembungkus otak seperti meningioma dura
- Tumor yang berkembang di dalam atau di atas saraf kranial seperti neuroma akustik
- Tumor yang berasal di dalam jaringan otak seperti glioma
- Lesi metastatik yang berasal dari bagian tubuh lainnya

## 2.2 Pengolahan Citra

*Image Processing* atau sering disebut dengan Pengolahan Citra digital merupakan suatu bentuk pemrosesan atau pengolahan sinyal dengan masukan berupa gambar (image) dan ditransformasikan menjadi gambar lain sebagai keluarannya dengan menggunakan teknik-teknik tertentu. Pengolahan citra digital adalah suatu disiplin ilmu yang mempelajari hal-hal yang berkaitan dengan perbaikan kualitas gambar (peningkatan kontras, transformasi warna, restorasi citra), transformasi gambar (rotasi, translasi, skala, transformasi geometrik), melakukan pemilihan citra ciri (feature image) yang optimal untuk tujuan analisis, melakukan proses penarikan informasi atau deskripsi objek atau pengenalan objek yang terkandung pada citra, melakukan kompresi atau reduksi data untuk tujuan penyimpanan data, transmisi data, dan waktu proses data. Input dari pengolahan citra adalah citra, sedangkan Output-nya adalah citra hasil pengolahan[2].

### 2.2.1 Grayscale

Proses awal yang banyak dilakukan dengan Pengolahan Citra adalah mengubah citra berwarna menjadi citra *grayscale*, hal ini digunakan untuk menyederhanakan model citra. Pada awalnya citra terdiri dari 3 layer matrix yaitu R-layer, G-layer dan B-layer. 3 layer tersebut dapat diubah menjadi 1 layer matrix *grayscale* dan hasilnya adalah citra *grayscale*. Dalam citra ini tidak ada lagi warna, yang ada adalah derajat keabuan[3].



**Gambar 2. 2** Derajat Keabuan [3].

### 2.2.2 Gaussian Filter

*Gaussian Filter* merupakan salah satu metode *linear spatial filtering* dan telah banyak digunakan dalam bidang analisis citra untuk proses *smoothing* (penghalusan), *blurring* (pengaburan), menghilangkan detail dan menghilangkan derau pada citra. *Gaussian Smoothing* atau biasa juga disebut *Gaussian Blur* adalah hasil dari pengaburan sebuah citra dengan menggunakan fungsi Gaussian. Secara matematis, mengaplikasikan sebuah *Gaussian Smoothing* ke sebuah citra sama saja dengan menyelimuti citra dengan sebuah fungsi Gaussian[4]. Aplikasi tersebut akan mengurangi komponen citra yang mempunyai frekuensi tinggi karena *Gaussian Smoothing* merupakan *Low Pass Filter*. Persamaan dari fungsi Gaussian dalam satu dimensi yaitu :

$$G(x) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{\left(\frac{-x^2}{2\sigma^2}\right)} \quad (2.1)$$

Sedangkan pada dua dimensi yaitu :

$$G(x) = \frac{1}{2\pi\sigma^2} e^{\left(\frac{-(x^2+y^2)}{2\sigma^2}\right)} \quad (2.2)$$

X adalah jarak dari asal sumbu horizontal, y adalah jarak dari asal sumbu vertikal, dan  $\sigma$  adalah deviasi standar dari distribusi Gaussian. Ketika diaplikasikan dalam dua dimensi, persamaan ini menghasilkan kontur permukaan berupa lingkaran konsentris dengan distribusi Gaussian dari titik pusat. Nilai dari distribusi ini digunakan untuk membangun sebuah matriks konvolusi yang diterapkan pada citra asli. Nilai baru setiap pixel di set pada nilai rata-rata dari daerah sekitarnya. Nilai pixel asli menerima nilai yang paling tinggi sedangkan pixel di sekitarnya menerima nilai yang lebih kecil di mana semakin dekat dengan pixel asli maka nilainya akan semakin mendekati. Hal inilah yang mengakibatkan timbulnya sebuah *blur* yang menjaga batas-batas dan *edge* yang baik[5].

### 2.2.3 Thresholding

*Threshold* adalah metode menetapkan suatu nilai, yang kemudian hanya mengambil nilai di atasnya saja atau di bawahnya saja, sedangkan nilai selainnya diabaikan. Hasil dari *Threshold* adalah citra yang tampak nyata perbedaan intensitasnya, yaitu hitam dan putih. Metode ini biasanya digunakan untuk segmentasi atau pemisahan suatu objek dengan selain *background*. *Thresholding* adalah proses mengubah suatu citra berwarna atau berderajat keabuan (*Greyscale*) menjadi citra biner atau hitam putih, sehingga dapat diketahui daerah mana yang termasuk objek dan *background* dari citra secara jelas. Citra hasil *thresholding* biasanya digunakan lebih lanjut untuk proses pengenalan objek serta ekstraksi fitur. Tipe data dari hasil proses

*thresholding* adalah tipe data *float*, yaitu antara 0 sampai dengan 1. Dengan parameter yang di set sebelumnya maka data citra yang jika melebihi batas yang ditentukan akan dibuat menjadi 1 atau putih dan jika di bawah batas yang ditentukan maka akan dibuat menjadi 0 atau hitam[6]. Berdasarkan nilai ambang (*threshold*) T dapat ditunjukkan pada persamaan di bawah ini. Dengan kata lain, nilai intensitas pixel semula dipetakan menjadi 2 nilai, yaitu hitam dan putih.

$$f(x, y) = \begin{cases} 1, & f(x, y) > T \\ 0, & f(x, y) \leq T \end{cases} \quad (2.3)$$

#### 2.2.4 Dilasi

Dilasi adalah proses menambahkan atau mempertebal objek pada gambar biner. Hal ini dilakukan dengan memberikan tambahan pixel pada batasan suatu objek pada suatu gambar. Jumlah pixel yang ditambahkan tergantung dari ukuran dan bentuk dari *structuring element* yang digunakan pada pengolahan gambar. Dimana *structuring element*, merupakan sebuah matrik yang hanya berisi 1 dan 0, yang dapat memiliki berbagai bentuk dan ukuran[7]. Pada umumnya, proses dilasi yang dilakukan pada sebuah citra menghasilkan objek yang lebih besar dan meleburkan titik-titik objek menjadi bagian dari objek berdasarkan kernel yang digunakan.

#### 2.2.5 Erosi

Erosi merupakan kebalikan dari dilasi. Erosi adalah proses untuk mengecilkan ataupun menipiskan objek pada gambar. Berbeda dengan dilasi, apabila erosi dilakukan maka yang dikerjakan adalah memindahkan piksel pada batasan-batasan objek yang akan di erosi. Jumlah dari pixel yang ditambahkan atau dihilangkan bergantung pada ukuran dan bentuk dari *structuring element* yang digunakan untuk memproses gambar tersebut[7]. Pada umumnya, proses erosi yang dilakukan pada sebuah citra menghasilkan objek yang lebih kecil dan menghilangkan titik-titik objek menjadi bagian dari *background* berdasarkan kernel yang digunakan.

#### 2.2.5 Contour

Kontur (*contour*) dapat didefinisikan sebagai suatu keadaan yang terjadi karena adanya perubahan intensitas pada pixel-pixel yang berdampingan. Dengan adanya perubahan intensitas ini, maka titik-titik tepi (*edge*) pada citra dapat terdeteksi. Kontur juga dapat didefinisikan sebagai urutan titik yang dapat menguraikan bentuk atau *region*. Deteksi kontur dapat diimplementasikan dengan menggunakan fungsi `findContours()` pada OpenCV.

Terdapat 2 parameter di sini, yaitu parameter *mode* yang digunakan untuk menemukan kontur dan parameter *method* yang digunakan untuk mendeteksi kontur.

*Mode* yang digunakan yaitu *mode* RETR\_EXTERNAL yang hanya mengambil luaran (*outer*) dari lintasan suatu objek. Dan *method* yang digunakan yaitu *method* CHAIN\_APPROX\_SIMPLE untuk menggambarkan segmen-segmen secara horizontal, vertikal, dan diagonal yang hanya menyisakan titik akhir atau ujung yang diperlukan untuk menggambar garis kontur. Hasil dari fungsi ini dapat memberikan gambar yang terdeteksi kontur dan memuat informasi tentang topologi dari gambar tersebut. Untuk menggambar kontur, digunakan fungsi `drawContours()`. Serta dapat digunakan juga untuk menggambar bentuk objek lain yang memiliki titik batas.

Di dalam kontur terdapat ekstraksi fitur tambahan yang berupa *extreme points*. *Extreme points* adalah titik (*point*) dari atas, bawah, kanan, dan kiri dari suatu objek. Sebelum dapat menemukan titik-titik yang ekstrem di sepanjang kontur, penting untuk dipahami bahwa kontur hanyalah serangkaian koordinat (x, y). Digunakan fungsi NumPy untuk menemukan letak koordinat dari *extreme points*.

### **2.3 Feature Extraction**

*Feature Extraction* (Ekstraksi fitur) menjelaskan informasi yang terkandung dalam suatu pola sehingga tugas mengklasifikasikan pola menjadi mudah. Dalam pengenalan pola dan pemrosesan gambar, ekstraksi fitur adalah bentuk khusus dari pengurangan dimensi. Tujuan utama ekstraksi fitur adalah untuk memperoleh informasi dari data asli dan merepresentasikan informasi tersebut dalam ruang dimensi yang lebih rendah. Mengubah data input ke dalam set fitur disebut ekstraksi fitur. Fitur yang diekstraksi diharapkan dapat memberikan karakteristik dari tipe input ke klasifikasi dengan mempertimbangkan deskripsi properti yang relevan dari gambar ke dalam ruang fitur. bagian dari objek berdasarkan kernel yang digunakan. Jenis fitur tergantung pada jenis sistem yang akan diimplementasikan. Dalam pemrosesan gambar jenis yang paling sering digunakan dapat dibagi menjadi fitur spektral, geometris dan tekstur.

#### **2.3.1 Gray Level Co-occurrence Matrix (GLCM)**

Metode GLCM merupakan suatu metode yang melakukan analisis terhadap suatu piksel pada citra dan mengetahui tingkat keabuan yang sering terjadi. Untuk melakukan analisis citra berdasarkan distribusi statistik dari intensitas pikselnya, dapat dilakukan dengan mengekstrak fiturnya. GLCM menggunakan perhitungan tekstur pada orde kedua. Pengukuran tekstur pada orde pertama menggunakan perhitungan statistika didasarkan pada nilai piksel citra asli semata, seperti varians dan tidak memperhatikan hubungan ketetanggaan piksel. Pada orde kedua, hubungan antarpasangan dua piksel citra asli diperhitungkan.

GLCM merupakan suatu metode untuk melakukan ekstraksi ciri berbasis statistik, perolehan ciri diperoleh dari nilai matriks yang mempunyai nilai tertentu dan membentuk sudut pola. Untuk sudut yang dibentuk dari nilai piksel citra menggunakan GLCM adalah  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$ . Berikut adalah gambaran pembentukan GLCM atas citra dengan 4 tingkat *gray level* pada jarak *distance* ( $d = 1$ ) dan arah  $0^\circ$ .

Dari piksel-piksel tersebut terbentuk *matrix co-occurrence* dengan pasangan pikselnya. Adanya matrik tersebut berdasarkan kondisi bahwa suatu matrik piksel akan mempunyai nilai perulangan sehingga terdapat pasangan arah keabuanannya. Terdapat 14 fitur tekstural yang digunakan untuk mengklasifikasikan citra. Namun, terdapat penelitian berikutnya yang menginvestigasikan karakteristik tekstural dengan menggunakan GLCM dan mengatakan bahwa terdapat 7 parameter yang paling relevan untuk mengklasifikasikan citra berdasarkan hasil matriks *Co-occurrence*. Berikut merupakan 7 parameter GLCM yang akan digunakan :

1. Contrast : Merupakan fitur yang merepresentasikan perbedaan tingkat warna atau skala grayscale yang muncul pada sebuah citra.

$$\sum_{i,j=0}^{n-1} P(i,j) \cdot (i - j)^2 \quad (2.4)$$

2. Energy : Merepresentasikan ukuran keseragaman pada citra. Semakin tinggi kemiripan citra maka akan semakin tinggi pula nilai energy.

$$\sqrt{\sum_{i,j=0}^{n-1} P(i,j)^2} \quad (2.5)$$

3. Correlation : Merepresentasikan keterkaitan linear dari derajat citra grayscale. Correlation bernilai antara -1 dan 1.

$$\sum_{i,j=0}^{n-1} P(i,j) \left[ \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{(\sigma_i^2)(\sigma_j^2)}} \right] \quad (2.6)$$

4. Dissimilarity : Mengukur nilai rata-rata dari perbedaan level grayscale di citra.

$$\sum_{i,j=0}^{n-1} P(i,j) |i - j| \quad (2.7)$$

5. Homogeneity : Merepresentasikan ukuran kebersamaan. Homogeneity akan bernilai tinggi jika semua piksel mempunyai nilai yang uniform.

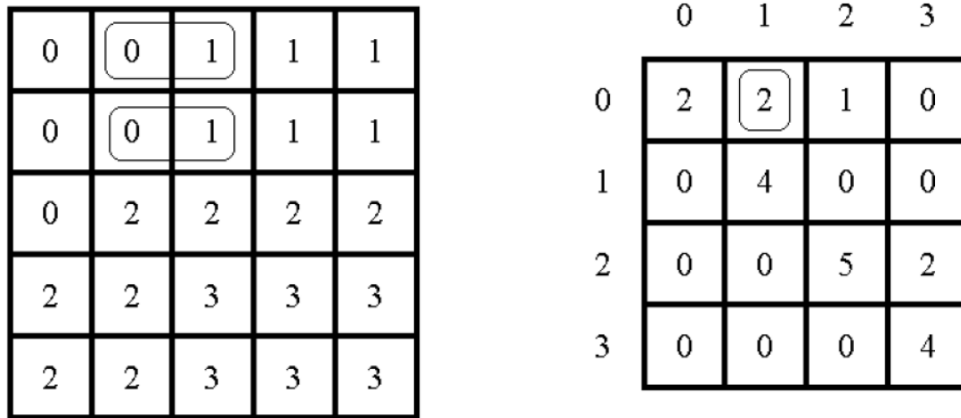
$$\sum_{i,j=0}^{n-1} P(i,j) / [1 + (i - j)^2] \quad (2.8)$$

6. Angular Second Moment (ASM) : Merupakan ukuran homogenitas dari suatu citra.

$$\sum_{i,j=0}^{n-1} P(i,j)^2 \quad (2.9)$$

7. Entropy : Menggambarkan kompleksitas tekstur pada citra.

$$- \sum_{i,j=0}^{n-1} P(i,j) \log (P(i - j)) \quad (2.10)$$



**Gambar 2. 3** (a) contoh citra dengan 4 tingkat keabuan, (b) GLCM pada jarak 1 dan arah 0°

## 2.4 Open CV

OpenCV (*Open Source Computer Vision Library*) adalah sebuah API (*Application Programming Interface*) library yang menyediakan struktur data dan algoritma yang cepat dan efisiensi serta dapat digunakan untuk pengolahan citra dan *real-time computer vision*. Tujuannya adalah agar komputer mempunyai kemampuan yang mirip dengan cara pengolahan visual pada manusia. Karena sifatnya yang *open source*, maka terdapat komunitas pengguna termasuk di dalamnya orang-orang dari perusahaan besar serta pusat-pusat penelitian.

Library pada OpenCV berisi lebih dari 500 fungsi yang menjangkau berbagai area dalam permasalahan visi, pencitraan medis, keamanan, anatarmuka pengguna, kalibrasi kamera, visi stereo, dan robotika. Adapun 5 library yang terdapat pada OpenCV, yaitu :

1. CV, merupakan komponen yang berisikan algoritma dasar pengolahan citra digital dan *computer vision* yang lebih tinggi.
2. ML, merupakan komponen yang berisikan pustaka dari *machine learning* untuk pengklasifikasian statistical dan *clustering tools* yang mencakup algoritma.
3. HighGUI, berisi fungsi-fungsi I/O untuk penyimpanan dan pembacaan video atau gambar.
4. CXCORE, berisi struktur data, *support XML* dan fungsi-fungsi grafis.
5. CvAux, berisi *both defunct areas* dan algoritma eksperimental.

Library OpenCV juga banyak digunakan pada penelitian sebelumnya di berbagai bidang, seperti medis, *monitoring*, HCI (*Human-Computer Interaction*), dan lain-lain.

## 2.5 Support Vector Machine (SVM)

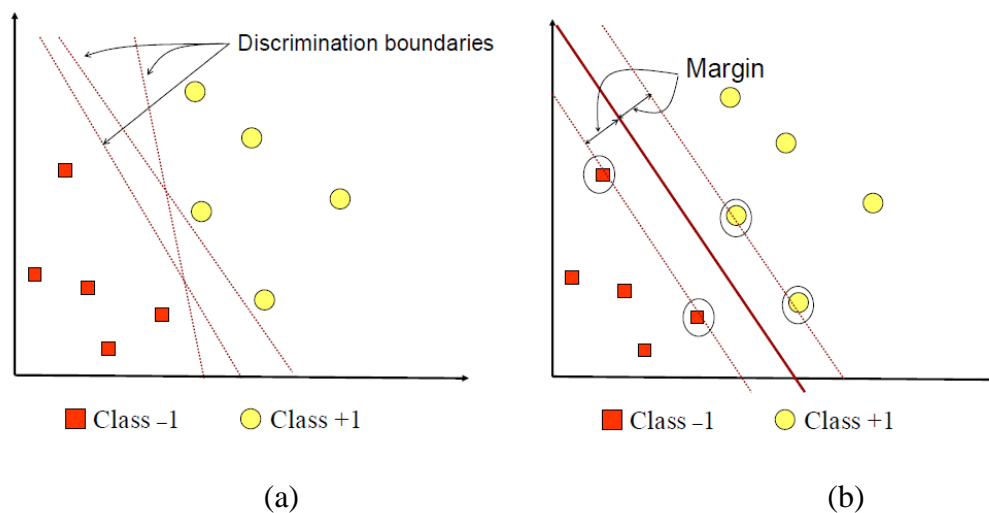
*Support Vector Machine* (SVM) pertama kali diperkenalkan oleh Vapnik pada tahun 1992 sebagai rangkaian harmonis konsep-konsep unggulan dalam bidang *pattern recognition*. SVM adalah metode *learning machine* yang bekerja atas prinsip *Structural Risk Minimization* (SRM) dengan tujuan menemukan *hyperplane* terbaik yang memisahkan dua buah *class* pada *input space*[8]. Pada gambar 3a terdapat beberapa *pattern* yang terbagi dalam dua buah *class* -1 dan *class* +1. *Pattern* yang tergabung dalam *class* -1 dilambangkan dengan kotak warna merah, sedangkan *pattern* yang ada dalam *class* +1 dilambangkan dengan lingkaran warna kuning. Masalah klasifikasi dapat diterjemahkan dengan usaha mencari garis (*hyperplane*) yang memisahkan antara kedua kelompok tersebut. Gambar 3b menunjukkan *hyperplane* yang memisahkan *class* -1 dengan *class* +1 yang memiliki margin tertentu. Margin adalah jarak antara *hyperplane* dengan *pattern* terdekat dari masing-masing *class*. *Pattern* terdekat ini disebut sebagai *support vector*.

Ketika  $\{x_1, \dots, x_n\}$  adalah dataset dan  $y_i \in \{+1, -1\}$  adalah label kelas dari data  $x_i$  dengan  $i = 1, 2, \dots, n$ . Dimana  $n$  adalah banyaknya data. Diasumsikan kedua *class* -1 dan *class* +1 dapat terpisah sempurna oleh *hyperplane*. Sehingga *hyperplane* dinotasikan pada Persamaan 2.11

$$f(x) = \mathbf{w}^T \mathbf{x} + b \quad (2.11)$$

Data  $x_i$  termasuk dalam kelas +1 bila  $\mathbf{w}^T \mathbf{x}_i + b > 0$  dan data  $x_i$  termasuk dalam kelas -1 bila  $\mathbf{w}^T \mathbf{x}_i + b < 0$ . Karena data tersebut dipisahkan secara linier maka persamaan *hyperplane* dapat ditulis seperti Persamaan 2.12

$$\mathbf{w}^T \mathbf{x}_i + b = 0 \quad (2.12)$$



**Gambar 2. 4** SVM mencari Hyperplane yang memisahkan kedua class [8].

Dimana semua observasi harus memenuhi Persamaan 2.13

$$(\mathbf{w}^T \mathbf{x}_i) + b \geq 1 \text{ untuk } y_i = +1 \quad (2.13)$$

$$(\mathbf{w}^T \mathbf{x}_i) + b \leq -1 \text{ untuk } y_i = -1$$

Dari persamaan diatas dapat disimpulkan bahwa nilai dari *support vector* akan selalu sesuai dengan Persamaan 2.14

$$y_i (\mathbf{w}^T \mathbf{x}_i) + b - 1 \geq 0 \text{ untuk } i = 1, 2, \dots, n. \quad (2.14)$$

Menurut Santosa[9], untuk mendapatkan hyperplane terbaik diperlukan nilai margin yang optimal. Nilai margin merupakan nilai jarak terdekat hyperplane dengan data yang paling dekat dengan hyperplane tiap kelas. Nilai margin dapat dihitung dengan Persamaan 2.15

$$\frac{2}{\|\mathbf{w}\|} \quad (2.15)$$

Nilai margin akan optimal apabila nilai  $\|\mathbf{w}\|$  minimal. Oleh karena itu, pencarian hyperplane terbaik dengan nilai margin optimal dapat dirumuskan menjadi masalah optimasi pada Persamaan 2.16

$$\min \frac{1}{2} \|\mathbf{w}\|^2 \quad (2.16)$$

dengan syarat  $y_i (\mathbf{w}^T \mathbf{x}_i) + b \geq 1$ ,  $i = 1, 2, \dots, n$ . Secara umum, persoalan optimasi konstrain pada Persamaan 2.16 akan lebih mudah diselesaikan jika diubah ke dalam fungsi Lagrange pada Persamaan 2.17

$$L_p(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^n \alpha_i \{y_i (\mathbf{w}^T \mathbf{x}_i) + b - 1\} \quad (2.17)$$

Dimana  $\alpha_i \geq 0$  yang merupakan nilai dari koefisien Lagrange. Nilai minimum  $w$  dan  $b$  dari fungsi Lagrange tersebut ada pada Persamaan 2.18

$$\frac{\partial L_p(\mathbf{w}, b, \alpha)}{\partial b} = 0 \rightarrow \sum_{i=1}^n \alpha_i y_i = 0 \quad (2.18)$$

$$\frac{\partial L_p(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \rightarrow \mathbf{w} = \sum_{i=1}^n \alpha_i \mathbf{x}_i y_i \quad (2.19)$$

Untuk mendapatkan dual problem, maka perlu menjabarkan dari Persamaan 2.17 menjadi Persamaan 2.20

$$L_p(\mathbf{w}, b, \alpha) = \frac{1}{2} \mathbf{w}^T \mathbf{w} - \sum_{i=1}^n \alpha_i y_i (\mathbf{w}^T \cdot \mathbf{w}^T) - b \sum_{i=1}^n \alpha_i y_i + \sum_{i=1}^n \alpha_i \quad (2.20)$$

Dengan menggunakan nilai-nilai dari  $w$ , maka didapatkan

$$\mathbf{w}^T \mathbf{w} = \sum_{i=1}^n \alpha_i y_i \mathbf{w}^T \mathbf{x}_i = \sum_{i=1}^n \sum_{j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.21)$$

Maka Persamaan 2.20 menjadi Persamaan 2.22

$$L_D(\mathbf{w}, b, \alpha) = \sum_{i=1}^n \alpha_i - \frac{1}{2} \sum_{i,j=1}^n \alpha_i \alpha_j y_i y_j \mathbf{x}_i^T \mathbf{x}_j \quad (2.22)$$

dengan batasan  $\alpha_i \geq 0$  dimana  $i = 1, 2, \dots, n$  dan  $\sum_{i=1}^n \alpha_i y_i = 0$

Data training dengan  $\alpha_i > 0$  terletak pada hyperplane disebut super vector. Sedangkan data training yang tidak terletak hyperplane mempunyai  $\alpha_i = 0$ . Setelah



ditemukan nilai  $\alpha_i$ , maka kelas dari data yang akan diprediksi atau data testing dapat ditentukan berdasarkan nilai pada fungsi Persamaan 2.23.

$$f(x) = \sum_{i,j=1}^{ns} \alpha_i y_i \mathbf{x}_i \cdot \mathbf{x}_j + b \quad (2.23)$$

Dari hasil perhitungan ini diperoleh  $\alpha_i$  yang kebanyakan bernilai positif. Data yang berkorelasi dengan  $\alpha_i$  yang positif inilah yang disebut *support vector*.

Dot Product  $\mathbf{x}_i \mathbf{x}_j$  dari persamaan 2.23 sering diganti dengan simbol K. K adalah matrik kernel. Matrik kernel ini digunakan untuk membuat dataset yang bersifat non-linear menjadi linear. Contoh dataset yang bersifat non-linear dapat dilihat pada Gambar 2.5. Oleh karena itu dalam SVM terdapat fungsi kernel yang dapat mengubah dataset yang tidak linear menjadi space baru. Terdapat 3 tipe kernel pada SVM, yaitu :

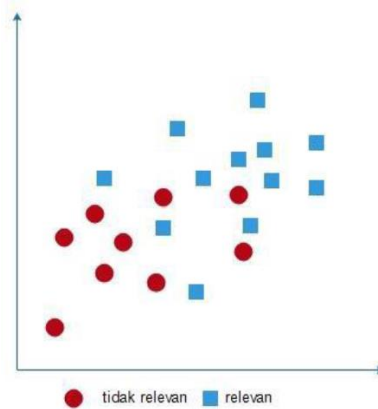
1. Linear :  $K(x_i, x_j) = x^T x$  (2.24)

2. Polynomial :  $K(x_i, x_j) = (x^T x_i + 1)^p$  (2.25)

3. Gaussian :  $K(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2\sigma^2})$  (2.26)

Pemilihan fungsi kernel yang tepat adalah hal yang sangat penting, karena fungsi kernel ini akan menentukan *feature space* dimana *classifier* akan dicari.

Untuk membangun hyperplane yang optimal, SVM menggunakan algoritma pelatihan secara berulang, yang digunakan untuk meminimalkan fungsi *error*. Menurut bentuk fungsi *error*, model SVM dapat diklasifikasikan ke dalam dua kelompok berbeda, yaitu tipe C-SVM dan tipe Nu-SVM. Yang membedakan kedua tipe tersebut adalah parameter C dan parameter Nu. Parameter C dan Nu adalah parameter regularisasi yang membantu menerapkan penalti pada kesalahan klasifikasi yang dilakukan saat memisahkan kelas. Dengan demikian membantu dalam meningkatkan akurasi output. Parameter C berkisar dari 0 hingga tak terbatas dan bisa agak sulit untuk diperkirakan dan digunakan. Parameter Nu beroperasi antara 0-1 dan mewakili batas



**Gambar 2. 5** Dataset Nonlinear [8].

bawah dan atas pada jumlah contoh yang merupakan vektor pendukung dan yang terletak di sisi hyperplane yang salah.

## 2.6 Confusion Matrix

Metode ini menggunakan tabel matriks seperti pada Tabel 2.1 , jika data set hanya terdiri dari dua kelas, kelas yang satu dianggap sebagai positif dan yang lainnya negatif. *True positives* adalah jumlah record positif yang diklasifikasikan sebagai positif, *false positives* adalah jumlah record negatif yang diklasifikasikan sebagai positif, *false negatives* adalah jumlah record positif yang diklasifikasikan sebagai negatif, *true negatives* adalah jumlah record negatif yang diklasifikasikan sebagai negative, kemudian masukkan data uji. Setelah data uji dimasukkan ke dalam confusion matrix, hitung nilai - nilai yang telah dimasukkan tersebut untuk dihitung jumlah *specificity*, *precision*, *accuracy* dan *error*.

1. Specificity =  $TN / (TN + FP)$
2. Precision =  $TP / (TP + FP)$
3. Accuracy =  $(TP + TN) / (TP + FP + TN + FN)$
4. Error =  $(FP + FN) / (TP + FP + TN + FN)$

## 2.7 Rasional

Dalam diagram *Fishbone* pada Gambar 2.6, terdapat beberapa referensi yang terkait dengan penelitian ini. Pada tahap *pre-processing*, semua gambar akan diproses terlebih dahulu agar informasi yang terkandung dalam gambar akan lebih mudah untuk didapatkan. Sebelum dilakukan proses *Automatic Cropping*, *threshold* merupakan proses yang sangat penting untuk menetapkan batasan dari suatu nilai. Menurut R. Manjula *et al* [10], proses *threshold* digunakan untuk membedakan area interior dari organ lain dalam dataset gambar. Algoritma ini juga dapat menemukan ukuran gambar dalam piksel. Menurut Alyaa. H. *et al* [11], proses *thresholding* digunakan untuk mensegmentasi dan mendeteksi kontur pada tumor. Kinerja dari *thresholding* dapat digunakan untuk mengekstraksi area tumor. Setelah dilakukan proses *binary thresholding*, area yang telah di *threshold* kemudian diberi kontur dan dicari area yang paling besar untuk proses *automatic cropping*. Sehingga menurut T. Chithambaram *et*

**Tabel 2. 1** Hasil dari Confusion Matrix

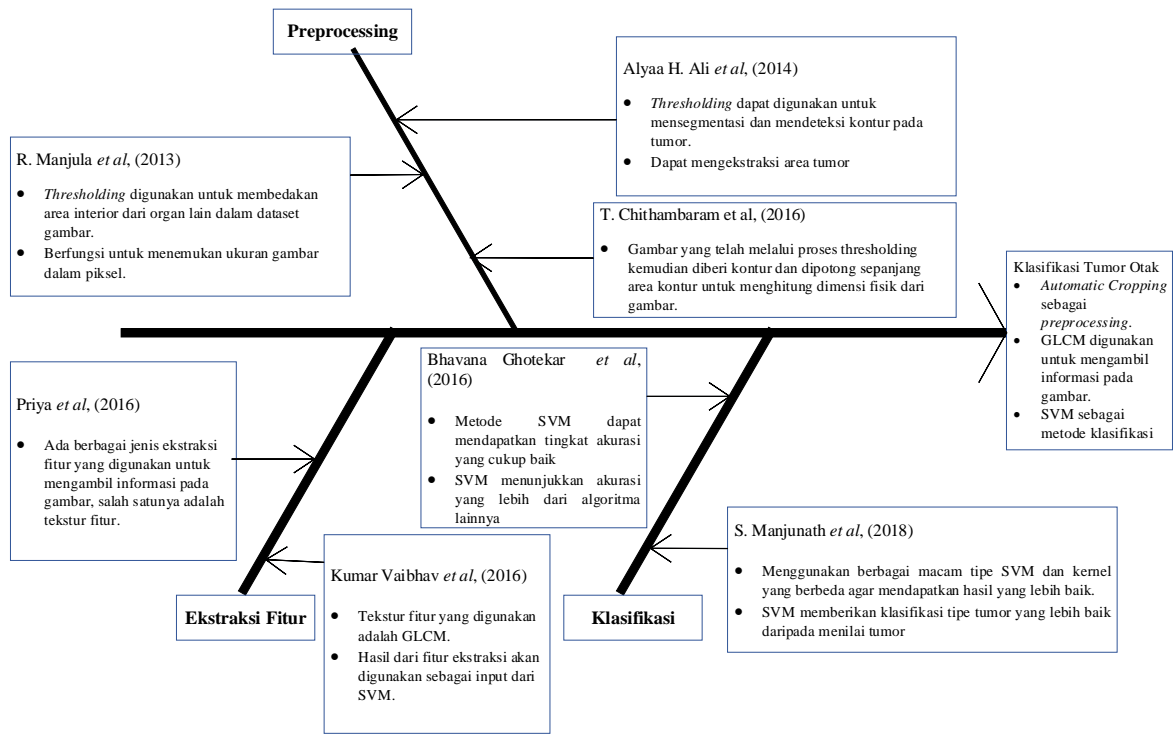
	0	1
0	True Negative (TN)	False Negative (FN)
1	False Positive (FP)	True Positive (TP)

al [12], gambar biner dipotong sepanjang kontur area otak untuk menghitung dimensi fisik dari gambar otak.

Untuk mengambil sejumlah informasi dari gambar yang telah proses sebelumnya, ekstraksi fitur ini sangat dibutuhkan. Menurut Priya *et al* [13], ada berbagai jenis fitur yang digunakan untuk mengekstraksi gambar. Salah satunya adalah tekstur fitur. Tekstur fitur ini nanti akan digunakan sebagai input dari proses pembelajaran dengan menggunakan SVM. Digunakan tekstur fitur untuk menemukan informasi dalam gambar tersebut. Menurut Kumar Vaibhav *et al* [14], tekstur fitur yang digunakan adalah GLCM (*Gray Level Co-occurrence Matrix*) yang diimplementasikan ke dalam tumor otak. Informasi yang terkandung dalam GLCM ini akan dilakukan proses pembelajaran dengan menggunakan SVM.

Bhavana Gothekar *et al* [15], menggunakan SVM untuk mengklasifikasi tumor otak dikarenakan dengan menggunakan SVM mendapatkan hasil akurasi yang cukup baik. SVM menunjukkan akurasi yang lebih baik dari algoritma lainnya. Menurut S. Manjunath *et al* [16], menggunakan metode SVM dengan menggunakan kernel yang berbeda untuk mengklasifikasi jenis tumor otak. Pada metode ini, hasil dari klasifikasi mendapatkan tingkat akurasi yang cukup baik dari tipe SVM dan tipe kernel yang berbeda. SVM memberikan klasifikasi tipe tumor yang lebih baik daripada menilai tumor.

Dari penjabaran beberapa metode yang ada, maka klasifikasi tumor otak dapat menggunakan serangkaian proses tersebut. Automatic Cropping sebagai proses dari Image Preprocessing. Dengan menggunakan GLCM sebagai pengambilan informasi pada gambar yang digunakan untuk proses input learning. Dan yang terakhir melakukan proses klasifikasi dengan menggunakan metode SVM yang memiliki tingkat keakuratan yang lebih baik. Dari proses tersebut akan mempunyai keluaran berupa 2 kelas dari klasifikasi tumor otak, yaitu ada tumor otak atau bukan tumor otak.



**Gambar 2. 6** *Fishbone Diagram..*

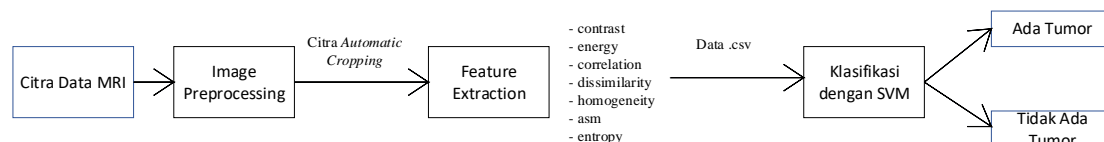
## BAB 3 METODOLOGI

Proses dari penelitian ini dibagi menjadi tiga bagian, yaitu : *Image Pre-processing*, Ekstraksi fitur, dan klasifikasi dengan menggunakan *Support Vector Machine* (SVM). *Image Pre-processing* untuk menghilangkan *noise* serta mengubah ukuran citra. Ekstraksi fitur untuk mengambil informasi yang penting dari citra. Klasifikasi dengan menggunakan *Support Vector Machine* (SVM) untuk mengklasifikasi data tumor otak menjadi dua kelas, yaitu ada tumor dan non tumor. Pada bab ini akan dijelaskan mengenai metode yang akan digunakan dalam pengolahan citra.

### 3.1 Desain Metode Penelitian

Secara umum, alur metode yang akan dibuat dijelaskan dalam diagram blok pada Gambar 3.1. Dalam metode penelitian ini dijabarkan tahapan-tahapan yang akan dilakukan dalam proses klasifikasi tumor otak. Sebelum dilakukan penelitian, dibutuhkan dataset berupa citra gambar otak dengan jumlah 100 gambar. Kemudian dari dataset tersebut diolah ke dalam *Image Pre-processing* untuk menghilangkan *noise* serta mengubah ukuran gambar. Hasil dari *Image Pre-processing* kemudian akan masuk ke dalam proses ekstraksi fitur untuk mengambil informasi-informasi penting yang hasilnya nanti akan menjadi *input* dalam proses klasifikasi dengan SVM.

Pada subbab 3.2 akan menjelaskan *dataset* yang digunakan untuk melatih SVM dalam mengklasifikasi data secara umum. Subbab 3.3 akan menjelaskan proses *Image Pre-processing* untuk melakukan proses normalisasi dan *automatic cropping*. Subbab 3.4 akan menjelaskan proses ekstraksi fitur yang digunakan untuk mendapatkan dan menghasilkan informasi penting yang ada pada gambar. Kemudian pada subbab 3.5 akan menjelaskan proses klasifikasi dengan menggunakan metode SVM untuk memisahkan dua kelas antara ada tumor dan non tumor.



**Gambar 3. 1** Diagram Blok Metode Penelitian

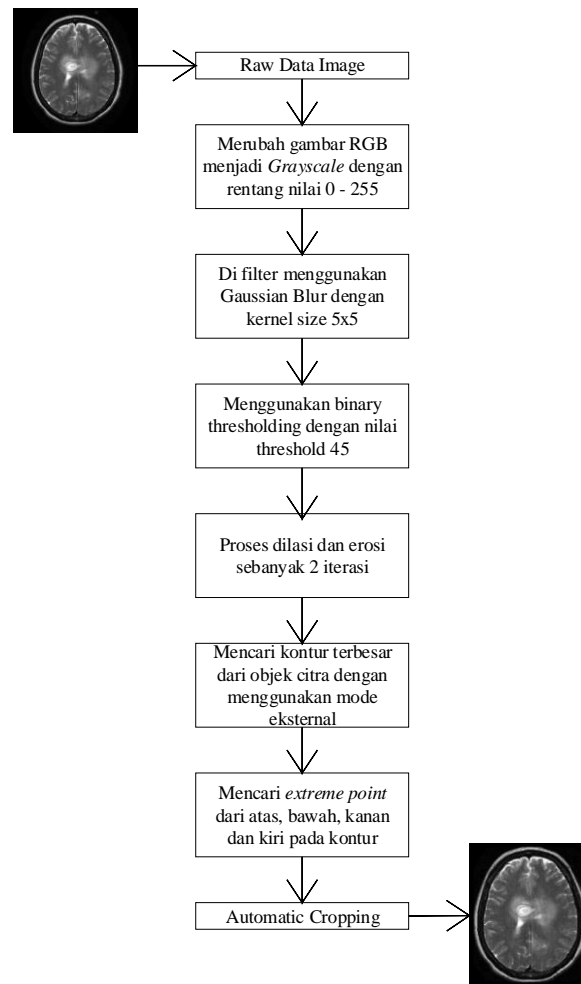
### 3.2 Dataset

Dataset yang digunakan dalam penelitian ini digunakan untuk melatih SVM dalam mengklasifikasi data secara umum. Dataset gambar otak yang akan digunakan didapatkan dari “Brain Tumor Images Dataset” serta “Brain MRI Images for Brain Tumor Detection” yang berasal dari Kaggle. Data dapat diunduh dan di dalam folder terbagi menjadi dua bagian, yaitu data “yes” yang berarti terdapat tumor dan sebaliknya data “no” yang berisi non tumor. Dataset tersebut disimpan dalam format .JPG dan dengan pixel yang berbeda-beda. Jumlah gambar yang akan digunakan merupakan gabungan dari dua dataset yang berbeda dengan jumlah 100 data.

### 3.3 Image Pre-processing

Bagian ini merupakan salah satu bagian yang paling penting sebelum data diolah ke dalam *Machine Learning*. Pada penelitian ini, dataset citra yang telah diunduh kemudian menjadi input dari *pre-processing*. Data gambar yang terdapat dalam folder di muat dalam *workspace* Jupyter Notebook. Kemudian data tersebut diubah ke dalam citra keabuan (*grayscale*). Setelah diubah ke *grayscale*, dilanjutkan dengan proses Gaussian Blur. Dengan mengimplementasikan matriks kernel Gaussian Blur 5 x 5, kemudian kernel tersebut dikonvolusikan dengan citra *grayscale*. Jika hasil konvolusi menghasilkan nilai piksel negatif, nilai tersebut dijadikan 0. Sebaliknya jika hasil konvolusi menghasilkan nilai piksel lebih besar dari nilai *grayscale* maksimum, maka nilai tersebut dijadikan nilai *grayscale* maksimum. Selanjutnya di proses ke dalam proses *Threshold*. *Threshold* merupakan proses konversi citra *grayscale* menjadi citra biner atau hitam putih. Nilai *threshold* yang digunakan yaitu 45 yang bertujuan untuk mengklasifikasikan nilai piksel tersebut. Jika nilai piksel pada citra melebihi 45, maka nilai piksel tersebut memiliki nilai yang ditetapkan yaitu 255. Jenis *threshold* yang digunakan yaitu *binary threshold*. Selanjutnya dilakukan proses menipiskan (*erode*) dan menebalkan (*dilate*) selama 2 iterasi yang bertujuan untuk menghilangkan beberapa bagian kecil yang terdapat *noise*.

Dari gambar yang telah di *threshold*, cari kontur terbesarnya dan kemudian dilakukan proses *cropping*. Mode pengambilan kontur memakai mode pengambilan eksternal. Dimana yang dilihat dan diambil pada kontur tersebut adalah bagian luar dari suatu objek pada citra. Mode pendekatan yang digunakan menggunakan metode simple, yang berfungsi untuk mengkompres segmen secara *horizontal*, *vertical* dan *diagonal* serta menyisakan titik akhir pada citra untuk menggambar garis kontur. Setelah dideteksi luaran objek dengan kontur, kemudian didapatkan nilai *tuple* yang sesuai pada kontur dan gambar kontur di area tersebut. Pada kontur tersebut, akan diambil titik (*point*) dari nilai terkecil dan terbesar pada sisi atas, bawah, kanan, dan



**Gambar 3. 2** Diagram Blok proses Image Pre-processing

kiri gambar serta lakukan proses pemotongan (*cropping*) gambar. Proses tersebut dinamakan dengan *Automatic Cropping*.

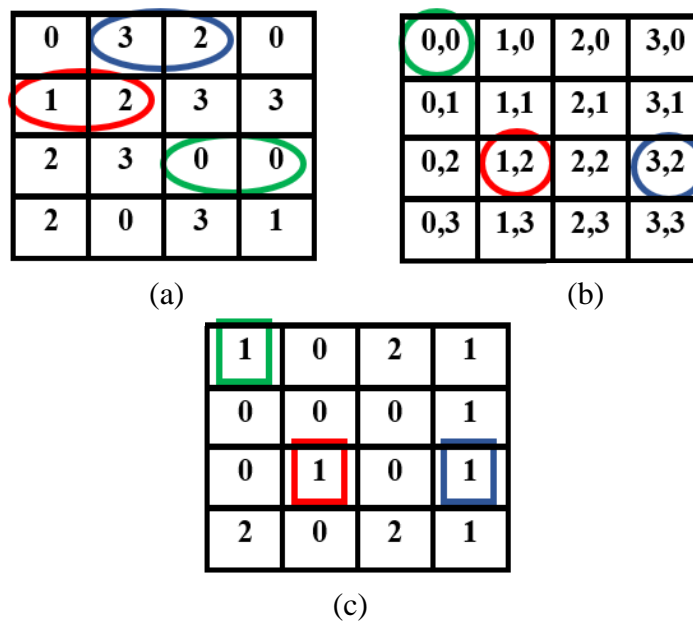
### 3.4 *Feature Extraction*

Pada *feature extraction*, gambar yang telah di-*cropping* sebelumnya, akan dicari informasinya yang terkandung dalam suatu pola pada gambar tersebut sehingga akan dapat dengan mudah untuk diproses ke dalam *machine learning*. *Feature extraction* yang digunakan menggunakan fitur tekstur. Digunakan metode GLCM (*Gray Level Co-occurrence Matrix*) untuk memperoleh informasi pada gambar tersebut. Dibutuhkan 7 fitur yang diperoleh dari gambar tersebut. 7 fitur tersebut yaitu : *Contrast, Correlation, Energy, Dissimilarity, ASM (Angular Second Moment), Homogeneity,*

dan *Entropy*. Setelah didapatkan 7 fitur tersebut, kemudian hasil tersebut disimpan ke dalam bentuk .CSV agar dengan mudah dibaca dan diolah ke dalam *machine learning*.

Cara untuk mendapatkan fitur tersebut adalah dengan cara membuat matriks *co-occurrence* terlebih dahulu. Matriks *co-occurrence* yang akan dibuat adalah hubungan spasial antara jarak ( $d = 1$ ) dan sudut  $0^\circ$ . Hasil dari matriks *co-occurrence* ini berupa jumlah hubungan spasial yang terdapat matriks *grayscale*. Setelah didapatkan nilai dari matriks *co-occurrence*, maka proses selanjutnya adalah menghitung nilai ciri dari setiap fiturnya. Dari setiap 7 fitur tersebut, masing-masing fitur diambil nilai cirinya kemudian disimpan ke dalam file .CSV. Hasil ekstraksi fitur ini nanti akan menjadi masukan dari proses klasifikasi tumor otak. Proses tersebut dapat dilihat pada Gambar 3.3.

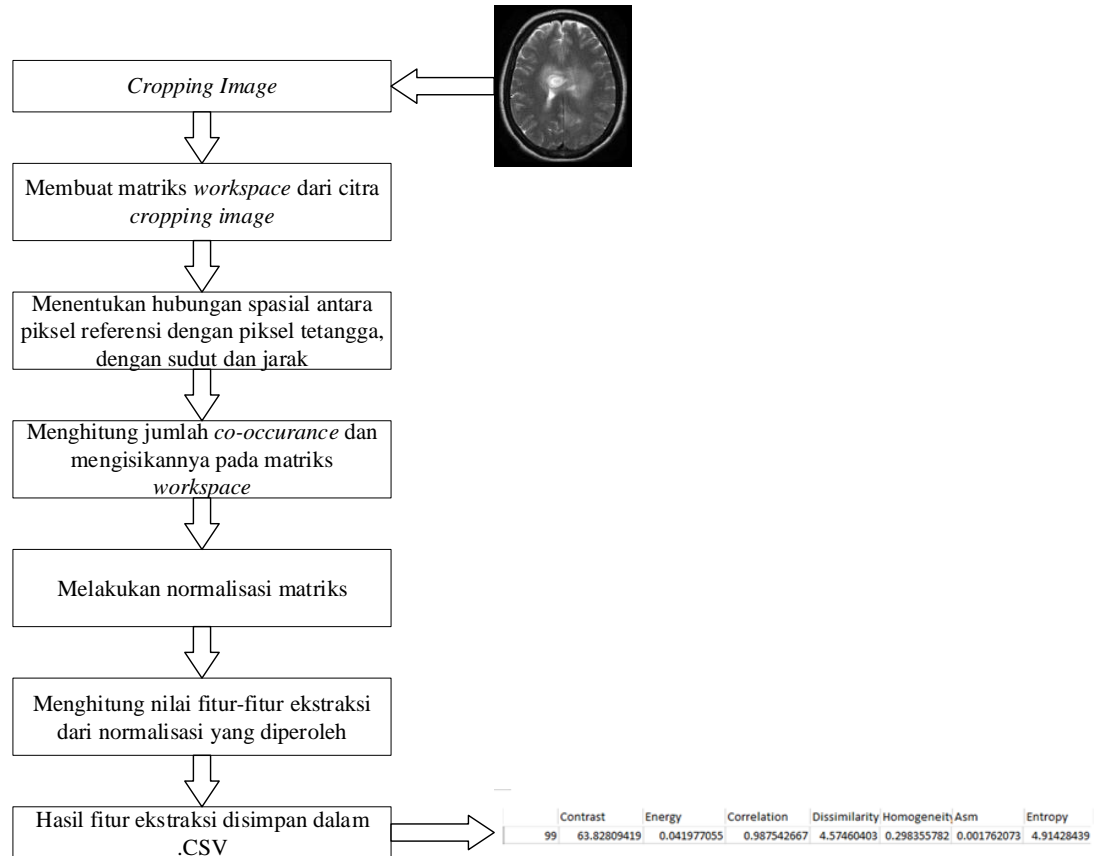
Gambar 3.3 (a) merupakan *pixel* dari gambar yang telah di proses sebelumnya. Kemudian membuat matriks *co-occurrence* dengan menggunakan jarak ( $d = 1$ ) dan sudut  $0^\circ$  yang terlihat seperti Gambar 3.3 (b). Dari matriks tersebut kemudian hitung jumlah *co-occurrence* dan hasil jumlah tersebut diisikan ke dalam matriks yang telah dibuat seperti pada Gambar 3.3 (c). Matriks yang telah terisi oleh jumlah piksel tersebut, lalu di normalisasikan untuk mengubah ke dalam bentuk probabilitas sehingga



**Gambar 3.3** (a) contoh citra dengan 4 tingkat keabuan, (b) Matrix Co-occurrence, (c) GLCM pada jarak 1 dan arah  $0^\circ$



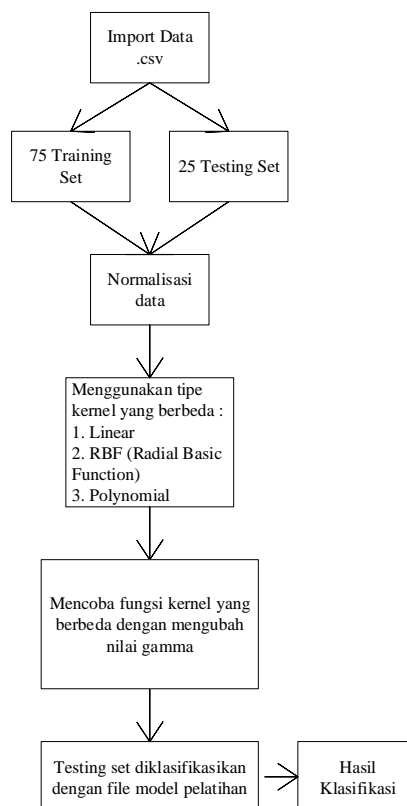
jumlahnya bernilai 1. Cara menormalisasikannya adalah dengan cara nilai piksel dibagi dengan jumlah nilai seluruh piksel yang terdapat pada matriks. Pada Gambar 3.3 (c), jumlah total keseluruhan piksel yang didapat ada 12, sehingga jumlah piksel tiap matriks dibagi dengan 12. Kemudian dihitung nilai fitur-fitur ekstraksi dari normalisasi yang diperoleh. Untuk proses alur fitur ekstraksi ini dapat dilihat pada Gambar 3.4.



**Gambar 3. 4** Diagram Blok proses Feature Extraction

### 3.5 Support Vector Machine (SVM)

Bagian ini merupakan proses untuk mengklasifikasikan data tumor menjadi ada tumor dan non tumor. Dari hasil *feature extraction* yang telah disimpan ke dalam ekstensi .CSV, kemudian data tersebut dibaca dan di tentukan bagian kolom X dan y, lalu dibagi menjadi 2 kelas, yaitu data *training set* dan *testing set*. Data tersebut dibagi dengan menggunakan test size sebesar 0.25, sehingga perbandingan jumlah data *training set* dan data *testing set* terbagi menjadi 75 data *training* dan 25 data *testing*. Data tersebut dibagi secara acak oleh program. Terdapat proses *Standard Scaler* yang bertujuan untuk menstandarisasikan fitur pada kolom X sebelum diaplikasikan ke dalam *machine learning*. Parameter yang digunakan yaitu parameter *fit* dan *fit\_transform*. *Fit* digunakan untuk menghitung *mean* dan *standard deviation* untuk proses *scaling*. Sedangkan untuk *fit\_transform* merupakan gabungan dari *fit* dan *transform* yang artinya sesuaikan (*fit*) dengan data kemudian di *transform*. Selanjutnya masuk ke proses klasifikasi. Proses klasifikasi ini akan diulang dengan menggunakan beberapa tipe SVM yang berbeda-beda, beberapa tipe kernel serta nilai gamma yang akan diubah-ubah.



**Gambar 3. 5** Diagram blok metode Klasifikasi

### 3.6 Metode Pengujian

Metode pengujian yang dilakukan adalah menentukan metode klasifikasi yang digunakan untuk melatih data *training*. Klasifikasi yang digunakan adalah SVM. Data tersebut akan dilatih dengan menggunakan berbagai tipe SVM. Untuk tipe SVM yang digunakan ada 2, yaitu klasifikasi SVM tipe 1 (C-SVM) dan klasifikasi tipe 2 (Nu-SVM). Setiap tipe SVM, diuji dengan berbagai macam kernel seperti linear, RBF (*Radial Basis Function*) serta polynomial.

Pada proses klasifikasi, yang harus dilakukan pertama kali adalah menginisiasi parameter yang akan digunakan. Parameter yang akan digunakan yaitu parameter  $C = 1.0$ ,  $\nu = 0.5$ ,  $\gamma = 0.5, 0.7$ ,  $\epsilon = 0.001$  serta  $\lambda = 0.5$ . Setelah parameter ditentukan, kemudian memasukkan data *training*, lalu mencari nilai *dot product* dari setiap data dengan memasukkan fungsi kernel ( $K$ ). Rumus fungsi kernel yang umum seperti pada Persamaan 2.24 sampai dengan Persamaan 2.26. Sebelumnya data di transpose terlebih dahulu dengan menggunakan perkalian matriks  $A \times A^T$ . Pada metode kernel, setiap data akan dibandingkan dengan dirinya dan data lainnya. Setelah didapatkan hasil perkalian *dot product*, kemudian dilakukan proses menghitung matriks dengan menggunakan Persamaan 3.1.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \quad (3.1)$$

Lalu mencari nilai error dari Persamaan 3.2.

$$E_i = \sum_{j=1}^l \alpha_j D_{ij} \quad (3.2)$$

Dari hasil error tersebut, kemudian dihitung delta *alpha* dengan menggunakan Persamaan 3.3.

$$\delta \alpha_i = \min \{ \max [\gamma (1 - E_i) - \alpha_i] C - \alpha_i \} \quad (3.3)$$

Setelah mencari nilai dari delta *alpha*, selanjutnya akan dilakukan proses mencari nilai *alpha* yang baru dengan menggunakan Persamaan 3.4.

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (3.4)$$

Apabila hasil *alpha* lebih besar dari epsilon, maka iterasi akan terus berlanjut sampai nilai *alpha* konvergen (tidak ada perubahan signifikan). Kemudian mencari nilai bias dengan menggunakan Persamaan 3.5.

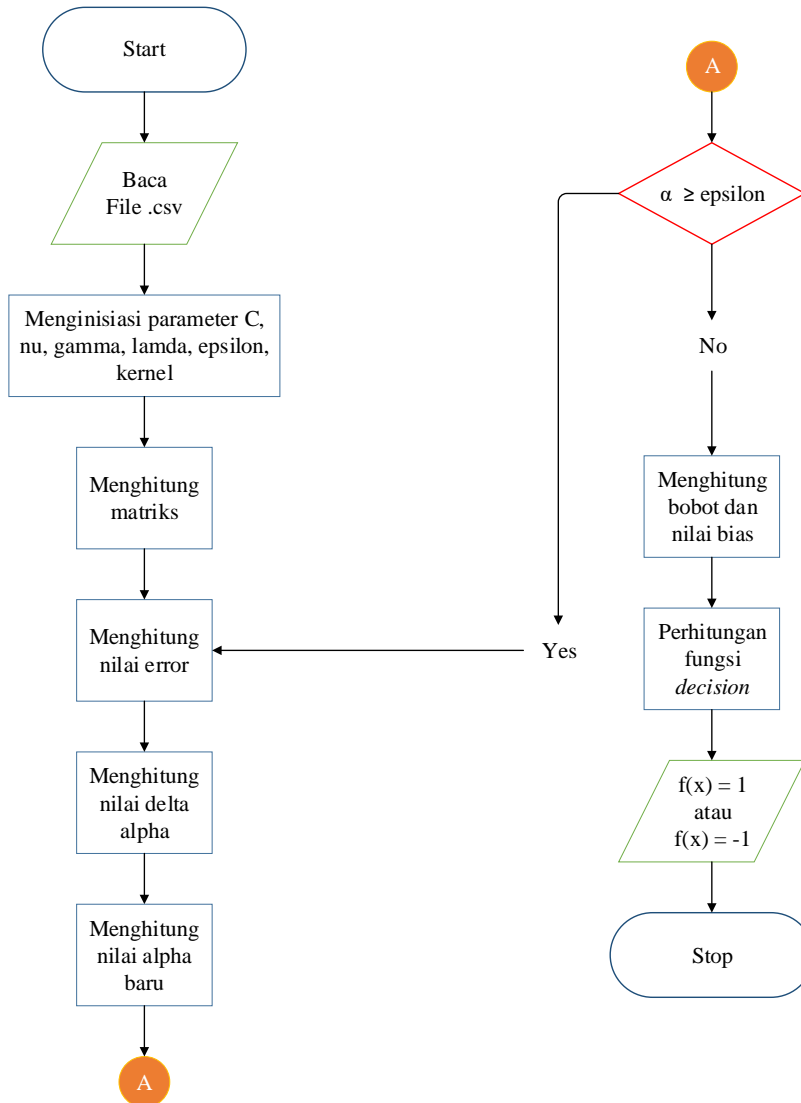
$$b = -\frac{1}{2} ((\mathbf{w}^T \mathbf{x}_{i+}) + (\mathbf{w}^T \mathbf{x}_{i-})) \quad (3.5)$$

Pada Persamaan 3.5, terdapat variabel  $\mathbf{w}$  yang belum memiliki nilai, sehingga perlu mencari nilai dari  $\mathbf{w}$ . Nilai  $\mathbf{w}$  merupakan bobot dari *dot product* data dengan *alpha* terbesar pada masing-masing kelas. Setelah didapatkan nilai  $\mathbf{w}$  maka dapat dicari nilai dari biasnya.

Setelah mendapatkan nilai dari  $\alpha$ , bias ( $b$ ) dan bobot ( $\mathbf{w}$ ), maka dapat dilakukan proses pengujian untuk menentukan fungsi keputusan (*decision*). Perhitungan fungsi keputusan dapat menggunakan Persamaan 3.6.

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i \cdot x_j) + b \quad (3.6)$$

Sehingga dapat diketahui data mana yang akan masuk ke kelas tumor dan kelas non tumor. Proses training tersebut dapat dilihat pada Gambar 3.6.



**Gambar 3. 6** Flowchart learning SVM

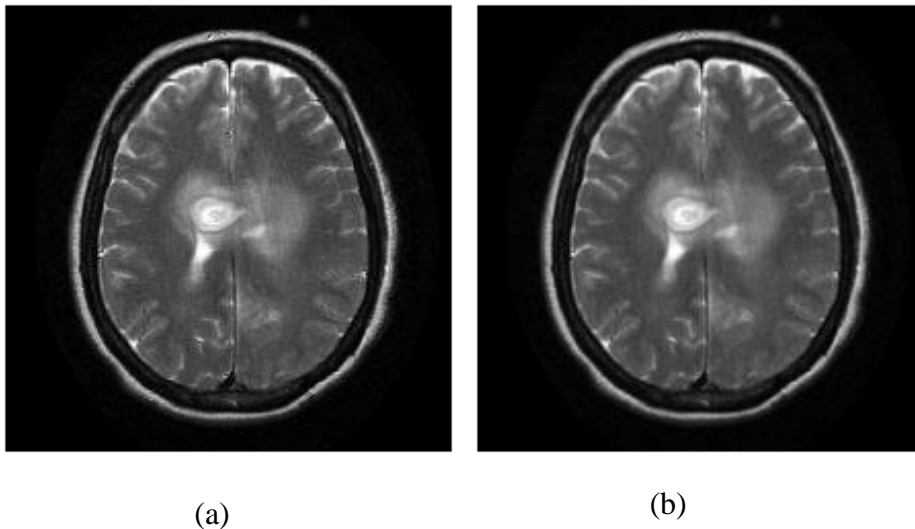
## BAB 4 PENGUJIAN SISTEM

### 4.1 *Pre-Processing*

Pada tahap *pre-processing*, citra masukan yang digunakan diambil dari dataset tumor otak yang telah dijelaskan pada subbab 3.2. Dari dataset tersebut, dipilih salah satu citra dimana pengujian ini, citra tumor 99.jpg digunakan. Citra tersebut kemudian dipanggil dan dijalankan dalam program untuk proses *Automatic Cropping*. Lokasi citra pada berkas komputer dituliskan secara manual dalam *source code*. Citra tersebut kemudian dibuka dan dibaca dengan menggunakan *library* OpenCV pada Python dalam bentuk matriks dengan ukuran panjang, lebar dan warna citra tersebut. Warna dari citra tersebut adalah RGB. Nilai matriks tersebut kemudian yang akan menjadi masukan operasi matriks untuk perhitungan Gaussian blur, *threshold* serta *contour* untuk proses *automatic cropping*.

#### 4.1.1 *Gaussian Blur*

Dataset yang telah terbaca oleh program, yang semula berwarna RGB diganti dan diubah ke dalam bentuk *grayscale*. Kemudian digunakan filter untuk mengurangi komponen citra yang mempunyai frekuensi tinggi. Filter yang digunakan yaitu Gaussian filter. Filter ini memberikan hasil berupa efek *blurring* pada citra yang ditunjukkan pada Gambar 4.1. Hasil ini memberikan efek yang cukup baik kepada citra dikarenakan citra masukan yang digunakan memiliki sedikit *noise*.



**Gambar 4. 1** Hasil dari proses Gaussian Blur (b)

#### 4.1.2 *Threshold*

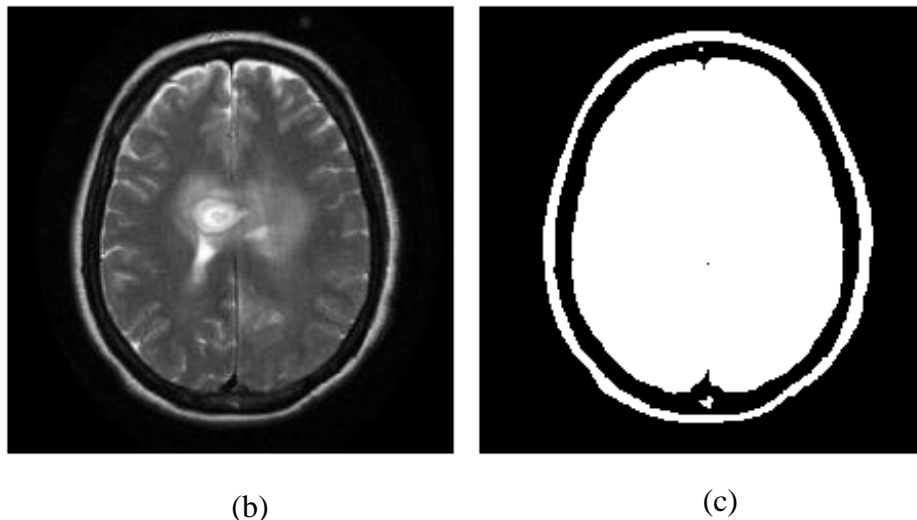
Citra yang telah melalui proses Gaussian blur, akan masuk ke dalam proses *thresholding*. Hasil dari citra ini akan menunjukkan intensitas warna *binary* atau hitam putih. Sehingga hasil dari proses ini akan sangat membantu untuk menemukan batasan dalam objek suatu citra. Hasil dari *thresholding* ini ditunjukkan dalam Gambar 4.2.

#### 4.1.3 *Contour*

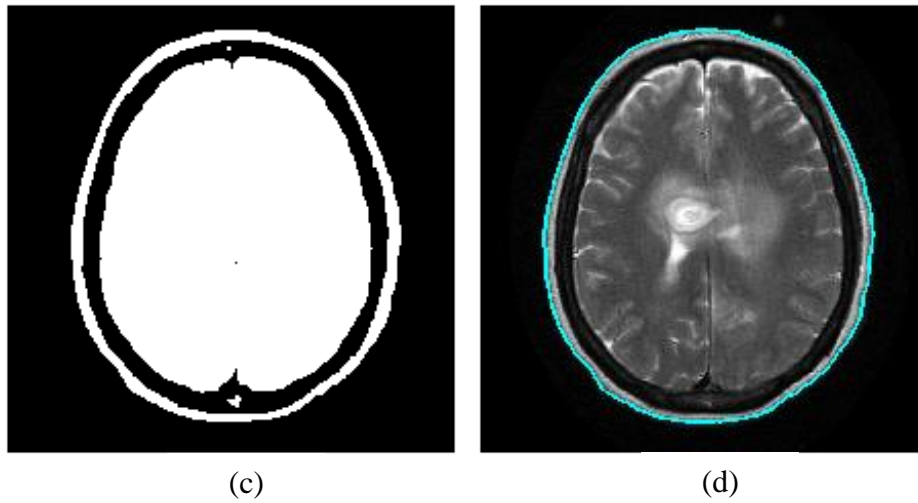
*Contour* digunakan untuk menemukan batasan citra yang akan diproses ke tahap selanjutnya. Karena pada *library* OpenCV pada kontur menggunakan mode *External*, maka kontur akan memberikan gambaran pada garis tepi dari batasan luar citra tersebut. Proses ini berguna untuk menentukan titik poin yang akan digunakan dalam proses *automatic cropping*. Hasil dari proses *contour* dapat dilihat pada Gambar 4.3.

#### 4.1.4 *Automatic Cropping*

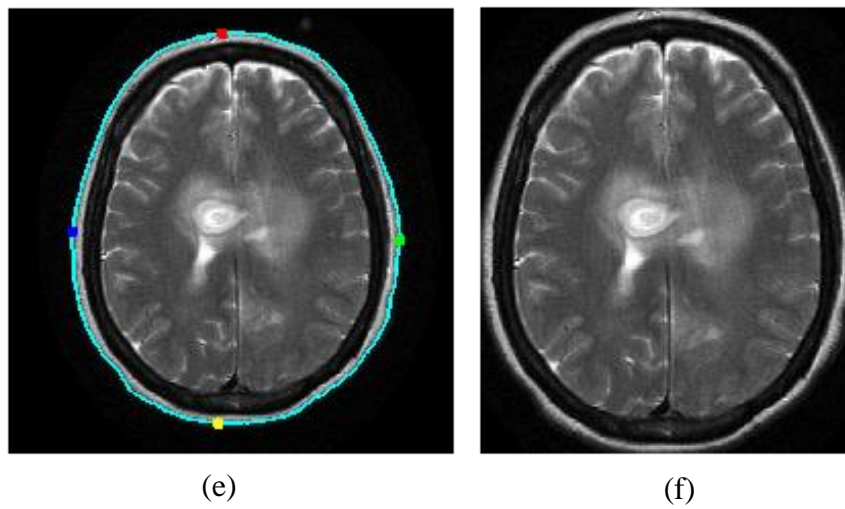
Ketika proses *contour* telah dilakukan, maka tahapan selanjutnya adalah menentukan lokasi dari *extreme point* pada citra *contour*. Proses ini dilakukan untuk mendapatkan titik atas, titik bawah, titik kanan, dan titik kiri untuk memotong tepian luar *contour* dengan *backgroundnya*. Hasil dari proses ini akan menjadi masukan pada tahapan ekstraksi fitur. Sehingga hasil dari proses *automatic cropping* dapat dilihat pada Gambar 4.4.



**Gambar 4. 2** Hasil dari proses Thresholding (c)



**Gambar 4. 4** Hasil dari proses Contour (d)



**Gambar 4. 3** Hasil dari proses Automatic Cropping (f)

## 4.2 Feature Extraction

Data yang telah di proses ke dalam pre-processing akan dilanjutkan ke dalam ekstraksi fitur. Ekstraksi fitur yang digunakan adalah GLCM dan 7 fitur yang akan diambil informasinya pada citra adalah *Contrast*, *Energy*, *Correlation*, *Dissimilarity*, *Homogeneity*, *Angular Second Moment (ASM)* serta *Entropy*. Untuk membuat matriks *co-occurrence* dan ekstraksi fitur tersebut digunakan *library* *skimage.feature* pada *python*. Data citra yang telah terambil fiturnya kemudian akan disimpan ke dalam file *.CSV*. Hasil dari ekstraksi fitur ini akan menjadi masukan untuk proses *learning*. Berikut merupakan 7 fitur yang terdapat pada 100 dataset dengan label yang telah ditentukan pada Tabel 4.1.

Pada Gambar 4.5, merupakan hasil dari data fitur yang telah diubah ke dalam bentuk titik-titik (*scatter*) dan dikelompokkan berdasarkan kelas tumor atau non tumor dengan perbandingan fitur 7 x 7. Terdapat dua contoh gambar dari masing-masing gambar tumor dan gambar non tumor yang akan dibandingkan pada hasil dari masing-masing fitur ini. Gambar 4.6 merupakan perbandingan dari fitur *Contrast* dan *Correlation*. Kemudian pada Gambar 4.7 terdapat fitur *Energy* yang akan dibandingkan dengan fitur *Dissimilarity*. Fitur *ASM* akan dibandingkan dengan fitur *Homogeneity* pada Gambar 4.8. Serta fitur *Entropy* akan dibandingkan dengan fitur *Dissimilarity* yang terdapat pada Gambar 4.9.

**Tabel 4. 1** Hasil dari proses ekstraksi fitur dengan GLCM

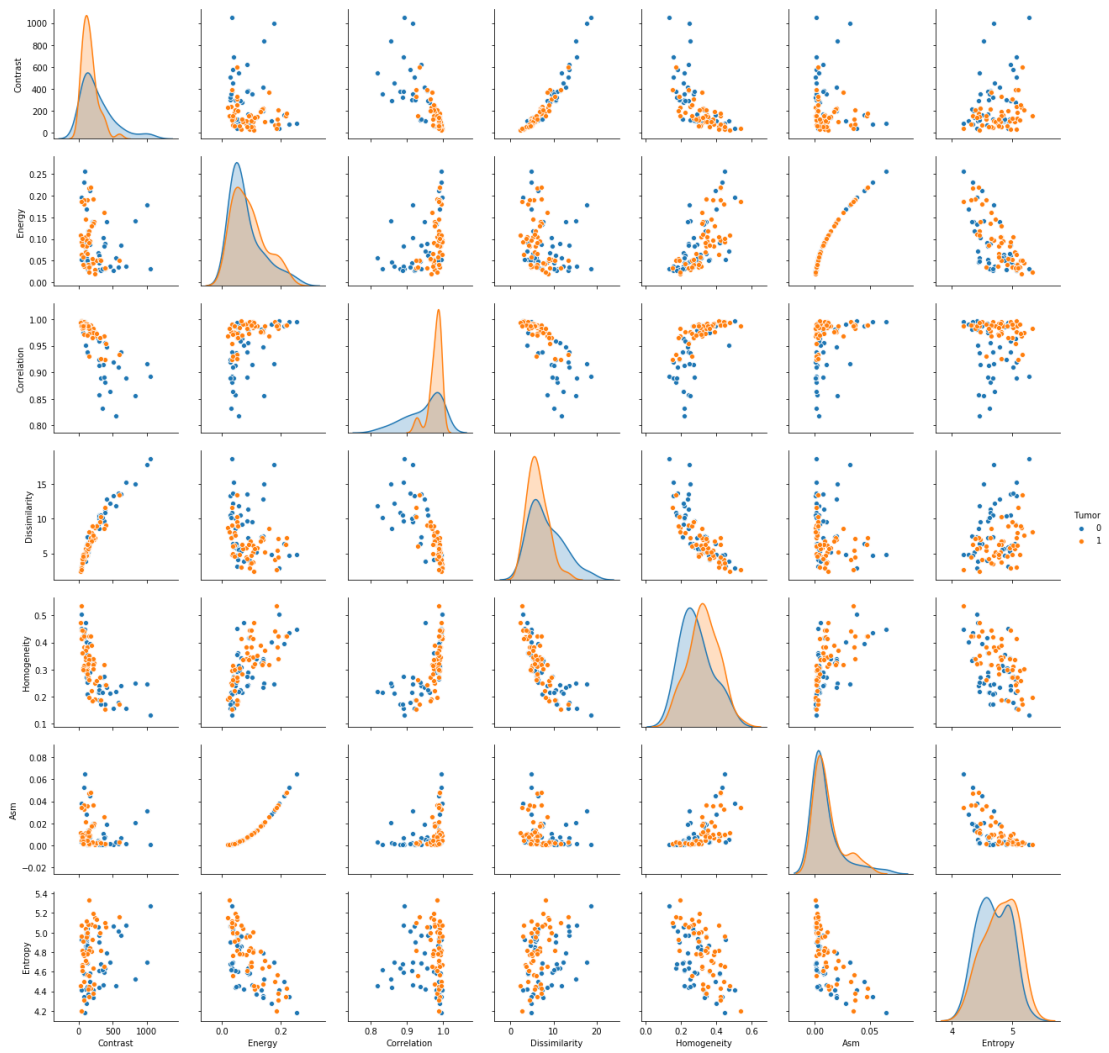
	Contrast	Energy	Correlation	Dissimilarity	Homogeneity	ASM	Entropy	Tumor
0	65.642	0.052	0.975	4.14	0.361	0.003	4.6	0
1	125.82	0.032	0.938	7.319	0.21	0.001	4.685	0
2	138.04	0.118	0.987	5.238	0.339	0.014	4.375	0
3	204.39	0.139	0.983	7.349	0.341	0.019	4.355	0
4	296.06	0.046	0.857	8.543	0.241	0.002	4.44	0
5	213.61	0.139	0.981	7.513	0.332	0.019	4.335	0
6	206.94	0.133	0.981	7.351	0.325	0.018	4.412	0
7	108.23	0.168	0.991	4.831	0.401	0.028	4.278	0
8	165.43	0.212	0.988	6.38	0.397	0.045	4.496	0
9	384.74	0.089	0.956	11.35	0.242	0.008	4.636	0
10	168.38	0.088	0.98	7.195	0.29	0.008	4.589	0
11	104.88	0.064	0.989	5.695	0.31	0.004	4.864	0
12	71.072	0.091	0.992	3.968	0.416	0.008	4.575	0
13	110.19	0.054	0.988	5.797	0.304	0.003	4.913	0



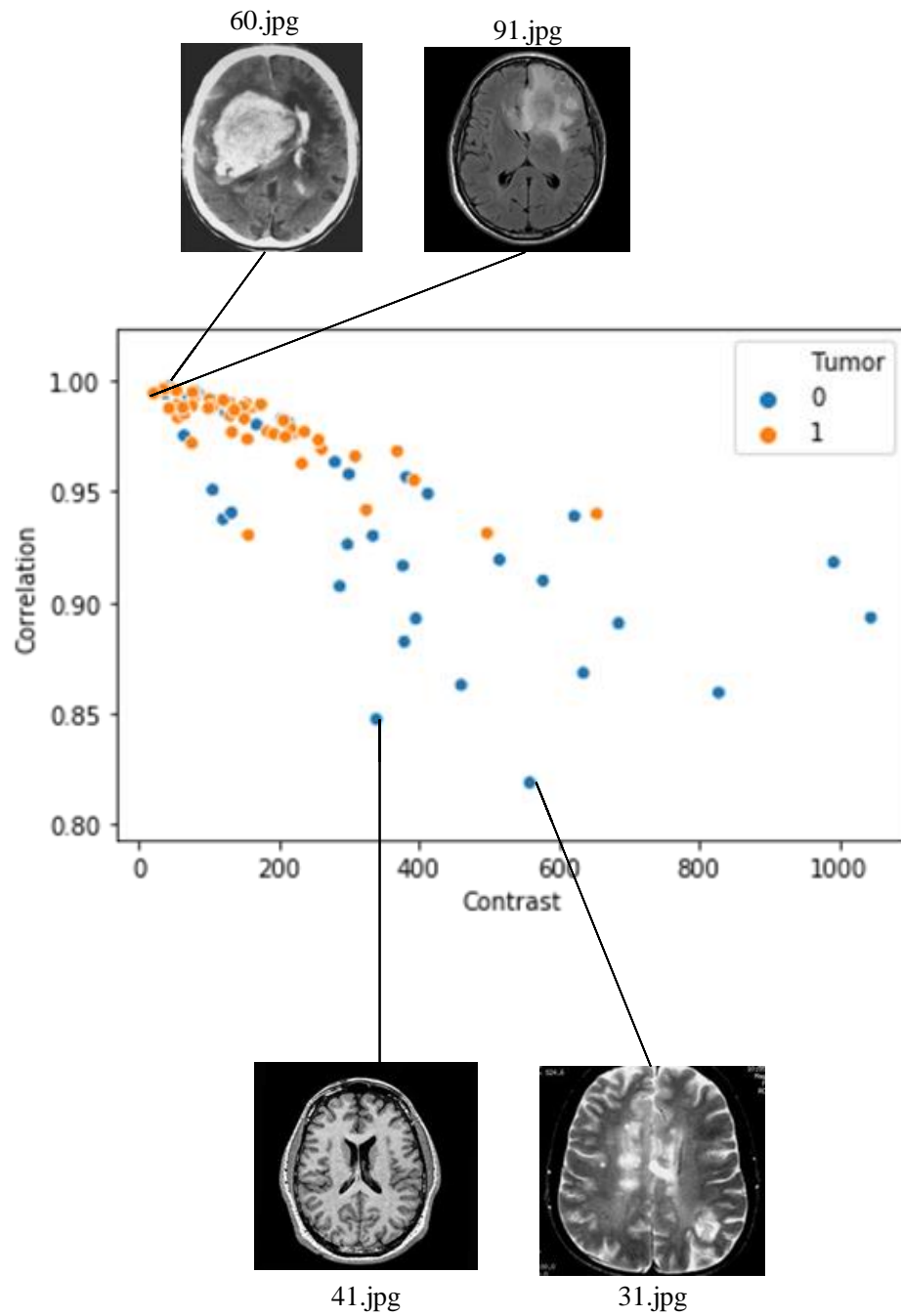
14	110.08	0.056	0.988	5.948	0.293	0.003	4.933	0
15	99.093	0.05	0.988	5.618	0.294	0.003	4.938	0
16	356.15	0.103	0.915	9.57	0.271	0.011	4.466	0
17	108.21	0.04	0.988	5.96	0.273	0.002	5.008	0
18	122.34	0.036	0.987	6.357	0.264	0.001	5.043	0
19	119.38	0.048	0.988	6.215	0.27	0.002	4.985	0
20	108.8	0.059	0.989	5.67	0.295	0.003	4.852	0
21	109.94	0.057	0.989	5.559	0.313	0.003	4.829	0
22	379.24	0.032	0.881	10.87	0.172	0.001	4.639	0
23	72.862	0.092	0.992	3.951	0.417	0.009	4.586	0
24	1046.9	0.032	0.893	18.65	0.131	0.001	5.273	0
25	280.37	0.068	0.963	9.145	0.214	0.005	5.054	0
26	299.92	0.046	0.913	10.07	0.217	0.002	4.61	0
27	832.54	0.142	0.856	15.05	0.25	0.02	4.53	0
28	688.58	0.038	0.89	15.3	0.157	0.001	5.071	0
29	106.33	0.073	0.95	3.839	0.472	0.005	4.422	0
30	624.71	0.085	0.938	13.58	0.24	0.007	4.969	0
31	509.82	0.027	0.919	13.34	0.156	7E-04	5.064	0
32	414.77	0.139	0.948	12.81	0.234	0.019	4.791	0
33	577.34	0.036	0.91	13.63	0.179	0.001	5.017	0
34	38.257	0.052	0.994	3.185	0.452	0.003	4.932	0
35	294.75	0.062	0.958	10.53	0.224	0.004	4.974	0
36	999.49	0.178	0.917	17.76	0.247	0.032	4.696	0
37	300.24	0.029	0.924	10.41	0.188	8E-04	4.903	0
38	320.85	0.028	0.889	10.58	0.173	8E-04	4.68	0
39	377.78	0.084	0.89	9.688	0.273	0.007	4.597	0
40	544.26	0.057	0.819	11.89	0.218	0.003	4.453	0
41	352.12	0.031	0.832	10.15	0.216	1E-03	4.619	0
42	120.99	0.048	0.937	6.419	0.252	0.002	4.45	0
43	456.46	0.038	0.864	12.26	0.213	0.001	4.7	0
44	125.82	0.032	0.938	7.319	0.21	0.001	4.685	0
45	124.77	0.051	0.986	4.647	0.356	0.003	4.873	0
46	44.635	0.196	0.997	2.891	0.503	0.038	4.41	0
47	123.95	0.066	0.99	5.271	0.341	0.004	4.964	0
48	83.979	0.255	0.994	4.833	0.449	0.065	4.184	0
49	79.32	0.23	0.995	4.763	0.434	0.053	4.342	0
50	47.075	0.097	0.996	3.48	0.422	0.009	4.674	1
51	54.088	0.104	0.995	3.773	0.416	0.011	4.817	1
52	79.822	0.084	0.992	4.488	0.383	0.007	4.775	1

53	110.1	0.083	0.99	5.811	0.341	0.007	4.955	1
54	102.48	0.069	0.99	5.643	0.338	0.005	5.043	1
55	183.87	0.043	0.977	7.975	0.221	0.002	5.082	1
56	220.67	0.14	0.976	7.02	0.324	0.019	4.471	1
57	131.68	0.104	0.984	6.113	0.313	0.011	5.009	1
58	136.34	0.041	0.988	6.106	0.27	0.002	5.082	1
59	160.87	0.096	0.988	6.802	0.346	0.009	4.844	1
60	33.961	0.064	0.996	3.012	0.414	0.004	5.072	1
61	36.023	0.094	0.996	2.904	0.446	0.009	4.967	1
62	153.42	0.217	0.989	6.256	0.425	0.047	4.346	1
63	174.82	0.219	0.989	7.197	0.424	0.048	4.435	1
64	237.89	0.029	0.977	7.259	0.249	8E-04	5.068	1
65	130.07	0.036	0.99	4.806	0.3	0.001	4.997	1
66	114.35	0.034	0.991	4.871	0.297	0.001	5.119	1
67	217.13	0.136	0.978	7.205	0.317	0.019	4.64	1
68	207.33	0.191	0.982	7.161	0.341	0.037	4.383	1
69	71.772	0.18	0.994	4.044	0.442	0.033	4.503	1
70	101.72	0.105	0.991	5.2	0.347	0.011	4.801	1
71	110.51	0.098	0.988	4.734	0.382	0.01	4.696	1
72	121.84	0.19	0.991	5.463	0.382	0.036	4.564	1
73	230.93	0.02	0.968	8.66	0.191	4E-04	5.153	1
74	193.51	0.131	0.976	6.713	0.389	0.017	4.812	1
75	99.72	0.146	0.987	5.186	0.37	0.021	4.716	1
76	394.12	0.034	0.955	9.128	0.243	0.001	5.06	1
77	366.57	0.16	0.968	8.604	0.318	0.026	4.657	1
78	149.02	0.118	0.988	5.756	0.352	0.014	4.449	1
79	309.27	0.052	0.965	9.511	0.271	0.003	4.999	1
80	210.25	0.044	0.974	7.634	0.301	0.002	5.197	1
81	258.01	0.05	0.973	8.964	0.259	0.003	5.134	1
82	309.27	0.052	0.965	9.511	0.271	0.003	4.999	1
83	156.04	0.091	0.973	6.249	0.336	0.008	4.816	1
84	74.142	0.037	0.972	4.95	0.314	0.001	4.763	1
85	390.13	0.034	0.924	11.67	0.152	0.001	5.102	1
86	204.7	0.054	0.965	8.971	0.184	0.003	4.932	1
87	598.89	0.051	0.933	13.46	0.172	0.003	5.16	1
88	153.23	0.023	0.982	8.074	0.195	5E-04	5.335	1
89	136.84	0.111	0.987	5.621	0.321	0.012	4.419	1
90	153.57	0.036	0.93	6.058	0.262	0.001	4.56	1
91	21.983	0.108	0.994	2.382	0.473	0.012	4.453	1

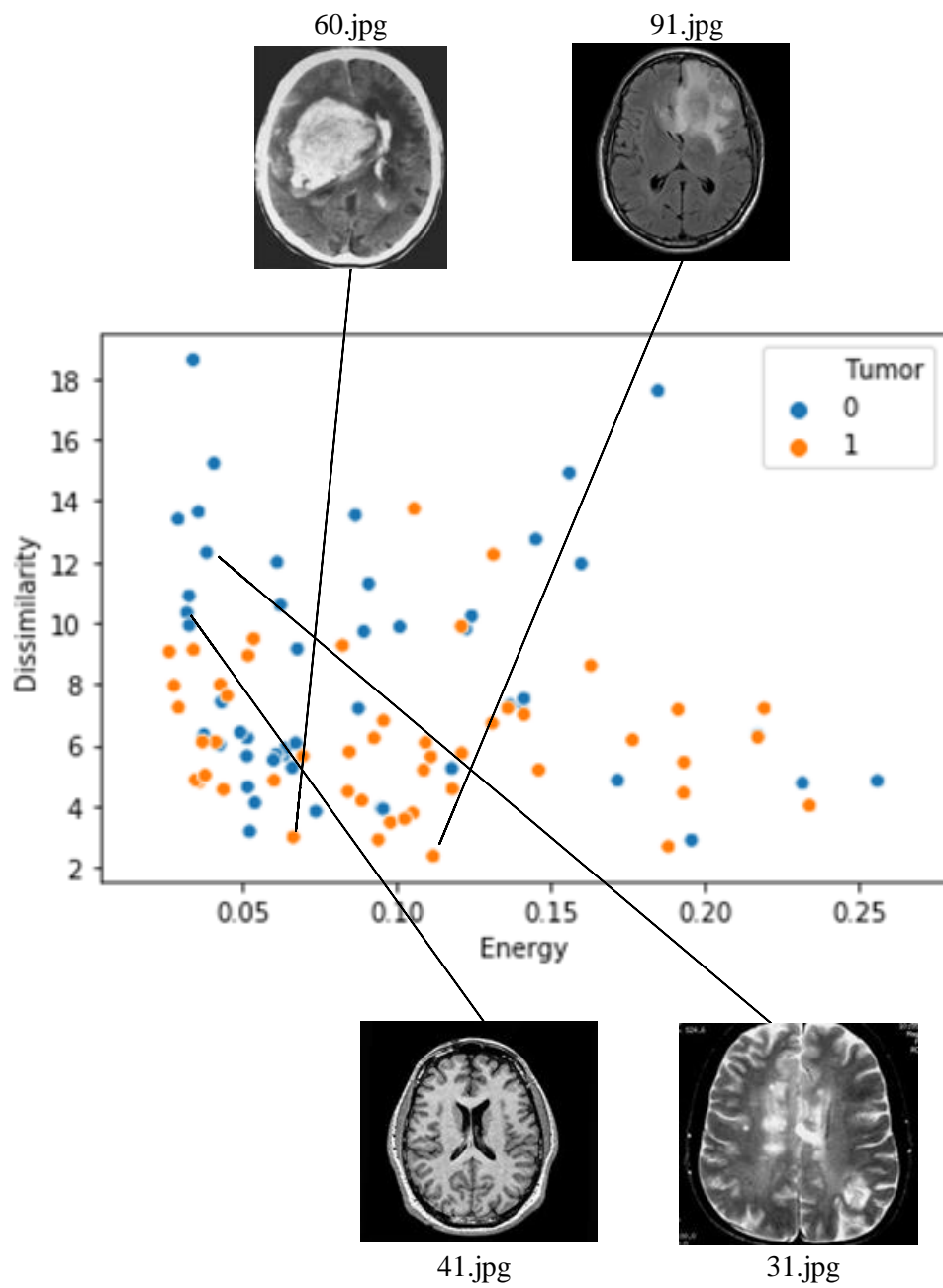
92	76.335	0.191	0.988	4.439	0.424	0.036	4.313	1
93	55.386	0.1	0.989	3.593	0.445	0.01	4.657	1
94	55.431	0.085	0.983	4.17	0.363	0.007	4.609	1
95	328.96	0.05	0.926	10.27	0.188	0.003	4.816	1
96	42.717	0.186	0.988	2.68	0.536	0.035	4.197	1
97	124.65	0.067	0.971	6.243	0.285	0.004	4.78	1
98	65.585	0.059	0.985	4.865	0.306	0.003	4.781	1
99	63.828	0.042	0.988	4.575	0.298	0.002	4.914	1



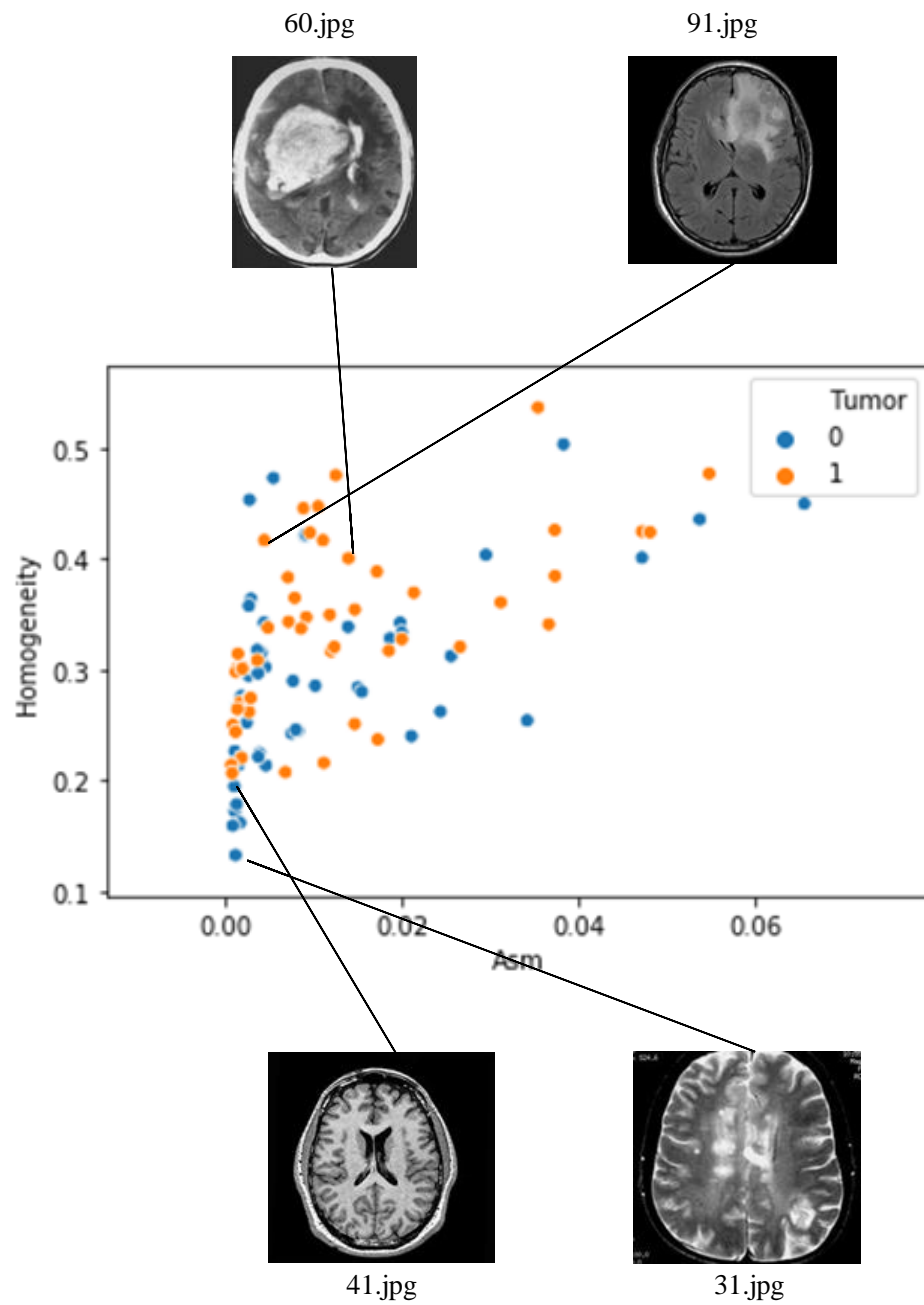
**Gambar 4.5** Hasil dari proses ekstraksi fitur tumor otak



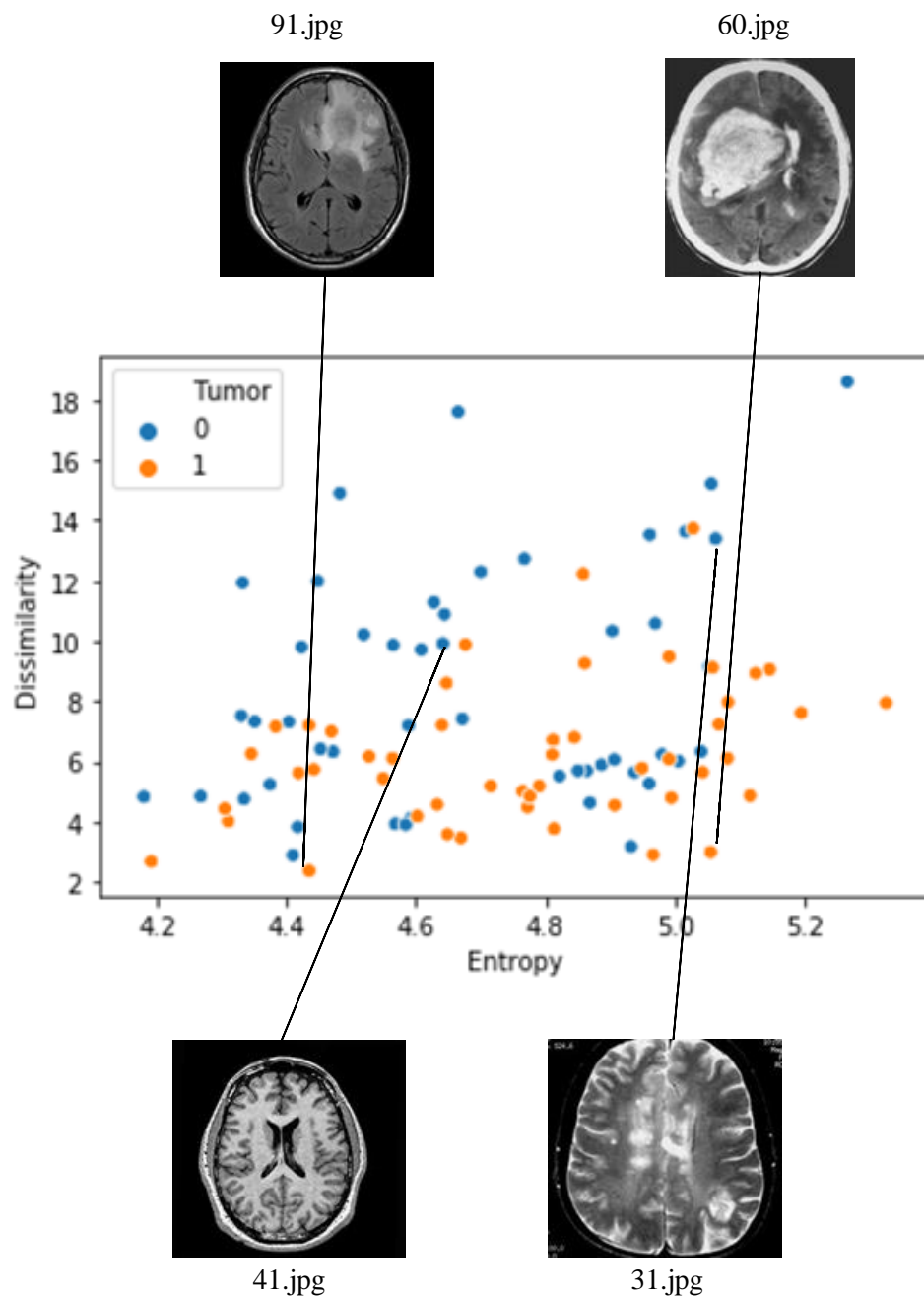
**Gambar 4. 6** Hasil dari ekstraksi fitur antara fitur Contrast dan fitur Correlation



**Gambar 4. 7** Hasil dari ekstraksi fitur antara fitur Energy dan fitur Dissimilarity



**Gambar 4. 8** Hasil dari ekstraksi fitur antara fitur ASM dan fitur Homogeneity



**Gambar 4. 9** Hasil dari ekstraksi fitur antara fitur Entropy dan fitur Dissimilarity



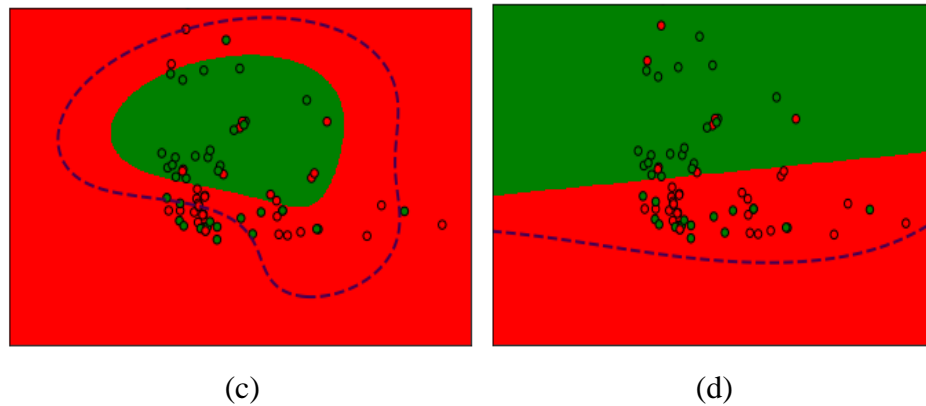
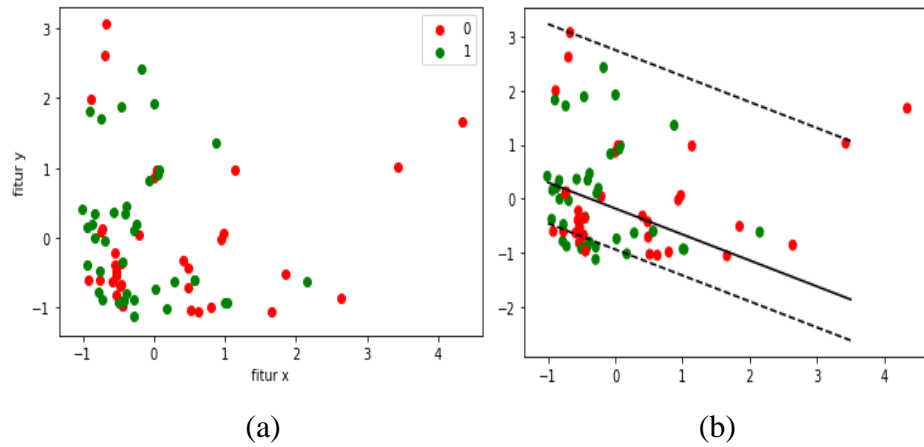
### 4.3 Klasifikasi

Dataset yang telah diambil fiturnya kemudian di simpan ke dalam file. Pada klasifikasi, perlu dilakukan plotting dataset agar dapat mengetahui persebaran data yang merupakan data tumor maupun bukan. Pelabelan tumor atau bukan dilakukan secara membaca pada citra dataset dan kemudian di kelompokkan berdasarkan kelas tumor atau bukan. Proses klasifikasi dengan SVM menggunakan SVC yaitu *Support Vector Classifier*. Classifier ini yang akan mencoba menemukan garis pemisah (*hyperplane*) yang memisahkan kelas tumor dengan kelas non tumor.

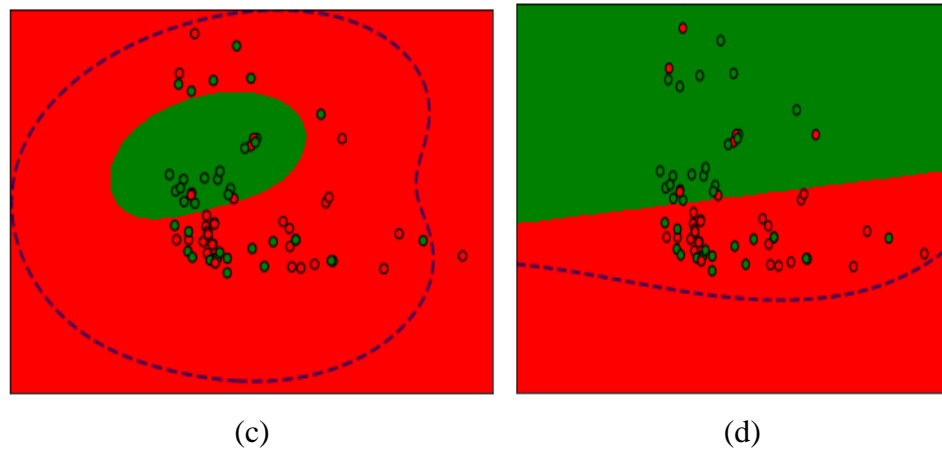
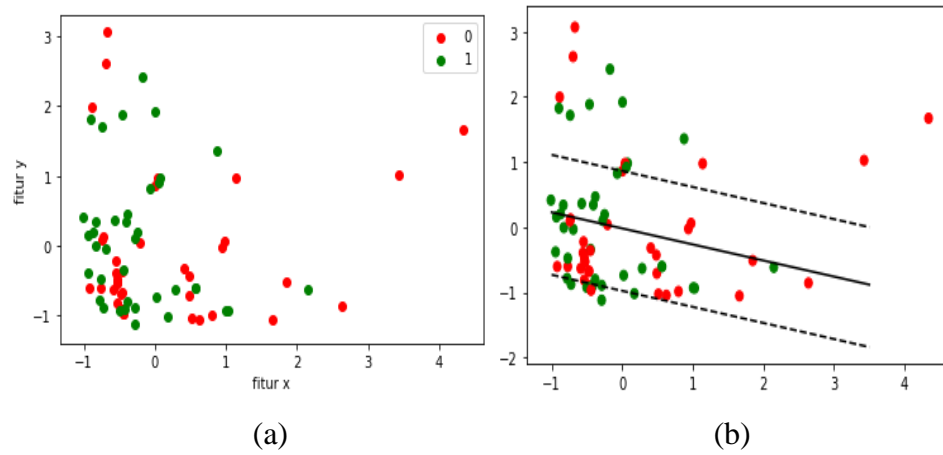
Setelah data dibagi menjadi 75 data training dan 25 data testing, data tersebut kemudian masuk ke dalam classifier SVM. Hal pertama yang akan dilakukan di dalam classifier adalah menentukan parameter yang akan digunakan. Untuk parameter  $C = 1.0$ ,  $\nu = 0.5$ ,  $\text{Epsilon} = 0.001$  serta  $\text{Gamma} = 0.5, 0.7$ . Setelah parameter ditentukan, kemudian memasukkan data *testing*, lalu mencari nilai *dot product* dari setiap data dengan memasukkan fungsi kernel ( $K$ ). Kemudian menghitung nilai  $\mathbf{w}$  yang merupakan bobot dari *dot product* data dengan  $\alpha$  terbesar pada masing-masing kelas, lalu mencari nilai bias ( $b$ ) dari Persamaan 3.5.

Setelah mendapatkan nilai  $\mathbf{w}, b, \alpha$ , maka dapat dilakukan proses pengujian. Nilai tersebut akan masuk ke dalam fungsi untuk mencari *hyperplane* antara kelas tumor dan kelas non tumor seperti yang ditunjukkan pada Gambar 4.10 dan Gambar 4.11. Pengujian ini menghasilkan parameter  $\text{Gamma}$  yang baik di antara 0.5 dan 0.7 serta tingkat akurasi yang paling baik.

Untuk proses klasifikasi dengan menggunakan SVM terbagi menjadi 2 proses, yaitu saat menggunakan SVM tipe 1 (C-SVM) dan menggunakan SVM tipe 2 (Nu-SVM). Yang membedakan dari kedua tipe tersebut adalah parameter  $C$  dan parameter  $\nu$ . Parameter  $C$  di set 1.0 sedangkan untuk parameter  $\nu$  di set 0.5. Karena parameter  $C$  berkisar dari 0 hingga tak terbatas, untuk parameter  $\nu$  berkisar antara 0 hingga 1. Parameter  $C$  dan parameter  $\nu$  merupakan parameter yang menentukan toleransi terhadap titik data yang tidak terklasifikasi. Setiap tipe SVM akan diuji dengan menggunakan 3 kernel berbeda, yaitu Linear, RBF, dan Polynomial. Parameter  $\text{gamma}$  yang digunakan akan bernilai 0.5 dan 0.7. Hasil dari proses klasifikasi ini dapat ditampilkan dalam *confusion matrix* yang ditunjukkan pada Tabel 4.2. Serta untuk menghitung nilai dari *specificity*, *precision*, *accuracy* dan *error* dapat dihitung melalui tabel *confusion matrix*. Sehingga hasil dari *specificity*, *precision*, *accuracy* dan *error* dapat dilihat pada Tabel 4.3.



**Gambar 4. 10** Hasil klasifikasi SVM untuk mencari *hyperplane* antara kelas tumor dan non tumor dengan menggunakan C-SVM, (a) Gambar sebelum diberikan *hyperplane*, (b) Gambar setelah diberi *hyperplane* dengan menggunakan kernel linear, (c) Gambar setelah diberi *hyperplane* dengan menggunakan kernel RBF, (d) Gambar setelah diberikan *hyperplane* dengan menggunakan kernel polynomial.



**Gambar 4. 11** Hasil klasifikasi SVM untuk mencari *hyperplane* antara kelas tumor dan non tumor dengan menggunakan Nu-SVM, (a) Gambar sebelum diberikan *hyperplane*, (b) Gambar setelah diberi *hyperplane* dengan menggunakan kernel linear, (c) Gambar setelah diberi *hyperplane* dengan menggunakan kernel RBF, (d) Gambar setelah diberikan *hyperplane* dengan menggunakan kernel polynomial.

**Tabel 4. 2** Hasil Confusion Matrix

Type	Kernel	TP	FP	TN	FN
C-SVM (SVC)	Linear	9	3	10	3
	RBF	10	2	9	4
	Polynomial	7	5	11	2
Nu-SVM (NuSVC)	Linear	9	3	7	6
	RBF	7	5	9	4
	Polynomial	7	5	11	2

**Tabel 4. 3** Hasil dari proses klasifikasi dengan SVM

Type	Kernel	Gamma	Spesifikasi	Presisi	Akurasi	Error
C-SVM (SVC)	Linear	0.5	0.76	0.75	0.76	0.24
	RBF	0.5	0.81	0.83	0.76	0.24
	Polynomial	0.7	0.68	0.58	0.72	0.28
Nu-SVM (NuSVC)	Linear	0.5	0.70	0.75	0.64	0.36
	RBF	0.7	0.64	0.58	0.64	0.36
	Polynomial	0.5	0.68	0.58	0.72	0.28

## BAB 5 HASIL DAN PEMBAHASAN

### 5.1 *Pre-processing*

Pada tahap pengujian ini, dataset yang telah dibaca oleh *library* OpenCV pada Python akan masuk ke tahap *pre-processing*. Dataset yang digunakan masih memiliki tipe warna RGB, sehingga untuk memudahkan proses selanjutnya dataset ini harus diubah ke dalam bentuk *grayscale* terlebih dahulu. Dataset yang telah diubah ke *grayscale* kemudian akan diproses ke dalam filter, *thresholding*, kontur dan proses *automatic cropping*.

Dengan menggunakan proses *pre-processing* hasil akurasi yang didapatkan sangat maksimal dibandingkan dengan tidak menggunakan *pre-processing*. Hal ini dikarenakan terdapat *noise* pada citra yang dapat mengakibatkan hasil akurasi yang tidak optimal. Oleh karena itu, tahap *pre-processing* sangat dibutuhkan dalam proses klasifikasi untuk menghilangkan *noise* yang terdapat pada citra agar hasil akurasi dari proses klasifikasi sangat optimal. Hasil dari perbandingan akurasi pada citra yang menggunakan *pre-processing* dan tidak menggunakan *pre-processing* dapat dilihat pada Tabel 5.1.

**Tabel 5. 1** Perbandingan hasil akurasi dari SVM dengan menggunakan pre-processing dan tidak menggunakan pre-processing

Tipe	Kernel	Akurasi tanpa Pre-processing	Akurasi dengan Pre-processing
C-SVM (SVC)	Linear	0.64	0.76
	RBF	0.68	0.76
	Polynomial	0.72	0.72
Nu-SVM (NuSVC)	Linear	0.64	0.64
	RBF	0.68	0.64
	Polynomial	0.72	0.72

### **5.1.1 Gaussian Blur**

Gaussian *Smoothing (Blur)* memberikan efek *blur* pada citra. Pada Gambar 4.1 terlihat citra yang telah diberikan efek *blurring*. Secara visual dapat dilihat bahwa efek *blurring* memberikan efek *blur* pada citra *grayscale*, tetapi efek tersebut tidak terlihat perbedaannya dengan citra sebelum diberikan efek *blurring*. Dikarenakan citra tersebut hanya memiliki sedikit *noise*, maka *noise* tersebut cukup dihilangkan dengan Gaussian *Blur* saja.

### **5.1.2 Thresholding**

*Thresholding* merupakan proses yang mengubah citra *grayscale* menjadi citra *binary* atau hitam putih. Pada Gambar 4.2 terlihat citra yang telah diubah ke dalam *thresholding*. Citra *thresholding* terbagi menjadi 2, yaitu citra berwarna putih yang mewakili sebagai objek dari citranya dan citra berwarna hitam yang merupakan *background* dari citra tersebut. Setelah *thresholding* dilakukan, kemudian lakukan serangkaian proses menebalkan dan menipiskan untuk menghilangkan daerah *noise* yang kecil. Kegunaan dari *thresholding* sendiri adalah untuk segmentasi. Sehingga *thresholding* ini digunakan untuk mengelompokan wilayah citra otak dengan *backgroundnya*.

### **5.1.3 Contour**

Setelah citra di *thresholding*, maka selanjutnya mencari kontur pada citra tersebut dan ambil bagian terbesarnya. Kontur disini digunakan untuk mendeteksi tepi pada citra. Dari citra yang telah di *thresholding*, digunakan *library* OpenCV dengan mode pengambilan kontur secara eksternal. Sehingga bagian tepi pada citra mempunyai gambaran kontur seperti yang terlihat pada Gambar 4.3. Ini berfungsi untuk menentukan *extreme point* pada citra kontur.

### **5.1.4 Automatic Cropping**

Citra yang telah diberi kontur sebelumnya, akan dicari titik-titik ekstrimnya agar mudah melakukan proses pemotongan (*cropping*) pada citra. Daerah kontur tersebut akan diberikan titik-titik ekstrim yang dimana posisinya terletak pada titik kanan, titik kiri, titik atas dan titik bawah seperti pada Gambar 4.4. Dengan memberikan titik-titik ekstrim pada citra otak, maka akan semakin mendekati daerah pada citra otak tersebut. *Automatic cropping* digunakan untuk fokus pada objek yang akan diolah, dengan memotong bagian yang telah diberi titik ekstrim seperti yang terlihat pada Gambar 4.5. Fokus pada penelitian ini adalah *input* pada citra otak.

## 5.2 Feature Extraction

Setelah melalui tahap *pre-processing*, selanjutnya akan diambil beberapa informasinya (*feature*) yang terkandung dalam citra tersebut dengan menggunakan *feature extraction* yang berupa *texture feature*. *Texture feature* mengacu pada karakteristik permukaan atau penampilan suatu objek yang diberikan berdasarkan ukuran, bentuk, kerapatan, susunan, dan proporsi bagian lainnya. Alasan memilih tekstur adalah untuk mengkarakterisasi berbagai jenis daerah menghadirkan tekstur yang berada pada jaringan organ yang berbeda dalam gambar MR dan dengan deskripsi dari tekstur tersebut diharapkan dapat membedakan antara berbagai jenis daerah. *Texture feature* disini dapat diekstrak dengan berbagai macam metode. Salah satu metode yang dipakai dalam penelitian ini adalah *Gray Level Co-occurrence Matrix* (GLCM).

Hasil dari proses *feature extraction* ini dapat dilihat pada Tabel 4.2. Fitur-fitur ini akan digunakan untuk learning *algorithm* pada proses klasifikasi. Pada Gambar 4.5, terlihat perbandingan antara 7 fitur yang telah terekstraksi. Pada fitur *contrast* antara data tumor dan data non tumor yang terlihat pada perbandingan nilai fitur *contrast* tersebut, terlihat bahwa distribusi data tumor memiliki kemunculan data yang tinggi dibandingkan dengan data non tumor. Untuk perbandingan fitur *energy* antara data tumor dan data non tumor terlihat bahwa distribusi data yang memiliki kemunculan data paling besar merupakan data non tumor, sehingga banyaknya data yang muncul pada non tumor sangat tinggi dibandingkan dengan data tumor. Kemudian untuk fitur *correlation*, terlihat bahwa distribusi data tumor memiliki kemunculan data yang tinggi dibandingkan dengan data non tumor. Begitu pula untuk fitur *dissimilarity* yang data tumornya memiliki distribusi kemunculan data yang tinggi dibandingkan dengan data non tumor. Pada fitur *homogeneity*, kedua data tersebut mempunyai ketinggian puncak yang sama, tetapi data tumor memiliki nilai distribusi data yang sedikit lebih tinggi dibandingkan dengan data non tumor. Fitur *Angular Second Moment* (ASM), juga memiliki tinggi yang sama diantara kedua data tersebut. Tetapi jika dilihat lebih dekat pada fitur ASM, distribusi data non tumor memiliki kemunculan data sedikit lebih tinggi dibandingkan dengan data tumor. Dan untuk fitur *entropy*, kedua data tersebut juga memiliki nilai distribusi data yang tingginya hampir sama. Untuk data non tumor memiliki distribusi kemunculan data yang tinggi dibandingkan dengan data tumor.

Dari 100 citra yang telah diambil fiturnya, dilampirkan 2 citra dari masing-masing kelas, tumor dan non tumor untuk melihat perbandingan di setiap fiturnya. Untuk citra non tumor, menggunakan citra 41.jpg dan 31.jpg, sedangkan untuk citra tumor, menggunakan citra 60.jpg dan 91.jpg. Pada Gambar 4.6, menunjukkan perbandingan antara fitur *contrast* dengan fitur *correlation*. Hasilnya citra tumor 60.jpg dan 91.jpg mempunyai nilai *contrast* yang kecil tetapi mempunyai nilai *correlation*

yang tinggi. Sedangkan untuk citra non tumor 31.jpg dan 41.jpg mempunyai nilai *contrast* yang lebih baik dari citra tumor tetapi nilai *correlation* yang dihasilkan lebih rendah. Artinya pada citra tumor 60.jpg dan 91.jpg, mempunyai perbedaan skala *grayscale* yang rendah dibandingkan dengan citra non tumor 31.jpg dan 41.jpg. Tetapi untuk korelasi antara derajat citra *grayscale* justru lebih tinggi citra tumor 60.jpg dan 91.jpg daripada citra non tumor 31.jpg dan 41.jpg.

Selanjutnya fitur *energy* yang akan dibandingkan dengan fitur *dissimilarity* yang terlihat pada Gambar 4.7. Citra tumor 60.jpg dan 91.jpg mempunyai *energy* yang besar dibandingkan dengan citra non tumor 31.jpg dan 41.jpg. Tetapi citra non tumor 31.jpg dan 41.jpg memiliki nilai *dissimilarity* paling tinggi dibandingkan dengan citra tumor 60.jpg dan 91.jpg. Sehingga yang mempunyai keseragaman citra yang paling besar dimiliki oleh citra tumor 60.jpg dan 91.jpg. Sedangkan citra non tumor 31.jpg dan 41.jpg memiliki nilai rata-rata dari perbedaan level citra *grayscale* yang besar.

Kemudian fitur ASM yang akan dibandingkan dengan fitur *homogeneity* yang dapat dilihat pada Gambar 4.8. Citra tumor 60.jpg dan 91.jpg memiliki nilai ASM sedikit lebih besar daripada citra non tumor 31.jpg dan 41.jpg. Begitu juga untuk nilai *homogeneity* dari citra tumor 60.jpg dan 91.jpg lebih besar daripada citra non tumor 31.jpg dan 41.jpg. Artinya citra tumor 60.jpg dan 91.jpg mempunyai ukuran homogenitas dan mempunyai *homogeneity* yang paling besar dibandingkan dengan citra non tumor 31.jpg dan 41.jpg.

Yang terakhir perbandingan dari fitur *entropy* dengan fitur *dissimilarity* yang ditampilkan pada Gambar 4.9. Nilai *entropy* dari citra tumor 60.jpg dan 91.jpg mempunyai nilai yang tidak begitu jauh dari citra non tumor 31.jpg dan 41.jpg. Karena kompleksitas tekstur pada citra tumor mempunyai kemiripan nilai yang sama dengan citra non tumor. Sehingga hasil dari ekstraksi fitur ini dapat dilakukan proses pengujian akurasi terhadap kelas tumor maupun non tumor.

### 5.3 Klasifikasi

Proses klasifikasi ini menggunakan metode *Support Vector Machine* (SVM). Hasil dataset dari proses fitur ekstraksi kemudian masuk ke dalam proses klasifikasi. Hasil perhitungan spesifikasi, presisi dan akurasi menggunakan perhitungan dari *confusion matrix* :

1. Specificity =  $TN / (TN + FP)$
2. Precision =  $TP / (TP + FP)$
3. Accuracy =  $(TP + TN) / (TP + FP + TN + FN)$
4. Error =  $(FP + FN) / (TP + FP + TN + FN)$

Di dalam proses klasifikasi, digunakan 2 tipe SVM untuk mengklasifikasi data tersebut. Klasifikasi pertama menggunakan tipe C-SVM. Kernel yang digunakan yaitu kernal



linier, RBF dan polynomial. Hasil yang didapatkan dengan menggunakan kernel linear adalah dengan gamma 0.5 menghasilkan tingkat akurasi 0.76, spesifikasi sebesar 0.76, presisinya 0.75 dan *error* sebesar 0.24. Sedangkan untuk kernel RBF dengan gamma 0.5 menghasilkan tingkat akurasi 0.76, spesifikasi sebesar 0.81, presisinya 0.83 serta *error* sebesar 0.24. Dan untuk kernel polynomial dengan gamma 0.7 menghasilkan tingkat akurasi 0.72, spesifikasi sebesar 0.68, presisinya 0.58 dan *error* sebesar 0.28.

Hasil akurasi dari tipe C-SVM dapat dibuktikan juga pada Gambar 4.10, dimana kelas non tumor ditandai dengan warna merah, sedangkan kelas tumor ditandai dengan warna hijau. Pengujian pertama menggunakan kernel linear, terdapat *hyperplane* yang memisahkan kedua kelas tersebut. Jarak *hyperplane* dengan margin pada kelas non tumor sangat jauh, tetapi garis margin tersebut mendapatkan *support vector* dari kelas non tumor sebanyak 2 data dan untuk margin dari kelas tumor mendapatkan *support vector* sebanyak 1 buah seperti yang terlihat pada Gambar 4.10 (b). Selanjutnya menggunakan kernel RBF, yang ditunjukkan pada Gambar 4.10 (c), menjelaskan bahwa masih terdapat *misclassified* antara data kelas tumor dan non tumor, dimana sebagian dari data non tumor berada di wilayah tumor dan data tumor yang berada di wilayah non tumor. Untuk data *support vector* yang terkena margin terdapat 2 buah dari kelas tumor dan 2 buah dari kelas non tumor. Untuk kernel yang terakhir, yaitu kernel polynomial yang dapat dilihat pada Gambar 4.10 (d), bahwa lebih banyak data yang *misclassified* dibandingkan dengan kernel RBF. Tetapi untuk data *support vector* yang terkena margin tidak ada. Hasil tersebut bergantung dari parameter C dan Gamma yang telah ditentukan sebelumnya.

Untuk klasifikasi kedua menggunakan Nu-SVM. Kernel yang digunakan juga sama seperti tipe yang pertama. Hasil yang didapatkan dengan menggunakan kernel linier dengan gamma 0.5 menghasilkan tingkat akurasi 0.64, spesifikasi sebesar 0.70, presisinya 0.75 dan *error* sebesar 0.36. Sedangkan untuk kernel RBF dengan gamma 0.7 menghasilkan hasil akurasi 0.64, spesifikasi sebesar 0.64, presisinya 0.58 serta *error* sebesar 0.36. Dan untuk kernel polynomial dengan gamma 0.5 menghasilkan akurasi 0.72, spesifikasi sebesar 0.68, presisinya 0.58 dan *error* sebesar 0.28.

Sama seperti pada tipe C-SVM, tipe Nu-SVM juga dapat dibuktikan hasil akurasinya yang terlihat pada Gambar 4.11 dengan data non tumor yang ditandai dengan warna merah dan data tumor ditandai dengan warna hijau. Pada kernel linear, jarak *hyperplane* yang memisahkan kedua data tersebut dengan margin non tumor agak dekat. Margin tersebut mendapatkan *support vector* dari non tumor sebanyak 1 data. Sedangkan margin tumor mendapatkan *support vector* sebanyak 3 data seperti yang ditunjukkan pada Gambar 4.11(b). Sedangkan untuk kernel RBF, yang ditunjukkan pada Gambar 4.11(c), terdapat banyak data tumor yang berada di wilayah non tumor. Untuk data non tumor justru mempunyai sedikit data yang berada di wilayah tumor.

Tetapi pada kernel ini tidak terdapat data *support vector* yang terkena margin. Dan yang terakhir pengujian dengan menggunakan kernel polynomial yang terlihat pada Gambar 4.11(d). Didapatkan hasil *misclassified* data yang sedikit jumlahnya dari data tumor maupun data non tumor. Sama seperti C-SVM, tipe kernel ini tidak memiliki data *support vector* yang terkena margin. Hasil tersebut juga dipengaruhi oleh parameter Nu dan Gamma.

Hasil yang didapatkan pada kedua tipe tersebut, tipe C-SVM memiliki nilai gamma yang semakin besar nilainya maka hasil akurasi akan semakin kecil. Sedangkan untuk spesifikasi dan presisi pada gamma 0.5 memiliki nilai yang cukup baik, sedangkan untuk gamma 0.7 memiliki nilai spesifikasi dan presisi yang kurang baik. Sehingga hasil akurasi terbaik pada tipe C-SVM menghasilkan akurasi sebesar 0.76 dengan menggunakan kernel linear dan RBF dengan gamma 0.5. Dan untuk nilai spesifikasi dan presisi terbaik berada pada kernel RBF dengan nilai spesifikasi 0.81 dan presisi 0.83. Sedangkan untuk tipe Nu-SVM yang memiliki nilai gamma terbesar, justru memiliki tingkat akurasi 0.64. Dan nilai gamma 0.5 memiliki tingkat akurasi sebesar 0.64 dan 0.72. Sedangkan pada gamma 0.5 memiliki nilai spesifikasi dan presisi yang cukup baik dibandingkan dengan gamma 0.7. Sehingga hasil akurasi terbaik pada tipe Nu-SVM menghasilkan akurasi sebesar 0.72 dengan menggunakan kernel polynomial dan gamma 0.5. Sedangkan untuk hasil spesifikasi dan presisi terbaik diperoleh dari kernel linear dengan spesifikasi sebesar 0.70 dan presisi 0.75. Ketika nilai gamma dibesarkan dari nilai yang ditentukan, wilayah antara data tumor dan non tumor tidak terlihat, sedangkan apabila nilai gamma dikecilkan dari nilai yang ditentukan, batas wilayah dari data tumor dan non tumor terlihat, tetapi hasil akurasi ketika nilai gamma dibesarkan dan dikecilkan sangat tidak optimal dibandingkan dengan nilai gamma yang sudah ditentukan sebelumnya. Sehingga dari hasil kedua tipe *classifier* tersebut, dengan nilai gamma 0.5 dapat memiliki tingkat akurasi yang cukup baik.

Pada penelitian sebelumnya oleh S. Manjunath *et al* [16], hasil akurasi dari klasifikasi tumor otak yang diperoleh sangat baik, dengan perolehan akurasi sebesar 87. Karena di dalam proses *pre-processing*, digunakan proses segmentasi menggunakan algoritma *k-means* dan algoritma *fuzzy*. Algoritma *fuzzy c-means* yang telah diimprovisasi, akan diimplementasikan kepada *membership* yang telah diperbarui untuk mensegmentasikan sifat berdasarkan citra yang sudah tersegmentasikan dengan menggunakan algoritma *k-means*. Sehingga proses dari *pre-processing* sangat mempengaruhi hasil akurasi dari proses klasifikasi. Dengan menggunakan SVM *classifier* untuk mengklasifikasi data yang bersifat non-linear dengan menggunakan fungsi kernel seperti RBF dapat menghasilkan hasil yang bagus ketika dataset yang digunakan lebih kecil.

## BAB 6 KESIMPULAN DAN SARAN

### 6.1 Kesimpulan

Pada tugas akhir ini, dilakukan proses klasifikasi pada citra tumor otak. Tahapan pertama yang dilakukan adalah tahap *pre-processing*. Tujuan dari proses *automatic cropping* adalah untuk mengubah ukuran pada citra agar fokus pada objek yang akan diproses. Dengan menggunakan *pre-processing*, hasil akurasi yang didapatkan pada proses klasifikasi sangat optimal. Setelah tahapan *pre-processing* dilakukan, maka tahapan selanjutnya adalah ekstraksi fitur. Ekstraksi fitur yang digunakan yaitu GLCM. Karena pada citra otak, yang akan diambil informasinya berupa tekstur dari citranya. Informasi yang diambil ada 7, yaitu : *contrast, energy, correlation, dissimilarity, homogeneity, angular second moment (ASM)*, serta *entropy*. Dari 7 fitur tersebut, akan disimpan ke dalam file .csv. Langkah terakhir yaitu proses klasifikasi. Data yang disimpan dalam .csv tersebut akan dibagi menjadi 75 data *training set* dan 25 data *testing set*. Kemudian data akan diuji dengan menggunakan SVM classifier. Dimana ada 2 tipe SVM yang digunakan, yaitu C-SVM dan Nu-SVM. Dari 2 tipe klasifikasi tersebut, didapatkan hasil akurasi di tiap tipenya. Pada tipe C-SVM, dengan menggunakan kernel linear dan RBF dengan gamma 0.5 mendapatkan tingkat akurasi sebesar 0.76. Sedangkan untuk tipe Nu-SVM, dengan menggunakan kernel polynomial dengan gamma 0.5 mendapatkan tingkat akurasi sebesar 0.72. Dari hasil tersebut, nilai gamma 0.5 dapat memiliki tingkat akurasi yang baik.

Dibandingkan dengan hasil akurasi dari penelitian sebelumnya oleh S. Manjunath *et al* [16], hasil akurasinya masih kurang cukup baik. Dikarenakan peneliti tersebut menggunakan metode *pre-processing* dengan mensegmentasi bagian citra menggunakan algoritma *k-means* dan algoritma *fuzzy*. Penulis juga sudah mencoba menguji hasil klasifikasi dengan tidak menggunakan *pre-processing* dan hasil akurasinya sangat tidak optimal. Sehingga proses *pre-processing* sangat mempengaruhi hasil akurasi dari proses klasifikasi. Dengan menggunakan SVM *classifier* untuk mengklasifikasi data yang bersifat non-linear dengan menggunakan fungsi kernel seperti RBF dapat menghasilkan hasil yang bagus ketika dataset yang digunakan lebih kecil.

### 6.2 Saran

Dikarenakan hasil akurasinya masih berada di jangkauan cukup baik, maka bisa ditambahkan metode yang lain pada proses ekstraksi fitur. Serta pada proses pembelajaran bisa dilakukan kombinasi antara SVM dengan model yang lain seperti algoritma *fuzzy* atau CNN sehingga hasil yang diharapkan bisa didapatkan. Diharapkan

untuk penelitian selanjutnya dapat mencoba untuk mengklasifikasi tumor otak dengan menggunakan data *Diffusion Tensor Imaging* (DTI) dan metode untuk fiturnya bisa dicoba menggunakan *shape texture*. Kemudian data tersebut bisa dilakukan proses pembelajarannya dengan menggunakan SVM maupun CNN atau *machine learning* yang lain.

## DAFTAR PUSTAKA

- [1] B. G. Smeltzer, S. C., Bare, *Buku Ajar Keperawatan Medikal-Bedah Brunner &Suddarth*, Vol. 2. Jakarta, 2001.
- [2] V. S. T Sutoyo, E Mulyanto, “Teori Pengolahan Citra Digital,” 2009.
- [3] C.N. Santi, “Mengubah Citra Berwarna Menjadi Grayscale dan Citra biner,” *Tekno. Inf. Din.*, vol. 16, pp. 14–19.
- [4] M. Y. dan F. B. G. F. Bozkurt, “Effective Gaussian Blurring Process on Graphics Processing Unit with CUDA,” *Int. J. Mach. Learn. Comput.*, vol. 5, pp. 57–61.
- [5] H. L. S. dan Y. S. A. Wedianto, “Analisa Perbandingan Metode Filter Gaussian, Mean dan Median Terhadap Reduksi Noise,” *J. Media Infotama*, vol. 12, pp. 21–30.
- [6] Munir. Rinaldi, “Pengolahan Citra Digital dengan Pendekatan Algoritmik.,” *Penerbit Inform.*, 2004.
- [7] Rashmi, M. Kumar dan R. Saxena, “Algorithm and Technique on Various Edge Detection: A Survey. Signal & Image Processing,” *An Int. J.*, vol. 4, pp. 65–75.
- [8] A. S. Nugroho, A. B. Witarto, and D. Handoko, “Support Vector Machine,” 2003.
- [9] B. Santosa, “Data Mining Teknik Pemanfaatan Data untuk Keperluan Bisnis,” 2007.
- [10] R. Manjula, “Tumor Identification Using Binary Thresholding in MRI Brain Images \*1,2,” vol. 2, no. 4, 2013.
- [11] A. H. Ali, K. A. Khalaph, and I. S. Nema, “Segmentation of brain tumour using Enhanced Thresholding Algorithm and Calculatethe area of the tumour,” vol. 4, no. 1, pp. 58–62, 2014.
- [12] T. Chithambaram and K. Perumal, “Edge Detection Algorithms Using Brain Tumor Detection and Segmentation Using Artificial Neural Network Techniques,” vol. 1, no. 3, pp. 135–140, 2016.
- [13] M. P. K, “Brain Tumor Types and Grades Classification based on Statistical Feature Set using Support Vector Machine,” 2015.
- [14] K. Vaibhav and A. K. Yadav, “Texture feature extraction and classification using radial basis function for diagnosis of brain tumour CLASSIFICATION USING RADIAL BASIS FUNCTION,” no. September 2017, 2016.
- [15] B. Ghotekar, “Brain Tumor Detection and Classification using SVM,” pp. 180–182, 2016.
- [16] S. Manjunath, “Classification of Brain Tumor Types in MRI Using Support Vector Machine,” vol. 10, pp. 277–281, 2018.

## **BIODATA PENULIS**



Penulis dilahirkan di Kota Surabaya pada tanggal 6 Februari 1997. Penulis merupakan anak kedua dari tiga bersaudara. Penulis menempuh pendidikan sekolah dasar di SDN Manukan Kulon III/540 Surabaya dan meneruskan ke jenjang sekolah menengah pertama di MTs PPMI Assalaam Solo dan melanjutkan ke sekolah menengah atas SMAN 15 Surabaya. Kemudian penulis meneruskan ke jenjang pendidikan yang lebih tinggi ke Departemen Teknik Biomedik Institut Teknologi Sepuluh Nopember, dan aktif sebagai asisten Laboratorium Instrumentasi dan Pengolahan Sinyal Biomedika.

*E-mail* : [ainanishabrina@gmail.com](mailto:ainanishabrina@gmail.com)