

# Klasifikasi Tumor Otak pada Citra Magnetic Resonance Image dengan Menggunakan Metode Support Vector Machine

Ainani Shabrina Febrianti, Tri Arief Sardjono, dan Atar Fuady Babgei  
Departemen Teknik Biomedik, Institut Teknologi Sepuluh Nopember (ITS)  
*e-mail*: sardjono@bme.its.ac.id

**Abstrak**—Di Indonesia, pasien yang terdiagnosis tumor otak semakin meningkat setiap tahunnya. Bukan hanya orang dewasa saja, tetapi anak-anak juga dapat terserang penyakit tumor otak. Untuk membedakan citra gambar *Magnetic Resonance Image* (MRI) yang terdeteksi tumor otak dengan yang tidak terdeteksi tumor otak, perlu dilakukan proses klasifikasi dengan menggunakan metode *Support Vector Machine* (SVM). Dalam penelitian ini, proses klasifikasi dilakukan dalam tiga langkah, *Pre-processing* yang mengubah ukuran citra, ekstraksi fitur yang mengambil informasi (fitur) secara teksturnya, kemudian diikuti dengan melatih data dan dilakukan pengujian pada SVM. Hasil dari penelitian ini adalah untuk mengklasifikasikan citra MRI 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 dengan menggunakan kernel Linear dan RBF pada tipe C-SVM.

**Kata Kunci**—Citra MRI, Ekstraksi Fitur, *Pre-processing*, *Support Vector Machine* (SVM), Tumor Otak.

## I. PENDAHULUAN

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 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 penelitian 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.

## II. TINJAUAN PUSTAKA

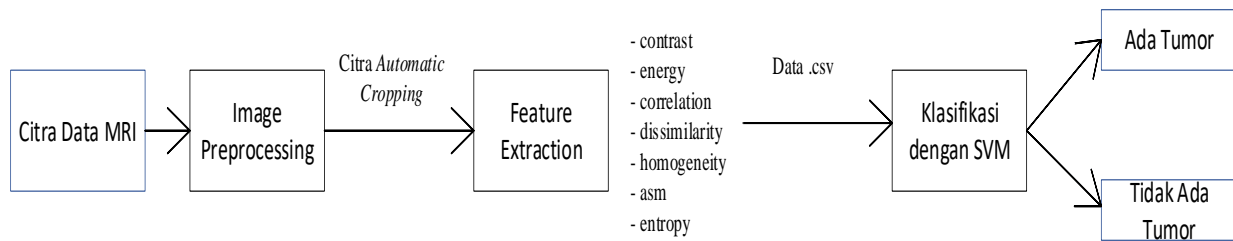
### A. *Pre-Processing*

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. *Input* dari pengolahan citra adalah citra, sedangkan *Output*-nya adalah citra hasil pengolahan. Untuk proses *pre-processing* terdiri dari proses *grayscale*, Gaussian filter, *thresholding*, dilasi, erosi dan kontur.

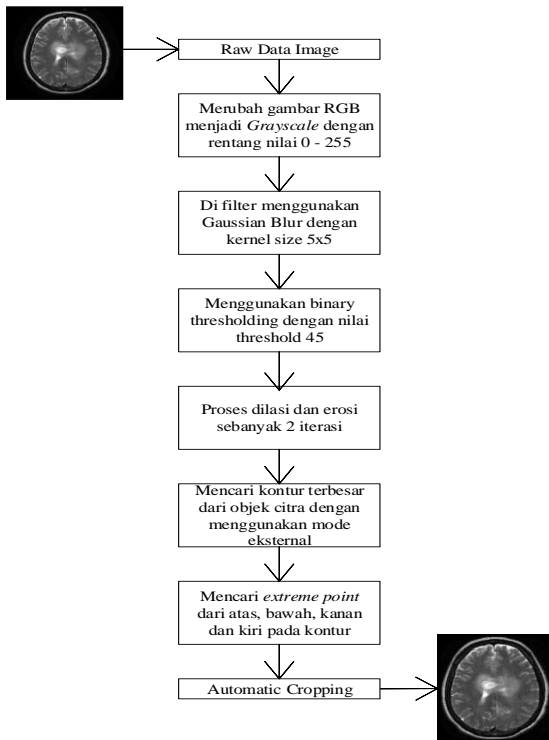
### B. Ekstraksi Fitur

*Feature Extraction* (Ekstraksi fitur) menjelaskan informasi yang terkandung dalam suatu pola sehingga tugas mengklasifikasikan pola menjadi mudah. Fitur ekstraksi yang digunakan yaitu tekstur fitur dengan menggunakan *Gray Level Co-occurrence Matrix* (GLCM). 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  $0^\circ$ ,  $45^\circ$ ,  $90^\circ$ ,  $135^\circ$  ([1],[2]). 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 keabuaannya.

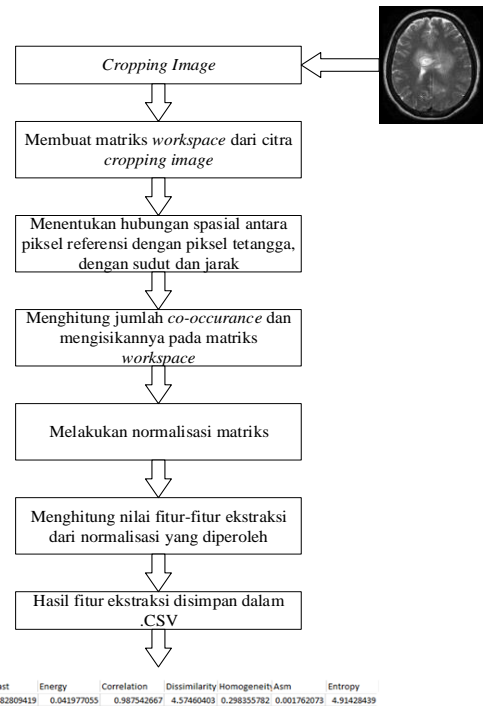
Terdapat 14 fitur tekstural yang digunakan untuk mengklasifikasikan citra. Namun, terdapat penelitian berikutnya yang menginvestigasikan karakteristik tekstural dengan menggunakan GLCM dan mengatakan bahwa terdapat 7 parameter yang paling relevan untuk mengklasifikasikan citra berdasarkan hasil matriks *Co-occurrence* [3].



Gambar 1. Diagram Blok Metode Penelitian



Gambar 2. Diagram Blok Image Pre-processing



Gambar 3. Diagram Blok Ekstraksi Fitur

### C. 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* [4]. Konsep SVM dapat dijelaskan secara sederhana sebagai usaha mencari *hyperplane* terbaik yang berfungsi sebagai pemisah dua buah kelas pada *input space*. *Pattern* yang merupakan anggota dari dua buah kelas: +1 dan -1 dan berbagi alternatif garis pemisah (*discrimination boundaries*). *Margin* adalah jarak antara *hyperplane* tersebut dengan *pattern* terdekat dari masing-masing kelas. *Pattern* yang paling dekat ini disebut sebagai *support vector*. Usaha untuk mencari lokasi *hyperplane* ini merupakan inti dari proses pembelajaran pada SVM [5].

### III. URAIAN PENELITIAN

Pada penelitian ini, penulis menggunakan dua referensi dari penelitian sebelumnya untuk membantu proses klasifikasi pada citra tumor dan non tumor. Mohana Priya *et*

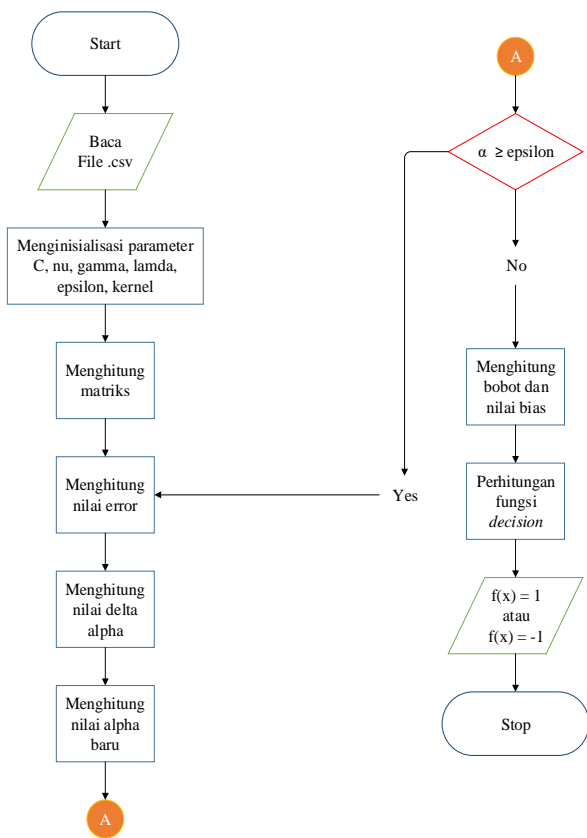
*al* mengklasifikasikan citra tumor otak berdasarkan tipe dan ukuran dari tumor otak dengan menggunakan SVM *classifier* [6]. Dataset yang digunakan berjumlah 421 citra dengan pembagian 208 citra untuk mengklasifikasikan berdasarkan tipe tumor otak dan 413 untuk mengklasifikasikan tumor otak berdasarkan ukuran tumor. Fitur ekstraksi yang digunakan yaitu tekstur fitur. Hasil akurasi yang didapatkan 85% untuk klasifikasi berdasarkan tipe tumor otak dan 78.26% untuk klasifikasi berdasarkan ukuran tumor otak.

S. Manjunath *et al*, menggunakan 80 dataset citra MRI untuk mengklasifikasikan tumor berdasarkan tipenya [7]. Ekstraksi fitur yang digunakan yaitu tekstur fitur dengan jumlah 12 fitur yang telah terekstraksi. Klasifikasi yang digunakan menggunakan SVM *classifier* dengan hasil akurasi berupa 68.51% untuk fitur orde pertama, 87% untuk fitur orde kedua dan 86.6% untuk gabungan fitur orde pertama dan orde kedua.

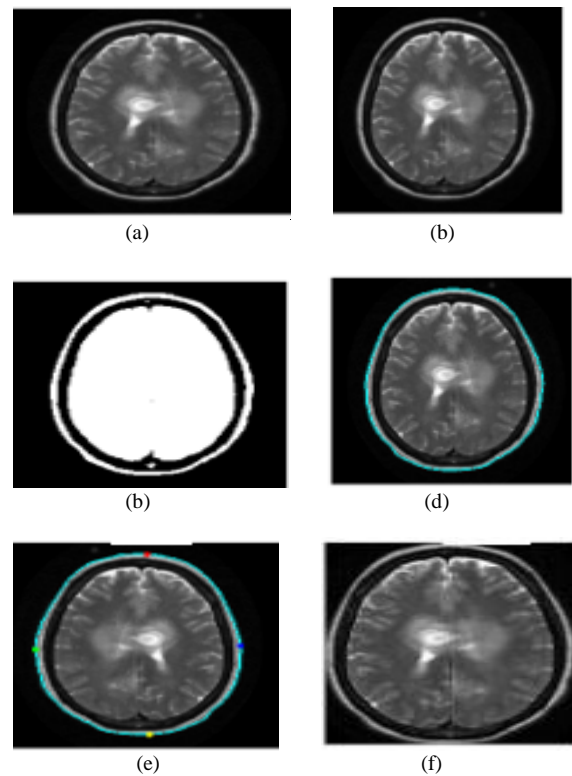
## IV. METODOLOGI

### A. Desain Metode Penelitian

Secara umum, alur metode yang akan dibuat dijelaskan dalam diagram blok pada Gambar 1. Dalam metode penelitian ini dijabarkan tahapan-tahapan yang akan



Gambar 4. Diagram Alir dari Proses Klasifikasi dengan SVM



Gambar 5. Proses dari Automatic Cropping, (a) Gambar Grayscale, (b) Gambar dari hasil Gaussian Filter (Blur), (c) Gambar Thresholding, (d) Gambar kontur, (e) Gambar kontur yang telah diberi Extreme Point, (f) Gambar yang telah ter-cropping.

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.

**B. 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” [8] serta “Brain MRI Images for Brain Tumor Detection” [9]. Data dapat diunduh dan di dalam folder terbagi menjadi dua bagian, yaitu data “yes” yang berarti terdapat tumor dan sebaliknya data “no” yang berisi tidak ada 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

**C. Image Pre-Processing**

Proses *Image Pre-processing* ini akan menghasilkan citra *automatic cropping*. Digunakan proses *automatic cropping* ini adalah untuk membantu meningkatkan hasil akurasi pada saat proses klasifikasi. Sehingga proses *automatic cropping* akan sangat membantu untuk mendapatkan hasil akurasi yang optimal.

Pada penelitian ini, dataset citra yang telah diunduh kemudian menjadi *input* dari *pre-processing* (Gambar 2). Data gambar yang terdapat dalam folder di muat dalam

*workspace* Jupyter Notebook. Kemudian data tersebut diubah ke dalam citra keabuan (*grayscale*) dengan rentang nilai intensitas 0-255, dimana 0 merupakan intensitas warna hitam dan 255 merupakan intensitas warna putih. 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 *Thresholding*. *Thresholding* merupakan proses konversi citra *grayscale* menjadi citra biner atau hitam putih. Nilai *threshold* yang digunakan yaitu 45 dari 255 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 *thresholding* yang digunakan yaitu *global binary thresholding*. 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 *thresholding*, cari kontur terbesarnya dan kemudian dilakukan proses *cropping*. Mode pengambilan kontur memakai mode pengambilan eksternal. Di mana 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

Tabel 1.  
Hasil dari Texture Feature menggunakan GLCM

	Contrast	Energy	Correlation	Dissimilarit y	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
...	...	...	...	...	...	...	...	...

Tabel 2.  
Hasil Confusion Matrix

Tipe	Kernel	True Positif	False Positif	True Negatif	False Negatif
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 3.  
Hasil Klasifikasi dengan menggunakan SVM

Tipe	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

sesuai pada kontur dan gambar kontur di area tersebut. Pada kontur tersebut, ambil titik (*point*) dari nilai terkecil dan terbesar pada sisi atas, bawah, kanan, dan kiri gambar serta lakukan proses pemotongan (*cropping*) gambar. Proses tersebut dinamakan dengan *Automatic Cropping*.

D. Feature Extraction

Pada *feature extraction*, gambar yang telah di-*cropping* sebelumnya, akan dicari informai fitur yang terkandung dalam suatu pola pada gambar tersebut sehingga akan dapat dengan mudah untuk diproses ke dalam *machine learning*. Ekstraksi fitur yang digunakan menggunakan fitur tekstur. Digunakan metode GLCM (*Gray Level Co-occurance Matrix*) untuk memperoleh informasi fitur 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-occurance* terlebih dahulu. Matriks *co-occurance* yang akan dibuat adalah hubungan spasial antara *distance* (d = 1) dan sudut 0°. Hasil dari matriks *co-occurance* ini berupa jumlah hubungan spasial yang terdapat matriks *grayscale*. Setelah didapatkan nilai dari matriks *co-occurance*, maka proses selanjutnya adalah menghitung nilai

ciri 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 input dari proses klasifikasi tumor otak. Proses pengambilan fitur dapat dilihat pada Gambar 3.

E. Klasifikasi

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*. 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).

Pada proses klasifikasi, yang harus dilakukan pertama kali adalah menginisialisasi 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 lamda = 0.5. Setelah parameter ditentukan, kemudian memasukkan data *training*, lalu mencari nilai *dot product* dari setiap data dengan memasukkan fungsi kernel (K). Setelah didapatkan hasil perkalian *dot product*, kemudian dilakukan proses menghitung matriks dengan menggunakan Persamaan 1.

$$D_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2) \tag{1}$$

Lalu mencari nilai error dari Persamaan 2.

$$E_i = \sum_{j=1}^l \alpha_j D_{ij} \quad (2)$$

Dari hasil error tersebut, kemudian dihitung delta  $\alpha$  dengan menggunakan Persamaan 3.

$$\delta \alpha_i = \min\{\max[\gamma(1 - E_i) - \alpha_i] C - \alpha_i\} \quad (3)$$

Setelah mencari nilai dari delta  $\alpha$ , selanjutnya akan dilakukan proses mencari nilai  $\alpha$  yang baru dengan menggunakan Persamaan 4.

$$\alpha_i = \alpha_i + \delta \alpha_i \quad (4)$$

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 5.

$$b = -\frac{1}{2}((w^T x_{i+}) + (w^T x_{i-})) \quad (5)$$

Pada Persamaan 5, terdapat variabel  $w$  yang belum memiliki nilai, sehingga perlu mencari nilai dari  $w$ . Nilai  $w$  merupakan bobot dari *dot product* data dengan  $\alpha$  terbesar pada masing-masing kelas. Setelah didapatkan nilai  $w$  maka dapat dicari nilai dari biasnya.

Setelah mendapatkan nilai dari  $\alpha$  ( $\alpha$ ), bias ( $b$ ) dan bobot ( $w$ ), maka dapat dilakukan proses pengujian untuk menentukan fungsi keputusan (*decision*). Perhitungan fungsi keputusan dapat menggunakan Persamaan 6.

$$f(x) = \sum_{i=1}^m \alpha_i y_i K(x_i, x_j) + b \quad (6)$$

Sehingga dapat diketahui data mana yang akan masuk ke kelas tumor dan kelas non tumor. Proses training tersebut dapat dilihat pada Gambar 4.

## V. HASIL DAN ANALISA

### A. Image Pre-Processing

Pada pengujian ini, 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*. *Gaussian Smoothing (Blur)* memberikan efek *blur* pada citra. Pada Gambar 5(b) terlihat citra yang telah diberikan efek *blurring*. Dikarenakan citra tersebut hanya memiliki sedikit *noise*, maka *noise* tersebut cukup dihilangkan dengan *Gaussian Blur* saja.

*Thresholding* merupakan proses yang mengubah citra *grayscale* menjadi citra *binary* atau hitam putih. Pada Gambar 5(c) 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 yang terdapat sedikit *noise*. Kegunaan dari *thresholding* sendiri adalah untuk proses segmentasi. Sehingga *thresholding* ini digunakan untuk mengelompokkan wilayah citra otak dengan *backgroundnya*. 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 5(d). Ini berfungsi untuk menentukan *extreme point* pada citra kontur.

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 5(e). 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 titik ekstrim seperti terlihat pada Gambar 5(f) (c) pada penelitian ini ac (d) ut pada citra otak.

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.

### B. Feature Extraction

Setelah melalui tahap *pre-processing*, selanjutnya akan diambil beberapa informasi fitur yang terkandung dalam citra tersebut dan (e) gunakan *feature* (f) *n* yang berupa *texture feat* . *texture feature* meng la 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 1. Dan nantinya hasil ini akan digunakan untuk *learning algorithm* pada proses klasifikasi.

### C. Klasifikasi

Proses klasifikasi ini menggunakan metode *Support Vector Machine (SVM)*. Hasil dataset dari proses fitur ekstraksi kemudian masuk ke dalam proses klasifikasi. Di dalam proses klasifikasi, digunakan 2 tipe SVM untuk mengklasifikasi data tersebut. Klasifikasi pertama menggunakan tipe C-SVM. Kernel yang digunakan yaitu kernel linier, RBF dan polynomial. Hasil perhitungan spesifikasi, presisi, akurasi dan error 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 =  $(FP + FN) / (TP + FP + TN + FN)$ . Hasil dari spesifikasi, presisi, akurasi dan error merupakan hasil perhitungan yang terdapat pada *confusion matrix* yang dapat

dilihat pada Tabel 2. 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. Sehingga secara keseluruhan, kernel RBF yang mendapatkan nilai spesifikasi, presisi dan akurasi yang paling baik.

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. Dari kedua hasil tersebut, dengan nilai gamma 0.5 dapat memiliki tingkat akurasi yang cukup baik. Hasil klasifikasi tersebut dapat dilihat pada Tabel 3.

## VI. KESIMPULAN

Pada penelitian 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. 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.

Dikarenakan hasil akurasi 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 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] N. V. Shree, "Identification and classification of brain tumor MRI images with feature extraction using DWT and probabilistic neural network," *Brain Informatics*, vol. 5, no. 1, pp. 23–30, 2018, doi: 10.1007/s40708-017-0075-5.
- [2] C. Prakarsah, "Brain Tumor Classification Using Gray Level Co-occurrence Matrix and Convolutional Neural Network," no. October, 2018, doi: 10.22146/ijeis.34713.
- [3] K. Vaibhav, S. Rawat, A. K. Yadav, S. Singh, and S. Shekhawat, "Texture Feature Extraction and Classification Using Radial Basis Function for Diagnosis of Brain Tumour," *Far East J. Electron. Commun.*, vol. SpecialVol, no. October, pp. 161–168, 2016, doi: 10.17654/ECSV3P116161.
- [4] A. S. Nugroho, A. B. Witarto, and D. Handoko, "Support Vector Machine," 2003.
- [5] R. Munawarah, O. Soesanto, and M. R. Faisal, "PENERAPAN METODE SUPPORT VECTOR MACHINE," vol. 04, no. 01, pp. 103–113, 2016.
- [6] M. P. K, "Brain Tumor Types and Grades Classification based on Statistical Feature Set using Support Vector Machine," 2015.
- [7] S. Manjunath and D. M. B. S. Pande, "Classification of Brain Tumor Types in MRI Using Support Vector Machine," vol. 10, no. 1, pp. 277–281, 2018, Accessed: Jul. 09, 2020. [Online]. Available: <http://www.jardcs.org/backissues/abstract.php?archiveid=2682>.
- [8] A. Simeon, "Brain Tumor Images Dataset." <https://www.kaggle.com/simeondee/brain-tumor-images-dataset>.
- [9] N. Chakrabarty, "Brain MRI Images for Brain Tumor Detection." <https://www.kaggle.com/navoneel/brain-mri-images-for-brain-tumor-detection>.