OPTIMASI PARAMETER *SUPPORT VECTOR MACHINE* (SVM) *CLASSIFIER* MENGGUNAKAN *FIREFLY ALGORITHM* (FFA) *OPTIMIZATION* UNTUK KLASIFIKASI MRI TUMOR OTAK

SKRIPSI

UIN SUNAN AMPEL
S U R A B A Y A

Disusun Oleh

DEWI WAHYUNI

H02216003

PROGRAM STUDI MATEMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL
SURABAYA

2019

PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini,

Nama	:	DEWI WAHYUNI
NIM	:	H02216003
Program Studi	:	Matematika
Angkatan	:	2016

Menyatakan bahwa saya tidak melakukan plagiat dalam penulisan skripsi saya yang berjudul " Optimasi Parameter *Support Vector Machine* (SVM) *Classifier* Menggunakan *Firefly Algorithm* (FFA) *Optimization* untuk Klasifikasi MRI Tumor Otak ". Apabila suatu saat nanti terbukti saya melakukan tindakan plagiat, maka saya bersedia menerima sanksi yang telah ditetapkan.

Demikian pernyataan keaslian ini saya buat dengan sebenar-benarnya.

Surabaya, 23 Desember 2019

Yang menyatakan,

METERAI TEMPEL 6000 ENAM RIBU RUPIAH

DEWI WAHYUNI

NIM. H02216003

LEMBAR PERSETUJUAN PEMBIMBING

Skripsi oleh

Nama	:	DEWI WAHYUNI
NIM	:	H02216003
Judul Skripsi	:	Optimasi Parameter *Support Vector Machine* (SVM) *Classifier* Menggunakan *Firefly Algorithm* (FFA) *Optimization* untuk Klasifikasi MRI Tumor Otak

telah diperiksa dan disetujui untuk diujikan.

Surabaya, 23 Desember 2019

Pembimbing

Nurissaidah Ulinnuha, M.Kom

NIP. 199011022014032004

PENGESAHAN TIM PENGUJI SKRIPSI

Skripsi oleh

Nama	:	DEWI WAHYUNI
NIM	:	H02216003
Judul Skripsi	:	Optimasi Parameter *Support Vector Machine* (SVM) *Classifier* Menggunakan *Firefly Algorithm* (FFA) *Optimization* untuk Klasifikasi MRI Tumor Otak

Telah dipertahankan di depan Tim Penguji
pada tanggal

Mengesahkan,
Tim Penguji

Penguji I

Nurissaidah Ulinnuha, M.Kom
NIP. 199011022014032004

Penguji II

Dian C. Rini Novitasari, M.Kom
NIP. 198511242014032001

Penguji III

Aris Fanani, M.Kom
NIP. 198701272014031002

Penguji IV

Putroue Keumala Intan, M.Si
NIP. 198805282018012001

Mengetahui,
Dekan Fakultas Sains dan Teknologi
UIN Sunan Ampel Surabaya

Dr. Eni Purwati, M.Ag
NIP. 196512211990022001

KEMENTERIAN AGAMA
UNIVERSITAS ISLAM NEGERI SUNAN AMPEL SURABAYA
PERPUSTAKAAN
Jl. Jend. A. Yani 117 Surabaya 60237 Telp. 031-8431972 Fax.031-8413300
E-Mail: perpus@uinsby.ac.id

LEMBAR PERNYATAAN PERSETUJUAN PUBLIKASI
KARYA ILMIAH UNTUK KEPENTINGAN AKADEMIS

Sebagai sivitas akademika UIN Sunan Ampel Surabaya, yang bertanda tangan di bawah ini, saya:

Nama : DEWI WAHYUNI
NIM : H0221 6003
Fakultas/Jurusan : SAINTEK/ MATEMATIKA
E-mail address : DEWI WAHYUNI 549 @ GMAIL. COM

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Perpustakaan UIN Sunan Ampel Surabaya, Hak Bebas Royalti Non-Eksklusif atas karya ilmiah :
☑ Skripsi ☐ Tesis ☐ Desertasi ☐ Lain-lain (.....)
yang berjudul :

OPTIMASI PARAMETER SUPPORT VECTOR MACHINE (SVM) CLASSIFIER MENGGUNAKAN FIREFLY ALGORITHM (FFA) OPTIMIZATION UNTUK KLASIFIKASI MRI TUMOR OTAK.

beserta perangkat yang diperlukan (bila ada). Dengan Hak Bebas Royalti Non-Ekslusif ini Perpustakaan UIN Sunan Ampel Surabaya berhak menyimpan, mengalih-media/format-kan, mengelolanya dalam bentuk pangkalan data (database), mendistribusikannya, dan menampilkan/mempublikasikannya di Internet atau media lain secara ***fulltext*** untuk kepentingan akademis tanpa perlu meminta ijin dari saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan atau penerbit yang bersangkutan.

Saya bersedia untuk menanggung secara pribadi, tanpa melibatkan pihak Perpustakaan UIN Sunan Ampel Surabaya, segala bentuk tuntutan hukum yang timbul atas pelanggaran Hak Cipta dalam karya ilmiah saya ini.

Demikian pernyataan ini yang saya buat dengan sebenarnya.

Surabaya, 31 DESEMBER 2019

Penulis

(DEWI WAHYUNI)
nama terang dan tanda tangan

ABSTRAK

Optimasi Parameter *Support Vector Machine* (SVM) *Classifier* Menggunakan *Firefly Algorithm* (FFA) *Optimization* untuk Klasifikasi MRI Tumor Otak

Penyakit tidak menular yang sering menjadi penyebab kematian di dunia, salah satunya adalah tumor otak. Selain itu, tumor otak juga menjadi tumor dengan tingkat keganasan kedua setelah tumor darah (leukemia). Penyakit tumor otak berdampak pada penurunan kualitas hidup penderitanya, mengakibatkan beban sosial dan ekonomi bagi negara, keluarga, dan masyarakat. Pada penelitian ini, dilakukan pengolahan citra digital untuk mengklasifikasikan tumor otak dan otak normal sebagai diagnosa penyakit tumor otak. Proses yang dilakukan pada citra digital yaitu *pre-processing* dengan *histogram equalization* dan *median filter*. Selanjutnya dilakukan ekstraksi fitur dengan GLCM dan klasifikasi menggunakan FFA-SVM. Penelitian dilakukan pada 120 data MRI otak dengan proses *learning* menggunakan metode *k-fold* dengan $k = 5$. Berdasarkan pada proses *learning* FFA-SVM, diperoleh dua model terbaik, yaitu FFA-SVM kernel RBF dengan $C = 1874.654$ dan $\gamma = 19.191$ dan FFA-SVM kernel *polynomial* dengan $C = 1796.125$, $\gamma = 19.765$, $r = 0.0005$ dan $d = 6.956$. Berdasarkan *confusion matrix*, akurasi dari kedua model yang didapatkan sebesar 95.83%. Berdasarkan waktu komputasi pada sistem, FFA-SVM kernel RBF meghasilkan waktu sebesar 0.016 detik dan FFA-SVM kernel *polynomial* menghasilkan waktu sebesar 4.784 detik.

Kata kunci: *Histogram Equalization*, *Median Filter*, GLCM, FFA-SVM

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR ISI

HALAMAN JUDUL . . . . . **i**
LEMBAR PERSETUJUAN PEMBIMBING . . . . . **ii**
PENGESAHAN TIM PENGUJI SKRIPSI . . . . . **iii**
HALAMAN PERNYATAAN KEASLIAN . . . . . **iv**
MOTTO . . . . . **v**
HALAMAN PERSEMBAHAN . . . . . **vi**
KATA PENGANTAR . . . . . **vii**
DAFTAR ISI . . . . . **ix**
DAFTAR TABEL . . . . . **xi**
DAFTAR GAMBAR . . . . . **xiii**
DAFTAR LAMBANG . . . . . **xv**
ABSTRAK . . . . . **xvi**
ABSTRACT . . . . . **xvii**
I PENDAHULUAN **1**
1.1. Latar Belakang Masalah 1
1.2. Rumusan Masalah 7
1.3. Tujuan Penelitian 7
1.4. Manfaat Penelitian 8
1.5. Batasan Masalah 8
1.6. Sistematika Penulisan 9
II TINJAUAN PUSTAKA **11**
2.1. Tumor 11
2.2. Tumor Otak 12
2.3. Citra Digital 19
2.4. Pengolahan Citra Digital 20
2.4.1. *Pre-processing* Citra Tumor Otak 20
2.4.2. Ekstraksi Fitur Citra Tumor Otak 25

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.5. *K-fold Cross Validation* 31
2.6. *Firefly Algorithm* (FFA) 32
2.7. *Support Vector Machine* (SVM) 38
2.7.1. Klasifikasi SVM 39
2.7.2. Kernel 47
2.8. Pengukuran Kerja Klasifikator 50
2.9. FFA-SVM 52
III METODE PENELITIAN 54
3.1. Jenis Penelitian 54
3.2. Pengumpulan Data 54
3.3. Analisis Data 54
3.4. Pengujian dan Evaluasi 55
IV HASIL DAN PEMBAHASAN 59
4.1. Perbaikan Citra 59
4.1.1. Data 60
4.1.2. Konversi *Grayscale* 60
4.1.3. *Histogram Equalization* 62
4.1.4. *Median Filter* 66
4.1.5. Ekstraksi Fitur 69
4.2. Pemilihan Model dan Proses *Learning* 82
4.2.1. Klasifikasi SVM 82
4.2.2. *Firefly Algorithm* 87
4.2.3. FFA-SVM 93
4.3. Evaluasi dan Validasi Model 99
4.3.1. Confusion Matrix 99
4.3.2. Evaluasi Waktu 102
V PENUTUP 104
5.1. Simpulan 104
5.2. Saran 105
DAFTAR PUSTAKA 106

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR TABEL

2.1 Kernel SVM 48

4.1 Area Kerja Matriks dengan ukuran matriks 8×8 71
4.2 Nilai Hubungan Spasial Antar Piksel 71
4.3 Nilai Probabilitas Matriks *Co-Occurence* 72
4.4 Nilai *Mean* Baris 74
4.5 Nilai *Mean* Kolom 74
4.6 Nilai Standar Deviasi Baris 75
4.7 Nilai Standar Deviasi Kolom 76
4.8 Nilai Energi 77
4.9 Nilai Homogenitas 78
4.10 Ekstraksi fitur GLCM sudut orientasi $\theta = 0^o$ dan jarak piksel $d_p = 1$ 80
4.11 Ekstraksi fitur GLCM sudut orientasi $\theta = 45^o$ dan jarak piksel $d_p = 1$ 80
4.12 Ekstraksi fitur GLCM sudut orientasi $\theta = 90^o$ dan jarak piksel $d_p = 1$ 81
4.13 Ekstraksi fitur GLCM sudut orientasi $\theta = 135^o$ dan jarak piksel $d_p = 1$ 81
4.14 Sampel Data untuk Klasifikasi SVM 83
4.15 Hasil Klasifikasi Citra MRI Otak Berdasarkan Kernel pada SVM . . 86
4.16 Inisialisasi Parameter *Firefly* 88
4.17 Nilai Acak Populasi *Firefly* 89
4.18 Hasil Penyortiran Nilai Intensitas Cahaya *Firefly* 89
4.19 Pergerakan *Firefly* ke-1 91
4.20 Pergerakan *Firefly* ke-2 91
4.21 *Firefly* Baru 1 92
4.22 Pergerakan *Firefly* ke-3 92
4.23 *Firefly* Baru 2 93
4.24 Nilai Parameter C pada Kernel Linear dan Sudut Orientasi GLCM . 94

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.25 Nilai Parameter C dan γ pada Kernel RBF dan Sudut Orientasi GLCM 95
4.26 Nilai Parameter C, γ, dan r pada Kernel *Sigmoid* dan Sudut Orientasi GLCM 96
4.27 Nilai Parameter C, γ, r, dan d pada Kernel *Polynomial* dan Sudut Orientasi GLCM 97
4.28 Hasil *Confusion Matrix* dari Hasil Terbaik Kernel RBF 99
4.29 Hasil *Confusion Matrix* dari Hasil Terbaik Kernel *Polynomial* 99
4.30 Hasil Analisa *Confusion Matrix* 101
4.31 Hasil Evaluasi Waktu Komputasi 102

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR GAMBAR

2.1 Struktur dan Fungsi Otak 12
2.2 Fakta Tumor Otak di Amerika Serikat 14
2.3 Supratentorial Otak 15
2.4 a.) Otak Normal b.) Tumor Otak 18
2.5 Proses Perubahan RGB ke Citra *Grayscale* 21
2.6 *Histogram Equalization* 23
2.7 Proses *Median Filter* 24
2.8 Empat Sudut dalam GLCM 26
2.9 Matriks Citra *Grayscale* derajat 8 27
2.10 Area Kerja untuk Matriks Citra 27
2.11 Ilustrasi Menghitung Jumlah *Co-Occurrence* 28
2.12 Ilustrasi Menghitung Jumlah *Co-Occurrence* pada Beberapa Sudut (i)-a Sudut $0°$,(ii)-b Sudut $45°$,(iii)-c Sudut $90°$,(iv)-d Sudut $135°$ . . 28
2.13 Penjumlahan Matriks *Co-Occurrence* dengan Transposenya 29
2.14 SVM Berusaha Menemukan *Hyperplane* Terbaik 39
2.15 Transformasi Data SVM Menggunakan Fungsi Kernel 47
2.16 *Confusion Matrix* 51

3.1 Diagram Alir untuk Klasifikasi Tumor Otak 56

4.1 a) Sampel Data Normal b) Sampel Data Tumor 60
4.2 Proses *Grayscale* 61
4.3 Hasil Citra Proses Konversi ke *Grayscale* dengan ukuran 593×479 62
4.4 Perhitungan Jumlah Kemunculan Nilai Piksel pada Citra *Grayscale* 62
4.5 Perhitungan Probabilitas dari Kemunculan Piksel pada Citra *Grayscale* 63
4.6 Perhitungan Transformasi Probabilitas dari Kemunculan Piksel pada Citra *Grayscale* 64

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.7 Perhitungan Perkalian Hasil Transformasi dengan Nilai Rentang Maksimal pada Citra *Grayscale* 64
4.8 Hasil Piksel dari Proses *Histogram Equalization* 65
4.9 Hasil Citra Proses *Histogram Equalization* dengan ukuran 593×479 66
4.10 Penambahan Pading untuk Proses *Median Filter* 66
4.11 Proses *Median Filter* dengan Mask ke-1 67
4.12 Proses *Median Filter* dengan Mask ke-2 67
4.13 Hasil Piksel dari Proses *Median Filter* 68
4.14 Hasil Citra dari Proses *Median Filter* dengan ukuran 593×479 . . . 68

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR LAMBANG

$y_i \in \{-1, 1\}$	: y_i anggota -1,1
$P(i, j)$	: entri matriks peluang P dengan posisi baris ke-i dan kolom ke-j
$\|w\|$	: norm dari w
$\forall_i$	: untuk semua i
$\sum_{j=1}^{k} P_r r_j$	: penjumlahan $P_r r_1 + P_r r_2 + \cdots + P_r r_k$
$\Re^D$	:bilangan real pada suatu dimensi D tertentu
$\sum_{i=1} \sum_{j=1} P(i, j)^2$	: penjumlahan $P(i, j)^2$ yang dimulai dari kolom, kemudian dilanjutkan berdasarkan baris
$\int_0^r P_r(w)dw$	: integral dari $P_r w dw$ dengan batas 0 sampai r
$\frac{\partial L_p}{\partial w}$	:turunan fungsi L_p terhadap w

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB I

PENDAHULUAN

1.1. Latar Belakang Masalah

Penyakit kanker merupakan salah satu penyakit tidak menular yang sering menyebabkan kematian di Indonesia. Penyakit kanker berkontribusi sebesar 0.13 dari 0.22 kematian akibat penyakit tidak menular di dunia. Masalah yang sering terjadi di Indonesia mengenai penyakit kanker yaitu banyak ditemukannya penderita dalam kondisi stadium lanjut (Oemati, Rahajeng, dan Kristanto , 2011). Kanker merupakan penyakit akibat pertumbuhan sel yang tidak normal. Sel yang tumbuh tidak normal dapat membentuk jaringan berupa tumor atau benjolan. Tumor memiliki massa jaringan abnormal yang mungkin padat atau berisi cairan. Ketika suatu pertumbuhan sel-sel tumor terbatas pada tempat asal dan memiliki fisik normal, maka dapat disebut sebagai tumor jinak. Namun, apabila sel-sel tumor tumbuh tanpa terkendali, maka disebut sebagai tumor ganas atau sel kanker (Sinha, 2018).

Penyakit tumor/kanker umumnya sering menyerang orang dewasa daripada anak-anak. Meskipun tumor/kanker pada anak masih jarang, data dari *Union for International Center Control* (UICC) menunjukkan bahwa terdapat 176.000 anak dari negara berpenghasilan rendah dan menengah di dunia yang didiagnosa tumor/kanker setiap tahunnya. Kasus tumor/kanker pada anak Indonesia terdapat sekitar 11.000 setiap tahunnya. Salah satu penyakit tumor/kanker terbanyak pada anak yaitu tumor otak (Kesehatan, 2015). Penyakit tumor otak berdampak pada

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

penurunan kualitas hidup penderitanya, mengakibatkan beban sosial dan ekonomi bagi negra, keluarga, dan masyarakat. tumor otak juga menjadi tumor dengan tingkat keganasan kedua setelah tumor darah (leukemia) (Heranurweni, Destyningtias, dan Nugroho, 2018). Menurut perkiraan *American Cancer Society*, khusus untuk remaja dan anak-anak, tumor otak telah melampaui leukemia dan menjadi pembunuh terbesar (Lang, Zhao, and Jia , 2016).

Otak merupakan salah satu organ yang berperan penting dalam tubuh manusia. Seluruh pergerakan tubuh maupun interpretasi informasi sensorik dikoordinir dan diatur oleh otak. Otak terdiri lebih dari 100 miliar saraf yang saling terhubung, yang disebut sinapsis. Komponen kompleks utama dalam sistem saraf pusat yaitu otak. Satu per lima puluh dari berat badan otak terletak di rongga tengkorak. Otak ditutupi dan dilindungi oleh tulang *Cranial*, *Meninges*, dan *Cerebro Spinal Fluid*. Fungsi otak akan terganggu jika terdapat tumor yang dapat mematikan fungsi saraf di dalam otak (Karuna and Joshi, 2013).

Tumor otak atau neoplasma intrakranial merupakan pertumbuhan tidak normal atau multipliaksi jaringan otak. Tumor otak dapat menghancurkan sel-sel otak dan meningkatkan peradangan di otak (Kumari and Saxena, 2018). Tumor otak terbagi menjadi dua jenis, yaitu tumor otak jinak dan tumor otak ganas. Tumor otak jinak merupakan sel yang tumbuh tidak normal di otak, namun kurang memiliki kemampuan menyebar ke bagian tubuh lainnya. Sedangkan tumor otak ganas dapat disebut sebagai kanker otak. Tumor otak jenis ini dapat dibedakan menjadi dua bagian lagi, yaitu tumor primer yang mulai tumbuh di otak dan tumor sekunder yang dapat disebut tumor metastatis otak yang menyebar ke bagian lain dari tubuh (Benson, et al., 2016).

Faktor resiko yang dapat menyebabkan terjadinya tumor otak merupakan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

suatu hal yang tidak pasti. Studi terus mencari penyebab tumor otak, tetapi hasilnya belum konklusif. Namun, terdapat banyak tren yang didapatkan oleh para peneliti, yaitu usia, jenis kelamin, geografi, pengaruh lingkungan, dan pola prilaku. Selain tren tersebut, penelitian juga memeriksa banyak faktor yang dapat menyebabkan tumor otak, yaitu pengaruh herediter dan genetik, radiasi pengion, paparan infeksi, virus, dan alergi, cidera dan kejang kepala, bahan kimia, diet, medan elektromagnetik telepon seluler dan frekuensi radio, ataupun polusi udara (National Brain Tumor Foundation).

Pencitraan memiliki peranan penting dalam mendiagnosis penyakit tumor otak. *Magnetic Resonance Imaging* (MRI) dan *Computed Tomography* (CT) merupakan pencitraan yang digunakan untuk mendiagnosis penyakit tumor otak. MRI merupakan teknik medis yang lebih disukai daripada CT. Hal ini dikarenakan MRI memberikan gambar yang lebih rinci dari tumor otak (Sehgal, et al., 2016). Dalam banyak kasus yang terjadi, CT menggunakan sinar radiasi pengion, sementara MRI tidak. Sinar ini dapat memberikan radiasi berbahaya jika ada paparan berulang. Selain itu, diagnosis tumor otak pada stadium lanjut sangat penting untuk menyelamatkan hidup seseorang. Deteksi tumor juga harus memiliki kecepatan dan akurasi yang tingggi, sehingga teknik medis yang memungkinkan adalah MRI (Mathew and Anto, 2017).

Deteksi dini tumor otak pada MRI sangat penting untuk mengetahui penanganan selanjutnya pada pasien. Namun, Kemampuan ahli saraf sangat bervariasi dan terbatas pada beberapa tanda-tanda ketidaknormalan awal. Masalah lain yaitu kecerahan dan kontras layar tampilan akan memengaruhi hasil segmentasi. Oleh sebab itu, diperlukan pengembangan metode yang dapat mendeteksi tumor pada MRI otak untuk pengobatan yang tepat pada waktu yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

awal. Sebelum mendeteksi penyakit tumor pada MRI otak, diperlukan pemrosesan terhadap citra untuk mengetahui secara jelas ciri penyakit pada MRI otak.

Proses terhadap citra dinamakan dengan *pre-processing*. *Pre-processing* memiliki beberapa tahap diantaranya *grayscale image*, *filtering*, dan *histogram equalization*. Data citra akan lebih mudah diproses ketika data tersebut berupa data numerik. Oleh sebab itu, perlu dilakukan ekstraksi fitur untuk membuat data numerik dari citra tersebut. Dalam penelitian ini metode yang digunakan untuk ekstraksi fitur adalah *Gray Level Co-Occurrence Matrix* (GLCM). GLCM merupakan metode ekstraksi fitur yang berbasis tekstur pada citra *grayscale*. Metode ini sangat cocok digunakan untuk proses klasifikasi terhadap perbedaan tekstur atau bentuk suatu citra. Penelitian yang dilakukan oleh Sonu Suhag dan L. M. Saini dalam papernya tentang deteksi tumor otak menggunakan metode GLCM sebagai ekstraksi fitur dan SVM sebagai metode klasifikasinya. Beliau mengatakan dalam mengekstraksi fitur sebuah citra *grayscale*, metode yang terbaik adalah GLCM karena dapat dengan mudah mengenali perbedaan tiap karakteristik citranya. Penelitian ini cukup memiliki hasil yang bagus dengan akurasi 94% (Suhag and Saini, 2015).

Tahap selanjutnya yaitu pengklasifikasian pada citra MRI otak. Tahap ini digunakan untuk menentukan penyakit tumor otak. Hasil ekstraksi fitur dari GLCM menjadi inputan awal dari proses ini. Pada proses klasifikasi sendiri, data akan dibagi menjadi dua bagian yakni data training dan data testing. Metode klasifikasi yang dipilih adalah *Support Vector Machines* (SVM). SVM merupakan teknik klasifikasi dengan menemukan *hyperplane* terbaik yang memisahkan data d-dimensional menjadi dua atau lebih kelas menggunakan konsep ruang fitur yang diinduksi kernel. SVM memiliki startegi yang berbeda dengan strategi *neural*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

network. *Neural network* berusaha mencari *hyperplane* pemisah antar kelas, sedangakn SVM berusaha untuk menemukan *hyperplane* terbaik pada *input space*. SVM berdasar pada prinsip *linear classifier* yang kemudian dikembangkan untuk menyelesaikan masalah *nonlinear* dengan memasukkan konsep *kernel trick* pada ruang kerja berdimensi tinggi (Adinegoro, Atmaja dan Purnamasari, 2015). SVM merupakan metode yang baik untuk analisis dan klasifikasi data. Klasifikasi SVM memiliki kecepatan belajar cepat bahkan dalam data besar. Selain itu SVM juga memiliki akurasi yang tinggi dan tingkat kesalahan yang relatif kecil (Harafani dan Wahono, 2015).

SVM merupakan metode yang memiliki cukup banyak keunggulan. Namun hal tersebut juga memungkinkan SVM memiliki kekurangan. Kekurangan dari metode klasifikasi SVM terletak pada penentuan nilai parameter yang optimal. Karena, jika nilai parameter jauh dari optimal, SVM yang dihasilkan tidak akan bekerja secara efektif. Oleh akrena itu perlu adanya algoritma optimasi untuk mengoptimalkan parameter SVM, seperti algoritma metaheuristik, yang menggunakan *tradeoff* antara pengacakan dan pencarian lokal (Sharma, et al., 2013). Beberapa algoritma optimasi yang direkomendasikan oleh peneliti dunia diantaranya *Particle Swarm Optimization* (PSO), *Genetic Algorithm* (GA), dan *Firefly Algorithm* (FFA). Dari ketiga algoritma tersebut, FFA merupakan algoritma optimasi yang memebrikan hasil optimasi terbaik (Yang, 2009). Dengan demikian, dalam penelitian ini, SVM ditingkatkan dengan bantuan salah satu algoritma metaheuristik yang baru dikembangkan, algoritma *Firefly Algorithm* (FFA).

Penemuan algoritma *Firefly* terjadi akhir-akhir ini pada tahun 2007 oleh Dr. Xin-She Yang. Pendekatan menggabungkan FFA dengan SVM adalah bahwa perilaku kunang-kunang dapat dirumuskan untuk membentuk fungsi objektif, yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dapat membantu mengoptimalkan parameter SVM. Dengan algoritma *Firefly*, ditetapkan bahwa semakin banyak perbandingan antara kunang-kunang untuk menemukan lokasi terbaik di kawanan, maka semakin baik hasilnya. Perbandingan ini didasarkan pada intensitas cahaya kunang-kunang. Konsep ini memanfaatkan fakta bahwa, dengan lebih banyak kecerahan kunang-kunang, lebih banyak daya tarik dengan kunang-kunang lainnya, yang mengurangi jarak di antara mereka. Dengan demikian, semakin pendeknya jarak kumpulan kunang-kunang dapat menemukan posisi terbaik, yang juga memberikan nilai parametrik terbaik yang dapat dimasukkan sebagai input ke pengklasifikasi (Al-Shammari, et al., 2015).

Penelitian yang dilakukan oleh Oluyinka Aderemi Adewumi dan Ayobami Andronicus Akinyelu menunjukkan bahwa dalam klasifikasi pengelabuan email dengan optimasi SVM mengggunakan algoritma FFA memberikan hasil akurasi yang sangat tinggi, yaitu 99,98%. Hasil ini mengungkapkan bahwa kinerja keseluruhan FFA-SVM mengungguli SVM standar (Adewumi and Akinyelu, 2016). Berdasarkan penelitian tersebut, maka dilakukan penelitian menggunakan *Firefly Algorithm* (FFA) *Optimization* dan *Support Vector Machine* (SVM) *Classifier* sehingga bisa mendapatkan hasil yang maksimal untuk deteksi dini penyakit tumor otak. Sesuai dengan ajaran Islam, bahwasannya dalam memberikan pelayanan, baik pelayanan dari usaha barang ataupun jasa seharusnya diberikan dengan pelayanan terbaik dan berkualitas. Hal ini telah difirmankan oleh Allah pada Al-Qur'an surah Al-Baqarah ayat 267, yang artinya "*Hai orang-orang yang beriman, nafkahkanlah (di jalan Allah) sebagian dari hasil usahamu yang baik-baik dan sebagian dari apa yang kamu keluarkan dari bumi untuk kamu dan janganlah kamu memilih yang buruk-buruk lalu kamu nafkahkan darinya padahal kamu sendiri tidak mau mengambilnya melainkan dengan memicingkan mata*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

terhadapnya. Dan ketahuilah bahwa Allah Maha Kaya lagi Maha Terpuji" (QS. Al-Baqarah(2):267).

Berdasarkan ayat tersebut, salah satu bentuk pelayanan yang baik dan berkualitas dalam dunia kesehatan adalah ketepatan dan kecepatan para ahli dalam mendiagnosa suatu penyakit. Oleh karena itu dilakukan penelitian ini sebagai pengembangan metode untuk mendeteksi penyakit tumor otak. Metode ini dapat digunakan sebagai bahan pertimbangan seorang ahli dalam menentukan diagnosa penyakit tumor otak pada pasien. Selain itu, sistem klasifikasi yang dibuat juga dapat digunakan untuk meminimalisir waktu karena sistem kerja yang dilakukan oleh komputer.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang ada, maka peneliti merumuskan permasalahan sebagai berikut:

1. Bagaimana hasil kinerja dari *Support Vector Machine* (SVM) *Classifier* dalam klasifikasi penyakit tumor otak?
2. Bagaimana hasil pengoptimasian parameter input pada SVM menggunakan metode *Firefly Algorithm* (FFA) dalam klasifikasi penyakit tumor otak?

1.3. Tujuan Penelitian

Berdasarkan rumusan masalah yang ada, maka tujuan penelitian ini adalah:

1. Dapat mengetahui hasil dari kenerja *Support Vector Machine* (SVM) *Classifier* dalam klasifikasi penyakit tumor otak.
2. Dapat mengetahui hasil optimasi parameter input SVM menggunakan metode

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Firefly Algorithm (FFA) dalam klasifikasi penyakit tumor otak.

1.4. Manfaat Penelitian

1. Bagi penulis

   Dapat menerapkan ilmu yang diperoleh selama perkuliahan dan dapat menambah wawasan berkaitan dengan citra digital.

2. Bagi Instansi Kesehatan

   Diharapkan dapat membantu diagnosis penyakit tidak menular, yaitu tumor otak.

3. Bagi Pembaca

   Dapat menambah pengetahuan dalam mengidentifikasi permasalahan serta dapat memberikan usulan terhadap penyakit tumor otak.

1.5. Batasan Masalah

Mengingat ruang lingkup penelitian yang begitu luas, maka penulis memberi batasan pada penelitian ini, yaitu:

1. Data yang digunakan dalam penelitian ini menggunakan data citra MRI tumor otak.

2. Klasifikasi tumor otak di bagi dalam dua kategori yaitu normal dan tumor otak.

3. Keluaran dari sistem ini adalah berupa informasi hasil klasifikasi normal dan tumor otak dengan menggunakan SVM dan optimasi FFA

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1.6. Sistematika Penulisan

BAB I PENDAHULUAN

Bab ini berisi tentang penjelasan mengenai permasalahan apa saja yang menjadi latar belakang dilakukannya penelitian, yaitu banyaknya kematian di dunia akibat penyakit tidak menular seperti tumor otak yang terus bertambah dan kemampuan ahli saraf yang sangat bervariasi dan terbatas untuk menentukan diagnosa dari hasil pencitraan medis. Selain itu pada bagian ini juga menjelaskan mengenai pemilihan dan gambaran singkat metode *preprocessing*, GLCM, *Firefly Optimization*, dan SVM *Classifier* yang digunakan untuk pengolahan citra medis dalam klasifikasi penyakit tumor otak, rumusan masalah, tujuan penelitian, manfaat penelitian, batasan masalah, dan sistematika penulisan.

BAB II DASAR TEORI

Bab ini berisi tentang teori-teori yang memiliki keterkaitan dan mendukung penyelesaian masalah dalam penelitian ini. Tinjauan pustaka terkait topik yang digunakan yaitu teori tentang tumor, tumor otak, citra digital, pengolahan citra digital, *Firefly Algorithm* (FFA), *Support Vector Machine* (SVM), dan pengukuran kerja klasifikasi.

BAB III METODE PENELITIAN

Bab ini berisi tentang cara melaksanakan penelitian yang dimulai dari bagaimana proses mendapatkan data dan mengolah data sedemikian sehingga rumusan masalah dapat terselesaikan. Bab ini biasanya terdiri dari waktu pelaksanaan penelitian, jenis penelitian, pengumpulan data, teknik analisis data, dan pengujian serta evaluasi sistem.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB IV HASIL DAN PEMBAHASAN

Bab ini berisi tentang uraian hasil penelitian dan pembahasan dalam rangka untuk menjawab rumusan masalah yang telah diuraikan pada bab pendahuluan. Dalam penelitian ini diuraikan mengenai hasil perbaikan citra MRI otak, ekstraksi fitur GLCM, hasil optimasi parameter SVM, klasifikasi menggunakan SVM, dan analisis dari hasil yang diperoleh.

BAB V PENUTUP

Bab ini berisi tentang kesimpulan dari keseluruhan isi yang telah dibahas dan saran perluasan mengenai hasil penelitian dari analisis citra MRI otak untuk mengklasifikasi penyakit tumor otak.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB II

TINJAUAN PUSTAKA

2.1. Tumor

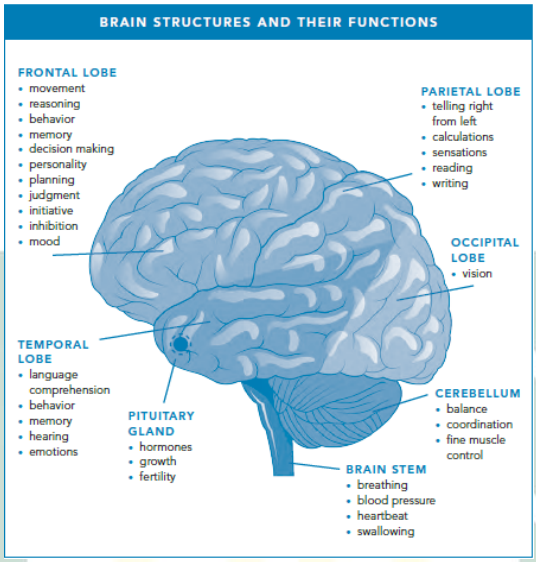
Tumor disebut juga dengan *neoplasia* yang tersusun dari kata *neo* dan *plasia*. *Neo* berarti baru dan *plasia* berarti pembelahan atau pertumbuhan (Saleh, 2016). Tumor dapat diartikan sebagai pertumbuhan sel baru yang tidak normal dan tumbuh secara berlebihan. Tumor memiliki massa jaringan abnormal yang mungkin padat atau berisi cairan. Ketika suatu pertumbuhan sel-sel tumor terbatas pada tempat asal dan memiliki fisik normal, maka dapat disebut sebagai tumor jinak. Namun, apabila sel-sel tumor tumbuh tanpa terkendali, maka disebut sebagai tumor ganas atau sel kanker. Dalam menentukan jinak atau ganasnya suatu tumor, maka dapat dilakukan prosedur biopsi dari sampel sel yang diambil. Kemudian biopsi dianalisis di bawah mikroskop (Sinha, 2018).

Tumor jinak tidak menyerang jaringan terdekat atau menyebar ke area tubuh yang lain. Tumor jinak tidak terlalu berbahaya, kecuali berada di sekitar organ, jaringan, saraf, atau pembuluh darah, dan menyebabkan kerusakan. Tumor jinak dapat diangkat dengan melakukan operasi. Biasanya tumor jinak tidak tumbuh lagi setelah dilakukan operasi pengangkatan tumor. Contoh tumor jinak yaitu tahi lalat, kista fibroid di payudara, dan polip usus besar (Crum, Lester, dan Cotran, 2007). Tumor ganas yaitu tumor yang terbuat dari sel kanker dan dapat menyerang jaringan di sekitarnya. Beberapa sel kanker dapat menyebar ke aliran darah atau kelenjar getah bening untuk menyebar ke jaringan lain di dalam tubuh. Hal ini disebut juga

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sebagai metastatis (Sinha, 2018).

2.2. Tumor Otak



Gambar 2.1 Struktur dan Fungsi Otak

(Cancer Support Community, 2013)

Central Nervous System (CNS) mengontrol fungsi dan pikiran tubuh manusia yang terdiri dari otak, sumsum tulang belakang, dan neuron kompleks. Komponen kompleks utama dalam sistem saraf pusat adalah otak. Gambar 2.1 menunjukkan bentuk, struktur, dan fungsi otak. 1 per 50 dari berat badan otak terletak di rongga tengkorak. Otak ditutupi dan dilindungi oleh tulang *Cranial*, *Meninge*, dan *Cerebro Spinal Fluid* (CSF). Otak adalah massa sel saraf yang lunak dan kenyal dan jaringan pendukung yang terhubung ke sumsum tulang belakang. Di pusat otak terdapat empat ruang kosong yang terhubung, disebut ventrikel. Ventrikel mengandung cairan yang disebut *cerebrospinal fluid* (CSF) yang bersirkulasi di seluruh sistem saraf pusat. Otak dan sumsum tulang belakang (sistem saraf pusat) mengendalikan fungsi fisiologis dan psikologis tubuh. Secara umum otak mencakup tiga bagian utama (National Brain Tumor Foundation):

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

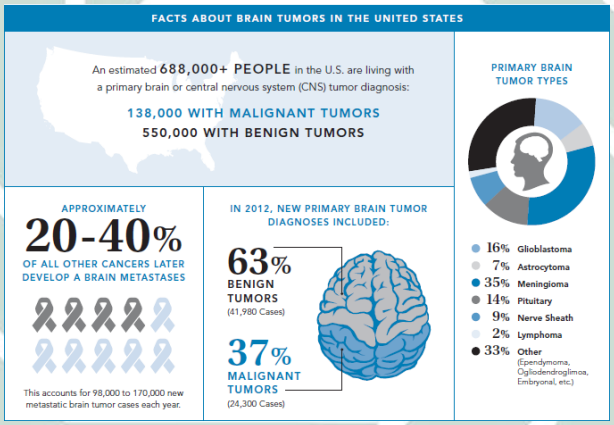
1. *Cerebrum* mengendalikan pikiran, pembelajaran, penyelesaian masalah, emosi, ucapan, membaca, menulis, dan gerakan sukarela.
2. Otak kecil mengendalikan gerakan, keseimbangan, dan postur.
3. Batang otak menghubungkan otak ke sumsum tulang belakang, dan mengontrol pernapasan, detak jantung, dan saraf serta otot yang digunakan untuk melihat, mendengar, berjalan, berbicara, dan makan.

Tumor otak atau neoplasma intrakranial merupakan pertumbuhan tidak normal atau multiplikasi jaringan otak. Tumor otak dapat menghancurkan sel-sel otak dan meningkatkan peradangan di otak, sehingga mengganggu fungsi otak (Kumari and Saxena, 2018). Tumor otak disebabkan oleh faktor genetik, yaitu adanya ketidaknormalan gen yang mengontrol pertumbuhan sel otak. Penyebab ketidaknormalan ini adalah gangguan pada kromosom atau kelainan yang langsung mengenai gen. Tumor terbagi menjadi tingkat 1 hingga 4 secara garis besar dikelompokkan menjadi dua jenis tumor otak, yaitu tumor otak jinak (tingkat 1 dan 2) dan tumor otak ganas (tingkat 3 dan 4). Pengelompokkan ini didasarkan oleh sifat dari tummor otak (Benson, et al., 2016).

Tumor otak jinak merupakan jenis tumor otak yang paling tidak agresif. Tumor ini berasal dari sel-sel yang berada di sekitar atau dalam otak, tidak mengandung sel kanker, pertumbuhannya lambat, dan biasanya memiliki batasan jelas yang tidak menyebar ke jaringan lain. Jika tumor ini dapat diangkat seluruhnya, tumor cenderung tidak tumbuh kembali. Namun, dapat menyebabkan gejala neurologis yang signifikan tergantung pada ukurannya, dan lokasi di dekat struktur lain di otak. Beberapa tumor jinak dapat berkembang menjadi ganas. Terdapat sel kanker pada tumor ganas dan seringkali tidak memiliki batas yang jelas. Tumor otak ganas dianggap mengancam jiwa karena tumor tumbuh dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

cepat dan menyerang jaringan otak di sekitarnya. Meskipun tumor otak ganas sangat jarang menyebar ke area lain dari tubuh, dapat menyebar ke seluruh otak atau ke tulang belakang (Cancer Support Community, 2013). Tumor otak jenis ini dapat dibedakan menjadi dua bagian lagi, yaitu tumor primer dan tumor sekunder yang dapat disebut tumor metastatis. Tumor primer dimulai di sel-sel otak dan dapat menyebar ke bagian otak lainnya atau ke tulang belakang. Tumor otak metastatik atau sekunder bermula di bagian lain dari tubuh dan kemudian menyebar ke otak. Tumor ini lebih umum daripada tumor otak primer (Kumari and Saxena, 2018).



Gambar 2.2 Fakta Tumor Otak di Amerika Serikat

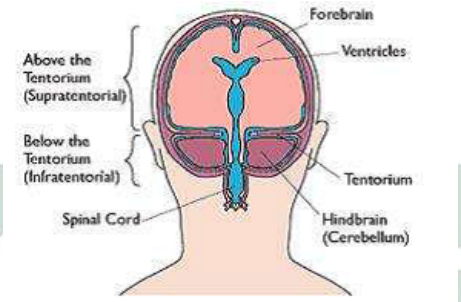
(Cancer Support Community, 2013)

Gambar 2.2 menunjukkan bahwa penyakit tumor otak di Amerika Serikat telah diderita oleh lebih dari 688.000 orang dengan rincian sekitar 138.000 orang menderita tumor ganas dan 550.000 orang menderita tumor jinak. Selain itu, sekitar 20-40% kasus kanker lain bermetastatis di otak. Pada tahun 2012, kasus tumor otak dengan kondisi tumor jinak sebesar 63% dan kondisi tumor ganas sebesar 37%. Tipe penyakit tumor otak yang sering terjadi di Amerika Serikat yaitu jenis tumor otak meningioma. Penyakit tumor otak memiliki bebrapa tipe, diantaranya *gliomas*, *meningiomas*, *schwannomas*, *pituitary tumors*, dan *central*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

nervous system lymphoma (Cancer Support Community, 2013).

Gliomas merupakan merupakan tumor yang tumbuh dan berkemabng di jaringan glia dan saraf tulang belakang. Tumor ini terletak di daerah supratentorial seperti pada Gambar 2.3.



Gambar 2.3 Supratentorial Otak

(Salamah, 2018)

Meningiomas adalah tumor otak yang biasanya tumbuh lambat. Tumor ini merupakan tumor jinak yang berasal dari selubung luar otak, tepat di bawah tengkorak. Tumor jenis ini menyumbang sekitar sepertiga dari tumor otak pada orang dewasa. *Schwannomas* adalah tumor jinak yang muncul dari sel-sel saraf pendukung yang disebut schwannoma vestibular atau neuroma akustik. Schwannomas vestibular sering menyebabkan gangguan pendengaran, masalah keseimbangan, atau kelemahan di satu sisi wajah. *Pituitary tumors* merupakan tumor yang tumbuh di permukaan kelenjar pituitari. Kelenjar ini berperan dalam mengatur aktivitas dan hormon seks. *Central nervous system lymphoma* merupakan tumor otak ganas yang berasal dari limfosit di otak, sumsum tulang belakang, atau mata (Cancer Support Community, 2013). Berikut akan dijelaskan mengenai gejala dan deteksi penyakit tumor otak (Cancer Support Community, 2013).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

1. Faktor Resiko

Faktor resiko terjadinya penyakit tumor otak diantaranya yaitu usia, genetik, riwayat kesehatan individu, paparan radiasi ion, serta telepon seluler dan radio. Penderita tumor otak banyak terjadi pada usia anak-anak dan usia tua ($>$ 50 tahun). Sedangkan menurut riwayat kesehatan individu meliputi:

a. Riwayat kanker sebelumnya.

b. Faktor infeksi, alergi, dan imunologis.

c. Riwayat trauma kepala.

d. Riwayat kejang.

e. Diet, vitamin, alkohol, rokok, dan agen kimia lain.

Radiasi ion yang dapat menyebabkan penyakit tumor otak yaitu paparan radiasi diagnostik dan terapi. Pada studi *case control*, resiko terjadinya tumor otak dapat diakibatkan oleh penggunakan telepon genggam selama 10 tahun dan banyaknya jumlah bicara menggunakan telepon genggam (Ardiansyah, 2018).

2. Deteksi Tumor Otak

Gejala tumor otak dapat dilihat dari ukuran, lokasi, dan tingkat pertumbuhan tumor itu sendiri. Penyakit tumor otak tumbuh secara perlahan-lahan, mungkin hampir tidak menunjukkan gejala yang signifikan, bahkan gejala yang nampak di awal hanya kelelahan dan sakit kepala. Setelah beberapa waktu, tumor akan mulai memberikan tekanan dan membuat sebagian fungsi otak tidak bekerja secara maksimal. Gejala yang mulai terlihat lebih signifikan, seperti kejang-kejang, sakit kepala berat, perubahan mental, penurunan tajam penglihatan dan pendengaran, kelemahan atau kesemutan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

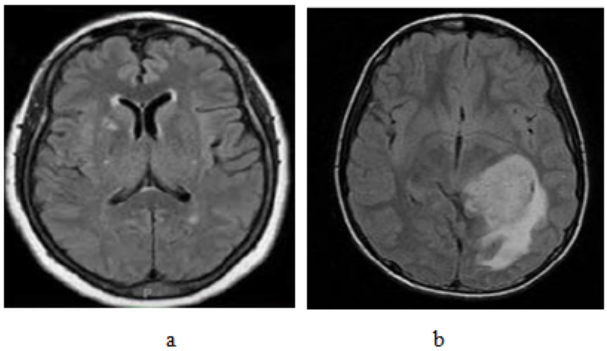
anggota tubuh, gangguan kesadaran, dan lain sebagainya (Heranurweni, Destyningtias, dan Nugroho, 2018).

Diagnosis atau deteksi tumor otak didasarkan pada gambaran klinis, tanda peninggian tekanan intra kranial dan manifestasi neurologis fokal, dan pemeriksaan menunjang. Langkah-langkah yang dilakukan setelah anamnasis dan pemeriksaan umum yaitu:

a. Pemeriksaan neurologis meliputi kesadaran, pemeriksaan syaraf kranial, tonus otot, kekuatan otot, pemeriksaan refleks tendon dalam dan superfisial baik yang fisiologis maupun patologis, pemeriksaan sensibilitas, koordinasi, gangguan gerak (*involunter movement*, ataxia).
b. *Computed Tomography Scan* (CT-*Scan*) yang merupakan pemeriksaan menggunakan sinar-X dan dengan penggunaan komputer yang akan menghasilkan gambar organ-organ tubuh manusia.
c. *Magnetic Resonance Imaging* (MRI).

   MRI otak adalah teknik pencitraan standar untuk tumor otak. MRI merupakan perangkat pemindaian yang mneggunakan medan magnet dan komputer untuk menangkap gambar otak pada film. Dalam pemeriksaannya, MRI tidak menggunakan sinar-X. MRI memungkinkan dokter untuk membuat citra tumor tiga dimensi. MRI mendeteksi sinyal yang dipancarkan dari jaringan normal dan abnormal, memberikan gambar yang jelas dari sebagian besar tumor (National Brain Tumor Foundation). Dalam laporan MRI, daerah yang terdeteksi kanker memiliki piksel intensitas tinggi dan jaringan normal memiliki piksel intensitas rendah seperti pada Gambar 2.4(Murthy and G.Sadashivappa, 2014).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.4 a.) Otak Normal b.) Tumor Otak

(Murthy and G.Sadashivappa, 2014)

Biasanya jaringan otak yang sehat meliputi materi putih, cairan serebrospinal dan materi abu-abu. Dalam jaringan otak yang sehat, materi putih adalah rasio spesifik. Tetapi jika dibandingkan dengan jaringan yang terkena tumor, ketiga nilai jaringan ini seluruhnya dalam bentuk yang berbeda dan tidak dalam rasio tertentu. Penerapan segmentasi tumor bertujuan untuk memisahkan area jaringan yang terkena tumor otak untuk tujuan analisis. Aplikasi MRI menyediakan berbagai jenis citra tumor otak yang sepenuhnya didasarkan pada tingkat kontras otak, komposisi cairan dan bentuk tumor dalam kandungan cairan (Virupakshappa and Amarapur, 2017).

d. *Ultrasonography* kepala dilakukan jika ubun-ubun masih terbuka, atau jika tidak bisa dilakukan CT-*Scan* kepala atau MRI.

e. Pemeriksaan cairan serebrospinal, pungsi lumbal merupakan kontraindikasi bila ditemukan peningkatan tekanan intrakranial. Namun jika diperlukan bisa dilakukan dengan hati-hati.

f. Pemeriksaan patologi anatomi, pemeriksaan ini dapat menentukan jenis tumor dengan pasti namun tidak selalu dapat dilakukan biopsi atau operasi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

terhadap tumor otak.

Tata laksana tumor memerlukan kerjasama tim yang terdiri dari dokter anak, syaraf anak, radiologi, bedah syaraf, dan radioterapi. Selain itu juga terdapat tindakan operatif untuk mengangkat tumor, radiotrapi untuk mematikan sel secara selektif, dan pemberian kemotrapi (Estlin and Lowis, 2005).

2.3. Citra Digital

Citra merupakan hasil dari proses sampling berupa gambar dua dimensi. Gambar tersebut adalah gambar yang dihasil dari gambar analog kontinu menjadi gambar diskrit. Gambar analog terbagi menjadi kumpulan matriks yang terdiri dari N baris dan M kolom. Hubungan antara baris dan kolom tertentu merepresentasikan suatu piksel. Sebagai contoh piksel $[n, m]$ merepresentasikan titik diskrit pada baris n dan kolom m (Tim Dosen, 2016).

Citra digital merupakan citra yang dapat diolah dengan komputer digital. Citra digital disusun oleh elemen-elemen yang disebut piksel. Kumpulan piksel direpresentasikan oleh indeks baris dan kolom (x, y) dengan bilangan bulat dalam struktur dua dimensi. x dan y adalah koordinat spasial dari f dalam fungsi $f(x, y)$. Nilai f menunjukkan *brightness* atau *grayness level* dari suatu citra pada titik tersebut. Suatu citra digital dapat direpresentasikan dalam format $f(x, y) = f(N, M)$ dengan $0 \leq x \leq N - 1$ dan $0 \leq y \leq M - 1$. Kemudian kedua faktor tersebut dinyatakan dengan nilai L yang merupakan nilai maksimal warna intensitas dengan ketentuan $L = 0 \leq f(x, y) \leq L - 1$. Citra digital dibagi dalam tiga jenis, yaitu citra *grayscale*, citra biner, dan citra warna (Tim Dosen, 2016).

Citra *grayscale* yaitu citra dengan skala keabuan yang memiliki nilai

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

intensitas maksimal 255 berwarna putih hingga warna hitam dengan nilai intensitas minimal 0. Citra biner adalah citra yang hanya memiliki dua nilai derajat keabuan, yaitu 0 dan 1 sebagai representasi dari putih dan hitam. Piksel yang bernilai 1 merupakan piksel objek, dan piksel yang bernilai 0 adalah latar belakang. Sedangkan citra warna atau yang biasa disebut citra RGB yaitu jenis citra yang merepresentasikan warna dalam komponen R (merah), G (hijau), dan B (biru). Setiap warna menggunakan 8 bit (nilainya berkisar dari 0 hingga 255). Oleh karena itu kemungkinan warna yang dapat disajikan adalah 255 x 255 x 255 (Wahyudi, Triyanto, dan Ruslianto, 2017). Citra digital dapat direpresentasikan dalam suatu fungsi $f(x, y)$. x adalah koordinat posisi dengan $0 \leq x \leq M$, dan y adalah koordinat posisi dengan $0 \leq y \leq N$, dimana M merupakan lebar citra, dan N adalah tinggi citra. Sedangkan f merupakan fungsi citra digital, dimana $0 \leq f \leq L$, dengan L sebagai derajat keabuan citra (Tim Dosen, 2016).

2.4. Pengolahan Citra Digital

Pada proses pengklasifikasian citra medik, perlu dilakukan teknik pengolahan citra untuk mendapatkan hasil yang akurat. Tahapan-tahapan yang perlu dilakukan pada proses pengolahan citra yaitu *pre-processing*, ekstraksi fitur, dan klasifikasi citra medik.

2.4.1. *Pre-processing* Citra Tumor Otak

Kualitas citra tumor otak yang telah diperoleh belum memenuhi standar pengolahan, sehingga tidak dapat digunakan secara langsung. Penyebaran intensitas pada citra tersebut masih kurang merata akibat pencahayaan dan kontras yang lemah. Oleh karena itu perlu dilakukan perbaikan atau peningkatan kualitas citra untuk memudahkan pemisahan objek dengan latar belakang pada citra.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Langkah-langkah *pre-processing* citra tumor otak adalah sebagai berikut :

Konversi Citra ke *Grayscale Image*

Gambar yang diterima dari MRI biasanya diwarnai oleh komponen RGB (merah, hijau dan biru). Namun, gambar dikonversi menjadi gambar skala abu-abu dengan menghilangkan informasi kecerahan, sehingga mengubah format gambar dari 512 x 512 x 3 warna RGB menjadi 512 x 512 gambar abu-abu dengan nilai s yang dapat dicari menggunakan Persamaan 2.1:

$$s = (0.299(r)) + (0.587(g)) + (0.114(r)) \tag{2.1}$$

Dimana, s adalah citra *grayscale*, r adalah nilai matriks warna merah, g adalah nilai matriks warna hijau, dan b adalah nilai matriks warna biru (Santi, 2011).

R

66	61	59
64	62	60
68	64	61

G

35	36	37
36	34	35
37	37	36

B

60	62	65
62	60	63
66	66	63

Grayscale Image

47	46	47
47	45	46
50	48	47

Gambar 2.5 Proses Perubahan RGB ke Citra *Grayscale*

Gambar 2.5 merupakan proses perubahan citra RGB menjadi citra *grayscale*. Piksel *grayscale* diperoleh dengan menggunakan Persamaan 2.1. Sebagai contoh, misalkan nilai s_{11} diperoleh dengan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$(0.299(66)) + (0.587(35)) + (0.114(60)) = 47$, dengan cara yang sama dihitung nilai piksel *grayscale* lainnya.

***Histogram Equalization* (HE)**

Peningkatan kontras merupakan bagian penting di pemrosesan gambar digital untuk persepsi visual manusia dan visi komputer. Proses ini banyak digunakan untuk pemrosesan citra medis sebagai langkah *pre-processing* dalam sintesis tekstur. Terdapat banyak metode yang dikembangkan dalam peningkatan kontras suatu citra. *Histogram equalization* (HE) merupakan salah satu metode yang populer dalam peningkatan kontras suatu citra. HE adalah teknik peningkatan kontras pada domain spasial dalam pemrosesan citra menggunakan histogram citra. HE biasanya meningkatkan kontras global dari citra yang diproses. Metode ini berguna untuk citra yang terang atau gelap (Patel, Maravi, and Sharma , 2013).

Gambar histogram menyediakan informasi terkait dengan penyebaran intensitas piksel ke dalam gambar. Gambar yang terlalu terang atau terlalu gelap akan memiliki histogram yang sempit. Oleh karena itu, diperlukan pengubahan penyebaran nilai intensitas gambar dengan menggunakan peningkatan gambar (Zulpe and Pawar, 2012). Histogram dari citra digital dengan tingkat intensitas total L yang mungkin dalam rentang $[0, G]$ didefinisikan sebagai fungsi terpisah:

$$h(r_k) = n_k \tag{2.2}$$

dengan r_k adalah tingkat keabuan ke-k dan n_k adalah jumlah piksel pada citra dengan tingkat keabuan r_k. Nilai G adalah 255 untuk gambar dari kategori uint8. Selanjutnya nilai histogram dinormalisasi dengan membagi setiap nilai histogram dengan jumlah total histogram. Berikut Persamaan 2.3 digunakan untuk

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menghitung nilai normalisasi histogram.

$$p(r_k) = \frac{n_k}{n} \tag{2.3}$$

dengan n adalah jumlah piksel dalam citra (Arif, Khan, and Siddique, 2018).

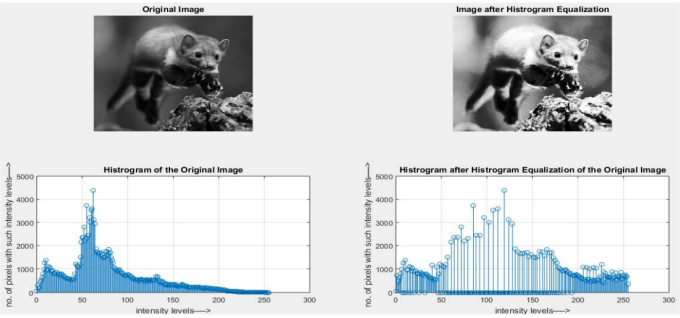
Kisaran normal tingkat dominasi adalah [0, 1]. Dalam fungsi PDF, perbedaan PDF dari citra input dan output ditandai oleh subskrip. Transformasi pada level input dilakukan untuk mencapai tingkat intensitas *output* dengan:

$$s = T(r) = \int_0^r p_r(w)dw \tag{2.4}$$

dimana $p_r(r)$ merupakan probabilitas kemunculan tingkat keabuan. Citra dihasilkan oleh transformasi sebelumnya yang berisi level intensitas serupa. Hasil asli dari proses pemerataan untuk tingkat intensitas ini adalah citra yang berisi rentang progresif dan kontras yang lebih baik. Fungsi transformasi dalam bentuk diskrit yaitu (Arif, Khan, and Siddique, 2018):

$$s_k = T(r_k) = \sum_{j=0}^{k} p_r(r_j) \tag{2.5}$$

Hasil dari proses *Histogram Equalization* dapat dilihat pada Gambar 2.6.



Gambar 2.6 ***Histogram Equalization***

(Arif, Khan, and Siddique, 2018)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

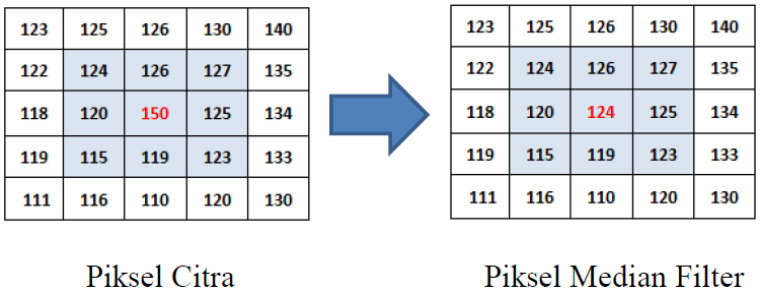
Filtering

Filter median adalah metode nonlinear yang memiliki fungsi untuk mengurangi gangguan pada citra berupa *noise* atau untuk menghaluskan citra. *Noise* yang efektif untuk dihilangkan oleh filter ini yaitu *noise* jenis *salt and pepper*. Selain itu, filter ini juga mampu untuk mempertahankan detail citra karena tidak tergantung pada perbedaan nilai-nilai ketetanggaannya (Pragathi and Patil, 2013).

Filter median bekerja dengan mengambil nilai *mask* yang telah ditentukan, umumnya 3 x 3. Proses pengklasifikasian piksel sebagai *noise* pada filter median dilakukan dengan proses perbandingkan setiap piksel pada piksel tetangga sekitarnya (*mask*). *Impulse noise* yang dilabelkan merupakan nilai piksel yang berbeda dari mayoritas tetangganya. *Noise* piksel tersebut akan diganti nilainya dengan nilai piksel median dari piksel pada ukuran tetangganya. Citra yang difilter, didefinisikan sebagai:

$$I(i,j) = median\{(c,d)\} \in W_{m,n}\{d(i \dot{+} c, j \dot{+} d)\} \tag{2.6}$$

dimana $W_{m,n}$ adalah *sliding window* berukuran m x n dengan piksel tengah pada (*i,j*) (Maulana, dkk., 2016).



Gambar 2.7 Proses *Median Filter*

(Maulana, dkk., 2016)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Gambar 2.7 merupakan proses dari *median filter* dengan *sliding window* 3×3. Nilai piksel pada *sliding window* diurutkan menjadi 115, 119, 120, 123, 124, 125, 126, 127, 150. Dari hasil pengurutan, diperoleh nilai *median* 124. Nilai *median* ini digunakan untuk menggantikan nilai pusat *sliding window*. Oleh karena itu nilai 150 diganti menjadi 124 (Maulana, dkk., 2016).

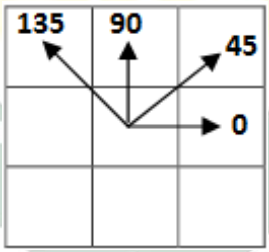
2.4.2. Ekstraksi Fitur Citra Tumor Otak

Fitur adalah informasi khusus pada citra. Dengan adanya fitur, peneliti dapat menentukan karakteristik dari sebuah citra sehingga citra tersebut dapat diklasifikasi dengan benar berdasarkan karakteristik yang didapat (Ahmad, 2005). Citra memiliki beberapa elemen di dalamnya sebagai bentuk karakteristik suatu citra. Elemen-elemen tersebut diantaranya yaitu kecerahan, kontras, kontur, warna, bentuk, dna tekstur. Ekstraksi fitur adalah langkah utama yang spesifik dalam proses analisis tekstur dan menghasilkan perhitungan fitur tekstur dari ROI yang telah ditentukan. Banyak pendekatan yang diusulkan untuk mengukur tekstur citra, salah satunya yang paling populer dan sukses digunakan untuk mengkarakterisasi jaringan MRI yaitu GLCM (*Gray Level Co-Occurrence Matrix*) yang diperkenalkan oleh Haralick pada tahun 1979 (Larroza, Bodi, and Moratal, 2016). Dalam analisis tekstur, fitur tekstur dihitung melalui distribusi statistik dari kombinasi intensitas yang diamati. Menurut jumlah titik intensitas (piksel) dalam setiap kombinasi, statistik dibedakan menjadi statistik orde satu, orde dua, dna orde lebih tinggi. GLCM merupakan teknik ekstraksi fitur tekstur orde dua, yang berarti GLCM mempertimbangkan hubungan antara kelompok dua piksel dalam citra asli (Mohanaiah, Sathyanarayana, and GuruKumar, 2013).

Hubungan spasial antara dua piksel tetangga dapat ditemukan dengan banyak cara melalui *offset* dan sudut berbeda, umumnya melalui piksel dan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tetangga terdekatnya di sebelah kanan. Kemungkinan sudut yang dapat diimplementasikan yaitu $0°, 45°, 90°, 135°$ (Hall-Beyer, 2017). GLCM merupakan suatu matriks yang terdiri dari baris dan kolom sebagai representasi jumlah tingkat keabu-abuan suatu citra. Elemen suatu matriks $P(i, j \mid \triangle x \triangle y)$ merupakan frekuensi relatif dari dua piksel yang dipisahkan oleh jarak piksel $(\triangle x, \triangle y)$ dalam ketertanggaan tertentu, satu dengan intensitas i dan yang lain dengan intensitas j. Elemen matriks $P(i, j|d, \theta)$ berisi nilai probabilitas statistik orde dua untuk perubahan tingkat abu-abu i dan j pada jarak perpindahan d_p dan sudut θ (Mohanaiah, Sathyanarayana, and GuruKumar, 2013). Secara umum, jarak perpindahan piksel yang digunakan adalah $d_p = 1$. Tekstur GLCM bekerja dalam hubungan antara dua piksel yang disebut piksel referensi dan piksel tetangga. Hubungan ini dapat dinyatakan sebagai hubungan (1,0), dimana 1 merupakan piksel dalam arah x, 0 piksel dalam arah y (Hall-Beyer, 2017).



Gambar 2.8 Empat Sudut dalam GLCM

(Sebastian, Unnikrishnan, and Balakrishnan, 2012)

Berikut adalah tahapan dalam proses ekstraksi fitur menggunakan GLCM (Satrio, 2016):

Membuat area kerja dari matriks citra

Citra yang digunakan harus dalam bentuk citra *grayscale*, sebagai contohnya, Gambar 2.9 merupakan hasil dari matriks citra *grayscale* yang akan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dilakukan perhitungan ekstraksi teksturnya, dan Gambar 2.10 merupakan area kerja dari matriks citra.

0	0	1	1
0	0	1	1
0	2	2	2
2	2	3	3

Gambar 2.9 Matriks Citra *Grayscale* derajat 8

(Satrio, 2016)

	0	1	2	3
0				
1				
2				
3				

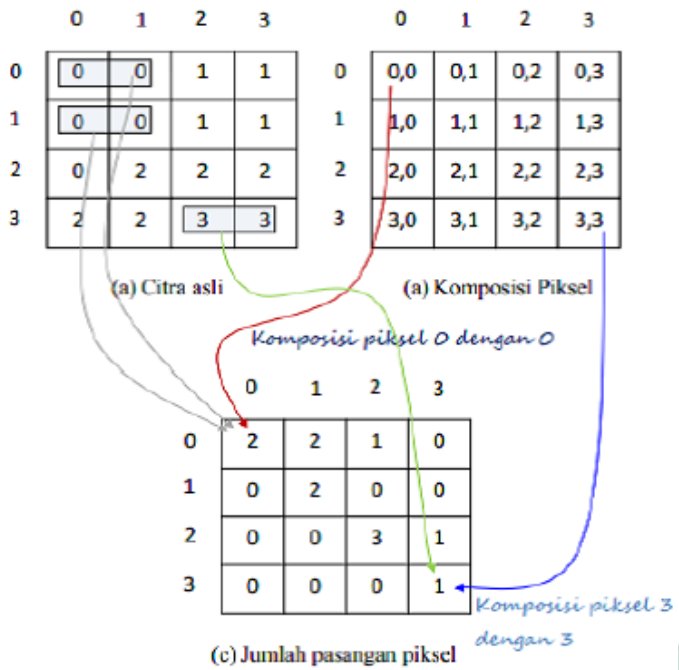
Gambar 2.10 Area Kerja untuk Matriks Citra

(Satrio, 2016)

1. Menentukan hubungan spasial antara piksel referensi dan piksel tetangga dengan sudut θ dan jarak d_p. Dalam kasus ini, digunakan θ sebesar $0°$ dengan jarak $d_p = 1$

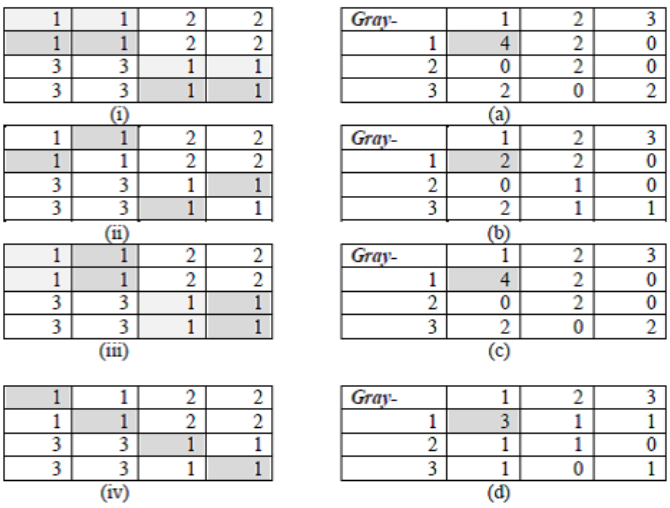
2. Menghitung jumlah *co-occurrence* dan mengisikannya pada area kerja matriks seperti ilustrasi pada Gambar 2.11 dan Gambar 2.12. Gambar 2.11 menunjukkan jumlah *co-occurrence* pada sudut 0° dan Gambar 2.12 adalah hasil dari implementasi seluruh sudut pada GLCM.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.11 Ilustrasi Menghitung Jumlah *Co-Occurrence*

(Satrio, 2016)

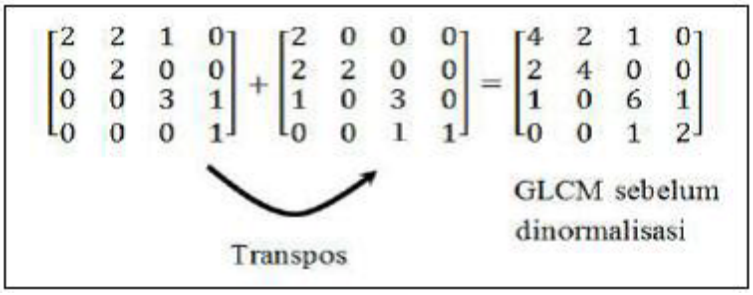


Gambar 2.12 Ilustrasi Menghitung Jumlah *Co-Occurrence* pada Beberapa Sudut (i)-a Sudut $0°$**,(ii)-b Sudut** $45°$**,(iii)-c Sudut** $90°$**,(iv)-d Sudut** $135°$

(Satrio, 2016)

3. Menjumlahkan matriks *co-occurrence* dengan transposenya agar menjadi simetris.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Gambar 2.13 Penjumlahan Matriks *Co-Occurrence* dengan Transposenya

(Satrio, 2016)

4. Melakukan normalisasi (probabilitas) matriks. Normalisasi matriks dilakukan untuk mengubahnya ke bentuk probabilitas. Normalisasi ini berfungsi untuk mengurangi jangkauan antar piksel. Persamaan peluang antar piksel pada matriks *co-occurrence* didefinisikan dalam Persamaan 2.7 (Hall-Beyer, 2017)

$$P(i,j) = \frac{f(i,j)}{\sum_i \sum_j f(i,j)} \tag{2.7}$$

Sehingga hasil dari normalisasi matriks *co-occurrence* adalah:

$$\begin{pmatrix} \frac{4}{24} & \frac{2}{24} & \frac{1}{24} & \frac{0}{24} \\ \frac{2}{24} & \frac{4}{24} & \frac{0}{24} & \frac{0}{24} \\ \frac{1}{24} & \frac{0}{24} & \frac{6}{24} & \frac{1}{24} \\ \frac{0}{24} & \frac{0}{24} & \frac{1}{24} & \frac{2}{24} \end{pmatrix}$$

5. Menghitung nilai fitur-fitur ekstraksi dari normalisasi yang diperoleh. Berikut beberapa fitur dalam GLCM yang akan dihitung dalam penelitian ini (Gadkari, 2000).

   a. Energi

      Statistik ini dapat disebut sebagai keseragaman atau ASM (*Angular*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Second Moment). Energi mengukur keseragaman dari tekstur yang merupakan pengulangan pasangan piksel. Energi dapat mendeteksi gangguan pada tekstur. Nilai maksimum yang dapat dicapai energi yaitu satu. Nilai energi tinggi terjadi kektika distribusi tingkat keabuan memiliki bentuk konstan atau periodik. Berikut ini merupakan persamaan untuk mendapatkan fitur energi dari GLCM (Novitasari, et al., 2019):

$$\text{Energi} = \sum_{i=1}^{N} \sum_{j=1}^{N} P(i,j)^2 \tag{2.8}$$

dengan N merupakan banyaknya nilai keabuan suatu citra.

b. Kontras

Fitur statistik ini mengukur frekuensi spasial dari suatu citra dan perbedaan momen dari GLCM. Fitur kontras menunjukkan perbedaan antara nilai piksel terendah dan tertinggi dari serangkaian piksel yang berdekatan. Fitur ini juga mengukur jumlah variasi lokal yang ada dalam citra. Citra dengan kontras yang rendah merepresentasikan konsentrasi GLCM di sekitar diagonal utama dan menunjukkan frekuensi spasial yang rendah. Berikut adalah persamaan untuk mendapatkan fitur kontras (Novitasari, et al., 2019):

$$\text{Kontras} = \sum_{i=1}^{N} \sum_{j=1}^{N} |i-j|^2 P(i,j) \tag{2.9}$$

c. Homogenitas

Statistik ini juga dapat disebut sebagai *Inverse Difference Moment*. Fitur ini mengukur homogenitas citra dengan mengasumsikan nilai homogenitas akan besar jika ragam keabuan pada citra mengecil. IDM memiliki nilai yang maksimum jika semua elemen pada citra sama. Namun

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

jika elemen pada suatu citra berbeda, maka IDM tidak bernilai amksimum. Berikut persamaan untuk mendapatkan nilai fitur IDM (Novitasari, et al., 2019):

$$\text{Homogenitas} = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{P(i,j)}{1+(i-j)^2} \tag{2.10}$$

d. Korelasi

Korelasi menunjukkan ukuran struktur linear tingkat abu-abu pada piksel tetangga dalam citra. Berikut persamaan untuk mendapatkan fitur korelasi (Novitasari, et al., 2019):

$$\text{Korelasi} = \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i\sigma_j} \tag{2.11}$$

dengan

$$\mu_i = \sum_{i=1}^{N}\sum_{j=1}^{N} i(P(i,j)) \tag{2.12}$$

$$\mu_j = \sum_{i=1}^{N}\sum_{j=1}^{N} j(P(i,j)) \tag{2.13}$$

$$\sigma_i = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)(i-\mu_i)^2} \tag{2.14}$$

$$\sigma_j = \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)(j-\mu_j)^2} \tag{2.15}$$

$$\tag{2.16}$$

2.5. *K-fold Cross Validation*

Cross Validation merupakan proses *learning* dengan menggunakan semua data yang tersedia sebagai contoh pelatihan dan pengujian. Metode ini didasarkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

pada penggunaan data data pelatihan dan data pengujian secara berulang kali sebanyak K kali. Dalam metode ini, kumpulan data dibagi menjadi k lipatan. Proses *training* digunakan dengan $k - 1$ *folds*. Untuk menghitung nilai eror klasifikasi digunakan data testing pada *fold* yang tersisa (Bengio and Grandvalet, 2004). Pada *K-fold Cross Validation* proses dilakukan secara acak sehingga akurasi rata-rata yang dihasilkan dari metode ini pada data set tidak konstan. Oleh karena itu, dampak dari prngacakan ini dapat diatasi dengan bias dan varians. Bias estimasi akurasi akan lebih kecil jika jumlah *folds* sebanyak lima atau sepuluh. Empat faktor yang mempengaruhi akurasi dari *K-fold Cross Validation* (Wong, 2015):

1. Jumlah *folds*
2. Jumlah *instance* dalam satu lipatan
3. Tingkat rata-rata
4. Pengulangan *K-fold Cross Validation*

2.6. *Firefly Algorithm* (FFA)

Firefly merupakan algoritma metaheuristik yang digunakan untuk optimasi. Algoritma metaheuristik biasanya terinspirasi oleh alam dengan banyak agen yang berinteraksi. Sekelompok metaheuristik biasanya disebut dengan algoritma berbasis *Swarm Intelligence* (SI). Algoritma ini telah berkembang dengan menirukan karakteristik dari SI seperti burung, ikan, manusia, maupun lainnya. Dalam dekade terakhir, banyak algoritma baru yang muncul dari sekolompok metaheuristik berbasis SI, seperti *Particle Swarm Optimization* (PSO), *Differential Evolution*, *Bat Algorithm*, *Firefly Algorithm*, dan *Cuckoo Search*. Seluruh

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

algoritma tersebut dapat menunjukkan kinerjanya yang baik dalam menyelesaikan masalah optimasi rekayasa yang sulit. Namun, dari jajaran algoritma tersebut, *Firefly* (FFA) merupakan algoritma yang sangat efisien dalam menangani masalah optimasi global multimodal. Intensitas cahaya pada kunang-kunang tergantung dari jarak mata yang melihatnya. Cahaya akan semakin besar jika jarak antara kunang-kunang semakin dekat, begitupun sebaliknya (Yang and He, 2013).

FFA merupakan suatu algoritma yang terinspirasi dari perilaku kunang-kunang. Kunang-kunang merupakan serangga yang menghasilkan cahaya pendek dan berirama hasil dari proses *bioluminescence*. Fungsi dari cahaya ini adalah untuk menarik pasangan dalam berkomunikasi atau menarik mangsa, dan sebagai peringatan terhadap adanya predator. Oleh sebab itu, cahaya pendek ini menjadi faktor kunang-kunang lainnya untuk bergerak menuju ke arah kunang-kuanng yang lain yang lebih terang (Ali, Othman, Husain, and Misran, 2014). Tiga aturan sebagai algoritma dalam FFA, yaitu (Johari, Zain, Mustaffa, and Udin, 2013):

1. Kunang-kunang merupakan kawanan unisex, sehingga satu kunang-kunang akan tertarik pada kunang-kunang lainnya tanpa memandang jenis kelamin mereka.

2. Daya tarik kunang-kunang sebanding dengan tingkat kecerahan, sehingga kunang-kunang yang lebih terang akan menarik kunang-kunang lainnya. Daya tarik akan berkurang jika jarak antara kunang-kunang meningkat. Apabila tidak ada kunang-kunang yang memiliki kecerahan paling cerah, maka kunang-kunang akan bergerak secara acak.

3. Kecerahan kunang-kunang ditentukan oleh fungsi objektif dari permasalahan yang sedang dipertimbangkan. Kecerahan dapat sebanding dengan nilai fungsi objektif jika masalah merupakan kasus maksimalisasi. Namun, kecerahan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menjadi berbanding terbalik dengan nilai fungsi objektif jika masalah merupakan kasus minimalisasi.

FFA telah banyak diterapkan dalam beberapa kasus penelitian. FFA telah banyak diterapkan untuk menyelesaiakan fungsi kontinu dalam matematika. Peneliti lainnya juga mengatakan bahwa FFA sangat efisien dan dapat mengungguli algoritma konvensional lainnya berdasarkan kinerja statistik yang diukur menggunakan uji stokastik standar. FFA dapat menemukan global dan lokal optimal secara bersamaan karena bekerja berdasar komunikais global di antara kunang-kunang (Johari, Zain, Mustaffa, and Udin, 2013).

Daya tarik suatu kunang-kunang didasarkan pada tingkat kecerahan cahanya. Misalkan terdapat n kunang-kunang dan x_i merupakan solusi untuk kunang-kunang i. Kecerahan dari kunang-kunang i dikaitkan dengan fungsi objektif $f(x_i)$. Oleh karena itu nilai I yang merupakan kecerahan kunang-kunang pada posisi terakhir dari fungsi objektif didapatkan dengan nilai akan sebanding jika kasusnya maksimasi dan akan berbanding terbalik jika kasusnya minimalisasi. Berikut adalah Persamaan yang digunakan untuk nilai maksimasi dan minimaliasi:
Kasus Maksimasi

$$I_i = f(x_i) \tag{2.17}$$

Kasus Minimasi

$$I_i = \frac{1}{f(x_i)} \tag{2.18}$$

I pada jarak tertentu antara kunang-kunang i dan kunang-kunang j. Maka, nilai I

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dapat dicari dengan menggunakan Persamaan 2.19

$$I(r) = \frac{I_s}{r^2} \tag{2.19}$$

I_s merupakan intensitas dari sumbernya. Intensitas cahaya I bervariasi dengan jarak r yang memiliki koefisien serapan cahaya γ, sehingga

$$I = I_o e^{-\gamma r} \tag{2.20}$$

dimana I_o adalah intensitas cahaya di titik awal. Untuk menghindari singularitas pada $r = 0$ dalam $\frac{I_s}{r^2}$ dari efek kombinasi dari hukum kuadrat terbalik dan penyerapan, dapat diperkirakan dalam bentuk Gaussian berikut:

$$I = I_o e^{-\gamma r^2} \tag{2.21}$$

Daya tarik kunang-kunang β sebanding dengan intensitas cahaya yang terlihat oleh kunang-kunang terdekat dan dapat didefinisikan sebagai berikut:

$$\beta = \beta_o e^{-\gamma r^2} \tag{2.22}$$

dimana, β_o adalah daya tarik pada $r = 0$. Karena menghitung dengan $\frac{1}{1+r^2}$ lebih cepat daripada bentuk eksponensial, maka Persamaan 2.22 diubah dalam bentuk pembagian atau pecahan agar mempermudah perhitungan seperti berikut:

$$\beta = \frac{\beta_o}{1+\gamma r^2} \tag{2.23}$$

Persamaan 2.23 mendefinisikan jarak karakteristik $\Gamma\frac{1}{\gamma}$ dimana, daya tarik

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

kunang-kunang berubah secara signifikan dari β_o ke $\beta_o e^{-1}$. Dalam implementasinya, bentuk dari fungsi penarik $\beta(r)$ dapat berupa fungsi yang penurunan secara monoton seperti pada Persamaan 2.24

$$\beta(r) = \beta_o e^{-\gamma r^m} (m \geq 1) \tag{2.24}$$

untuk penetapan, nilai panjang karakteristik menjadi:

$$\Gamma = \gamma^{\frac{-1}{m}} \to 1, m \to \infty \tag{2.25}$$

sebaliknya, untuk skala panjang tertentu Γ dalam masalah optimasi, parameter γ dapat digunakan sebagai nilai awal yang khas, yaitu:

$$\gamma = \frac{1}{r^m} \tag{2.26}$$

Jarak antara dua kunang-kunang dapat dihitung menggunakan metode jarak Cartesian.

$$r_{i,j} = \|x_i - x_j\| = \sqrt{\sum_{k=1}^{d} (x_{i,k} - x_{j,k})^2} \tag{2.27}$$

$x_{i,k}$ adalah komponen ke-k dari koordinat spasial x_i yang merupakan kunang-kunang ke-i. Dalam kasus 2D, dapat digunakan:

$$r_{i,j} = \sqrt{(x_i - x_j)^2 - (y_i - y_j)^2} \tag{2.28}$$

Pergerakan kunang-kunang ke-i yang tertarik pada kunang-kunang ke-j yang lebih

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

terang ditentukan oleh:

$$x_i = x_i(1 - \beta_o e^{\gamma r_{i,j}^2}) + xo_j(\beta_o e^{\gamma r_{i,j}^2}) + \alpha\left(rand - \frac{1}{2}\right) \tag{2.29}$$

α merupakan parameter pengacakan, x_i adalah kunang-kunang setelah disortir, xo_j adalah kunang-kunang sebelum disortir, $rand$ adalah generator angka acak yang didistribusikan secara seragam dalam rentang $[0, 1]$. Secara umum, dapat digunakan $\beta_o = 1$ dan $\epsilon \in [0, 1]$. Selanjutnya proses pengacakan dapat diperluas ke distribusi normal $N(0, 1)$ atau distribusi lainnya. Parameter γ mencirikan suatu daya tarik dan nilainya sangat penting dalam menentukan kecepatan konvergensi. Secara teori, $\gamma \in [0, \infty]$, tapi secara umum dalam pengaplikasiannya sangat bervariasi dari 0,01 hingga 100 (Arora and Singh, 2013).

Proses dari *Firefly Alorithm* yaitu (Perdana, 2017):

1. Inisialisasi populasi kunang-kunang, jumlah iterasi, dan parameter dari FFA.
2. Hitung intensitas pada setiap kunang-kunang.
3. Pindahkan masing-masing kunang-kunang menuju kunang-kunang dengan tingkat kecerahan paling cerah, jika tidak ada kunang-kunang paling cerah, maka kunang-kunang dipindahkan secara acak.
4. Perbarui *solution set* atau fbest setelah dilakukan pergerakan pada posisi/titik terbaru.
5. Lakukan sampai mendapatkan kunang-kunang dengan nilai fitnes terbaik atau sampai iterasi berhenti.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.7. *Support Vector Machine* (SVM)

SVM merupakan teknik pembelajaran terawasi yang digunakan untuk menyelesaikan masalah klasifikasi data dan regresi. SVM memiliki kecepatan belajar dalam data yang besar dan dapat mengklasifikasikan dua atau lebih kelas data (Parveen and Singh , 2015). SVM menggunakan teori pembelajaran mesin untuk memaksimalkan akurasi predikitif yang secara otomatis menghindari *over-fit* data. Mesin SVM dapat didefinisikan sebagai sistem yang menggunakan ruang hipotesis fungsi linear dalam ruang fitur dimensi tinggi dengan memasukkan konsep kernel, dilatih dengan algoritma pembelajaran dari teori optimasi yang mengimplementasikan bias pembelajaran dari teori pembelajaran statistik (Jakkula).

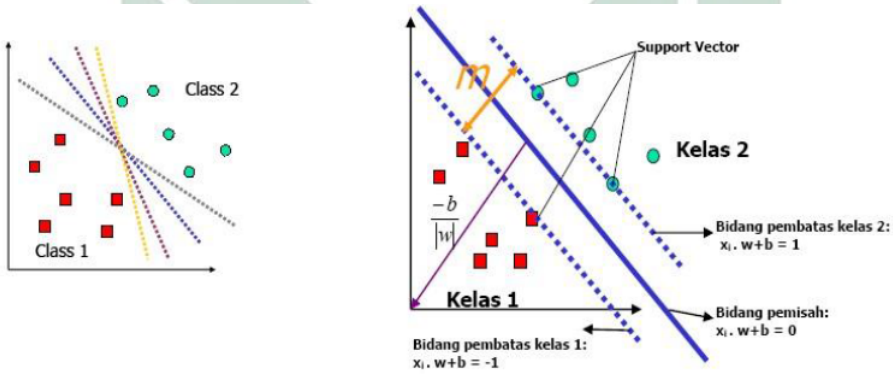
Support Vector Machine (SVM) diperkenalkan pertama kali oleh Guyon, Boser, dan Vapnik pada tahun 1992 di Annual Workshop on Computational Learning Theory. Konsep dasar SVM yaitu kombinasi harmonis dari teori-teori komputasi yang telah ada, seperti *margin hyperplane*, kernel, dan konsep pendukung lainnya (Nugroho, Witarto, and Handoko, 2003). Berbeda dengan strategi *neural network* yang berusaha mencari *hyperplane* pemisah antar kelas, SVM berusaha menemukan *hyperplane* yang terbaik pada *input space*. *Hyperplane* dikatakan optimal jika jarak antara *hyperplane* dengan data terdekat dari masing-masing kelas sudah optimal (Foeady, Novitasari, and Asyhar, 2018).

SVM telah berhasil diaplikasikan dalam dunia nyata dan secara umum memberikan solusi yang lebih baik dibandingkan dengan metode konvensional, seperti *artificial neural network*. SVM telah terbukti sukses diaplikasikan oleh Vapnik dalam masalah klasifikasi dan estimasi fungsi dengan memisahkan data pelatihan menjadi dua kelas (Adinegoro, Atmaja dan Purnamasari, 2015).

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

2.7.1. Klasifikasi SVM

Strategi SVM dalam menentukan *hyperplane* terbaik berfungi untuk memisahkan dua buah kelas dalam *input space* (Foeady, Novitasari, and Asyhar, 2018). Pada Gambar 2.14 menunjukkan bahwa terdapat dua pola yang merupakan anggota dari dua buah kelas, yaitu +1 dan -1. Pola yang tergabung dalam kelas -1 disimbolkan dengan kotak berwarna merah, sedangkan kelas +1 disimbolkan dengan lingkaran berwarna hijau. Pada Gambar 2.14 terlihat berbagai alternatif bidang pemisah yang dapat memisahkan semua data set sesuai dengan kelasnya. Gambar sebelah kiri merupakan alternatif bidang pemisah sesuai kelasnya, sedangkan gambar sebelah kanan merupakan bidang pemisah terbaik *optimal hyperplane* dengan *margin* terbesar. *Margin* merupakan jarak antara *hyperplane* dengan pola terdekat dari masing-masing kelas. Adapun data yang terletak pada bidang pembatas disebut dengan *support vector* (Adinegoro, Atmaja dan Purnamasari, 2015).



Gambar 2.14 SVM Berusaha Menemukan *Hyperplane* Terbaik

(Adinegoro, Atmaja dan Purnamasari, 2015)

Hyperplane sebagai pemisah terbaik anatara dua kelas dapat ditemukan dengan mengukur *margin* kelas tersebut dna mencari titik maksimalnya. Misalkan terdapat N data training, dimana setiap inputan x_i memiliki atribut D (berdimensi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

D) dan berada di salah satu dari dua kelas, $y_i = -1$ atau $y_i = +1$ dalam bentuk:

$$\{x_i, y_i\} \text{ dimana } i = 1, ...N, y_i \in \{-1, 1\}, x \in \Re^D$$

Data dalam permasalahn ini diasumsikan dapat dipisahkan secara linear. Sebuah *hyperplane* dapat didefinisikan dalam Persamaan 2.30.

$$x.w + b = 0 \tag{2.30}$$

Dengan w merupakan bobot dari SVM yang berupa vektor kolom dan b suatu skalar. Bidang pemisah (*separating hyperplane*) akan memisahkan data ke dalam dua kelas positif dan negatif. Data positif akan berada di sebelah kanan *hyperplane*, sedangkan data negatif berada di sebelah kiri *hyperplane*. Data dipisahkan dengan ketentuan sebagai berikut:

$$x_i.w + b \geq +1 \quad \text{untuk} \quad y_i = +1 \tag{2.31}$$

$$x_i.w + b \leq -1 \quad \text{untuk} \quad y_i = -1 \tag{2.32}$$

Selanjutnya persamaan dapat digabungkan menjadi

$$y_i(x_i.w + b) - 1 \geq 0 \ \ \forall i \tag{2.33}$$

Jika terdapat tiga *hyperplane*, yaitu *hyperplane* utama yang membagi tepat ke tengah, *hyperplane* kedua (H1) yang melalui *support vector* pada kelas positif (+1), dan *hyperplane* ketiga yang melalui *support vector* pada kelas negatif (-1). Maka *hyperplane* $H1$ dan $H2$ dapat didefinisikan sebagai berikut (Fletcher, 2008):

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

s

$$x_i.w + b = +1 \text{ untuk } H1 \tag{2.34}$$

$$x_i.w + b = -1 \text{ untuk } H2 \tag{2.35}$$

Bidang $H1$ dan $H2$ sejajar dengan *hyperplane* utama, maka kedua bidang tersebut disebut sebagai *supporting hyperplane* dengan jarak $H1$ terhadap *hyperplane* utama adalah d^+ dan jarak $H2$ terhadap *hyperplane* utama yaitu d^-. Maka jarak $H1$ terhadap *hyperplane* utama adalah:

$$d^+ = \|x\| = \sqrt{x'x} = \sqrt{\frac{(1-b)^2}{w'w}} = \frac{1-b}{\|w\|} \tag{2.36}$$

Untuk menghitung jarak $H2$ terhadap *hyperplane* utama, dapat dihitung dengan meminimalkan $x'x$ dan memperhatikan persamaan berikut:

$$x_i.w + b \leq -1$$
$$\text{min } x'x + \lambda(w'x + b + 1)$$

Selanjutnya diturunkan untuk mendapatkan nilai minimal

$$\frac{d}{dx} = 0 = 2x + \lambda w = 0$$
$$x = -\frac{\lambda}{2}w$$

Disubtitusikan ke dalam Persamaan 2.35, sehingga diperoleh:

$$\begin{aligned} -\frac{\lambda}{2}w'w + b &= -1 \\ \lambda &= \frac{2(b+1)}{w'w} \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Subtitusi kembali λ terhadap x, sehingga:

$$\begin{aligned} x &= \frac{(-b-1)}{w'w}w \\ x'x &= \frac{((-b-1)^2)}{(w'w)^2}w'w \\ x'x &= \frac{(-b-1)^2}{w'w} \end{aligned}$$

Maka jarak $H2$ terhadap *hyperplane* utama adalah:

$$d^- = \|x\| = \sqrt{x'x} = \sqrt{\frac{(-b-1)^2}{w'w}} = \frac{-b-1}{\|w\|} \tag{2.37}$$

Margin maksimum dapat dihitung dengan memaksimalkan jarak antara $H1$ dan $H2$ sebagai *supporting hyperplane* menggunakan Persamaan 2.38.

$$|d^+ - d^-| = \left|\frac{(1-b)}{|w|} - \frac{(-b-1)}{|w|}\right| = \frac{2}{\|w\|} \tag{2.38}$$

Dalam hal ini, jika $\|w\|$ diminimalkan, maka *margin* yang didapatkan akan maksimal, sedangkan jika $\|w\|$ dimaksimalkan, maka *margin* yang didapatkan akan minimal. Cara yang dapat digunakan untuk meminimalkan $\|w\|$ yaitu menurunkan sebuah fungsi yang memuat $\|w\|$ terhadap variabel $\|w\|$. Agar nilai w berpengaruh saat diminimumkan tanpa menghilangkan sifat dari $\|w\|$, maka digunakan pemisalan dengan mengganti meminimumkan $\|w\|^2$ (Adyanti, 2018). Oleh karena itu diperoleh persamaan fungsi sebagai berikut:

$$\min \frac{1}{2}\|w\|^2$$
$$y_i(x_i.w + b) \geq 1, \forall i$$

Untuk memenuhi kendala dalam minimalisasi kasus di atas, maka perlu digunakan pengali Lagrange α, dimana $\alpha \geq 0, \forall i$. Oleh karena itu, model permasalahn diubah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menjadi:

$$\begin{aligned}
\min L_p &= \frac{1}{2}\|w\|^2 + \sum_{i=1}^{N} \alpha_i(1 - y_i(x_i.w + b)) \\
&= \frac{1}{2}\|w\|^2 - \sum_{i=1}^{N} \alpha_i y_i(x_i.w + b) + \sum_{i=1}^{N} \alpha_i
\end{aligned}$$

Untuk meminimumkan L_p digunakan $\frac{\partial L_p}{\partial b} = 0$ dan $\frac{\partial L_p}{\partial w} = 0$. Oleh karena itu dihasilkan bentuk persamaan sebagai berikut:

$$\begin{aligned}
\frac{\partial L_p}{\partial b} &= \sum_{i=1}^{N} \alpha_i y_i = 0 \\
\frac{\partial L_p}{\partial w} &= w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \longrightarrow w = \sum_{i=1}^{N} \alpha_i y_i x_i
\end{aligned}$$

Hasil minimum diperoleh dari turunan parsial L_p terhadap variabel yang ada dengan dua variabelnya diubah ke masalah dual, sehingga hanya satu variabel yang tidak diketahui, yaitu α_i. Perubahan ke masalah dual menyebabkan fungsi minimum menjadi fungsi maksimum seperti perhitungan di bawah ini.

$$\begin{aligned}
\text{maks } L_p &= \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i x_i\right)\left(\sum_{i=1}^{N} \alpha_i y_i x_i\right) \\
&\quad - \sum_{i=1}^{N} \alpha_i y_i \left(x_i \left(\sum_{i=1}^{N} \alpha_i y_i x_i\right) + b\right) + \sum_{i=1}^{N} \alpha_i \\
\text{maks } L_p &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i.x_j) - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i.x_j) \\
&\quad - b\sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i
\end{aligned}$$

dengan perhitunga di atas, maka diperoleh nilai maks L_p untuk data yang dapat

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dipisahkan secara linear sebagai berikut:

$$\text{maks } L_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i x_i \alpha_j y_j x_j \tag{2.39}$$

dengan syarat $\sum_{i=1}^{N} \alpha_i y_i = 0$ dengan $\alpha_i \geq 0$ dan $i = 1, 2, ..., N$

Persamaan 2.39 hanya dapat diterapkan dalam kasus data yang dapat dipisahkan secara linear. Sedangkan untuk data yang tidak dapat dipisahkan secara linear, dapat ditambahkan variabel slack $\varepsilon_i \geq 0$ ke dalam pertidaksamaan $y_i(x_i.w + b) \geq 1$. Oleh karena itu fungsi tujuan dan fungsi kendala menjadi:

$$y_i(x_i.w + b) - 1 + \varepsilon_i \geq 0$$

Didefinisikan sebuah parameter baru C sebagai batasan antara memaksimalkan *margin* dan mengurangi jumlah data yang salah ketika diklasifikasikan. Oleh karena itu, model masalah menjadi:

$$\text{min } \frac{1}{2}\|w\|^2 + C \textstyle\sum_{i=1}^{N} \varepsilon_i$$

dengan $y_i(x_i.w + b) - 1 + \varepsilon_i \geq 0, \varepsilon_i \geq 0, \forall_i$. Oleh karena itu permasalahan meminimumkan dengan menggunakan pengali Lagrange menjadi:

$$\begin{aligned}
\text{min } L_p &= \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \varepsilon_i + \sum_{i=1}^{N} \alpha_i (1 - y_i(x_i.w + b) - \varepsilon_i) \\
&\quad + \sum_{i=1}^{N} q_i(-\varepsilon_i) \\
&= \frac{1}{2}\|w\|^2 + C \sum_{i=1}^{N} \varepsilon_i + \sum_{i=1}^{N} \alpha_i y_i (x_i.w + b) + \sum_{i=1}^{N} \alpha_i \\
&\quad - \sum_{i=1}^{N} \alpha_i \varepsilon_i - \sum_{i=1}^{N} q_i \varepsilon_i
\end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Untuk meminimumkan L_p digunakan $\frac{\partial L_p}{\partial b} = 0$,$\frac{\partial L_p}{\partial w} = 0$, dan $\frac{\partial L_p}{\partial \varepsilon_i} = 0$ Sehingga dihasilkan bentuk persamaan sebagai berikut:

$$\begin{aligned}
\frac{\partial L_p}{\partial b} &= \sum_{i=1}^{N} \alpha_i y_i = 0 \\
\frac{\partial L_p}{\partial w} &= w - \sum_{i=1}^{N} \alpha_i y_i x_i = 0 \longrightarrow w = \sum_{i=1}^{N} \alpha_i y_i x_i \\
\frac{\partial L_p}{\partial \varepsilon_i} &= C - \alpha_i - q_i = 0
\end{aligned}$$

Akibatnya $C \sum_{i=1}^{N} \varepsilon_i - \sum_{i=1}^{N} \alpha_i \varepsilon_i - \sum_{i=1}^{N} q_i \varepsilon_i = 0$. Dengan menggunakan turunan parsial, masalah dualnya menjadi:

$$\begin{aligned}
\text{maks } L_p &= \frac{1}{2}\left(\sum_{i=1}^{N} \alpha_i y_i x_i\right)\left(\sum_{i=1}^{N} \alpha_i y_i x_i\right) + C\sum_{i=1}^{N} \varepsilon_i \\
&\quad - \sum_{i=1}^{N} \alpha_i y_i \left(\left(\sum_{i=1}^{N} \alpha_i y_i x_i\right).x_i + b\right) \\
&\quad + \sum_{i=1}^{N} \alpha_i - \sum_{i=1}^{N} \alpha_i \varepsilon_i - \sum_{i=1}^{N} q_i \varepsilon_i \\
\text{maks } L_p &= \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i.x_j) - \sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i.x_j) \\
&\quad - b\sum_{i=1}^{N} \alpha_i y_i + \sum_{i=1}^{N} \alpha_i + C\sum_{i=1}^{N} \varepsilon_i - \sum_{i=1}^{N} \alpha_i \varepsilon_i - \sum_{i=1}^{N} q_i \varepsilon_i \\
\text{maks } L_p &= \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i.x_j)
\end{aligned}$$

dengan syarat $\sum_{i=1}^{N} \alpha_i y_i = 0$ karena $\frac{\partial L_p}{\partial \varepsilon_i} = C - \alpha_i - q_i$ (Fletcher, 2008)

Nilai q pada fungsi tujuan tidak diperhitungkan, sehingga terdapat kendala baru $0 \leq \alpha_i \leq C$. Adanya bias yang telah didefinisikan sebagai b menyebabkan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

adanya kendala:

$$\sum_{i=1}^{N} \alpha_i y_i = 0$$

Sehingga data input $x_i = [x_{i1}, x_{i2}, ..., x_{im}]$ menjadi $x'_i = [x_{i1}, x_{i2}, ..., x_{im}\lambda]$ dan vektor bobot w menjadi $w' = \left[w_1, w_2, ..., w_m \frac{b}{\lambda}\right]$ dengan λ merupakan konstanta. Bentuk masalah menjadi:

$$\begin{aligned} L'_p = \min\ L_p &= \frac{1}{2}\|w'\|^2 + C\sum_{i=1}^{N}\varepsilon_i + \sum_{i=1}^{N}\alpha_i(1 - y_i(x'_i.w') - \varepsilon_i) \\ &\quad + \sum_{i=1}^{N} q_i(-\varepsilon_i) \end{aligned}$$

Untuk meminimalkan L_p digunakan $\frac{\partial L_p}{\partial w'} = 0$ dan $\frac{\partial L_p}{\partial \varepsilon_i} = 0$, maka persamaan menjadi:

$$\begin{aligned} \frac{\partial L_p}{\partial w'} &= w' - \sum_{i=1}^{N}\alpha_i y_i x'_i = 0 \rightarrow w' = \sum_{i=1}^{N}\alpha_i y_i x'_i \\ \frac{\partial L_p}{\partial \varepsilon_i} &= C - \alpha_i - q_i = 0, \forall_i \\ \text{maks } L_p &= \frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i x'_i\right)\left(\sum_{i=1}^{N}\alpha_i y_i x'_i\right) + C\sum_{i=1}^{N}\varepsilon_i \\ &\quad - \sum_{i=1}^{N}\alpha_i\left(1 - y_i\left(\sum_{i=1}^{N}\alpha_i y_i x'_i.x'_i\right) - \varepsilon_i\right) + \sum_{i=1}^{N} q_i(-\varepsilon_i) \\ \text{maks } L_p &= \frac{1}{2}\left(\sum_{i=1}^{N}\alpha_i y_i x'_i\right)\left(\sum_{i=1}^{N}\alpha_i y_i x'_i\right) + C\sum_{i=1}^{N}\varepsilon_i \\ &\quad - \sum_{i=1}^{N}\alpha_i\left(y_i\left(\sum_{i=1}^{N}\alpha_i y_i x'_i.x'_i\right)\right) + \sum_{i=1}^{N}\alpha_i - C\sum_{i=1}^{N}\varepsilon_i \\ &\quad - \sum_{i=1}^{N}\alpha_i\varepsilon_i - \sum_{i=1}^{N} q_i\varepsilon_i \\ \text{maks } L_p &= \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\alpha_i y_i x'_i \sum_{i=1}^{N}\alpha_i y_i x'_i \end{aligned}$$

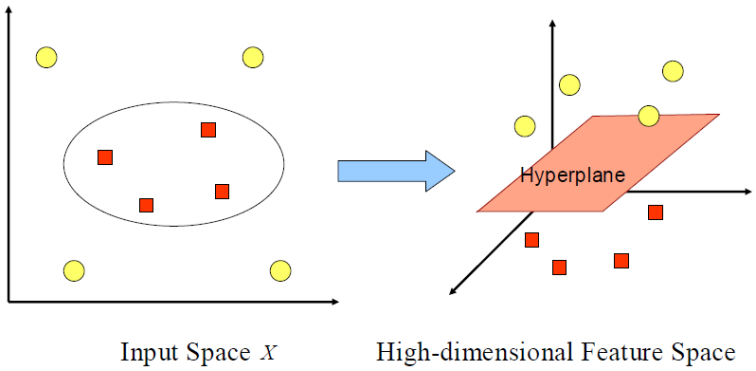
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Maka bentuk akhir dari maks L_p menjadi (Adyanti, 2018):

$$\text{maks } L_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i'.x_j') \tag{2.40}$$

2.7.2. Kernel

Ide dasar SVM yaitu mengklasifikasikan data secara linear. Akan tetapi, salah satu kendala dalam proses klasifikasi yaitu dispersi data yang cenderung beragam. Oleh karena itu, data akan sulit dipisahkan secara linear. Dalam hal ini, SVM memperkenalkan fungsi kernel yang dapat mengubah ruang data asli menjadi ruang baru dalam dimensi yang lebih tinggi. Tujuannya yaitu untuk memudahkan pemisahan data yang telah ditransformasikan pada dimensi yang lebih tinggi (Nanda, et al, 2018). Perubahan ruang data menjadi dimensi yang lebih tinggi dapat dilihat dalam Gambar 2.15.



Gambar 2.15 Transformasi Data SVM Menggunakan Fungsi Kernel

(Nugroho, Witarto, and Handoko, 2003)

Pemetaan data menggunakan fungsi kernel dilakukan dengan menjaga topologi data, dimana dua data yang berjarak dekat pada *input space* akan berjarak dekat juga pada *feature space*. Sebaliknya, dua data yang berjarak jauh pada *input space* akan berjarak jauh juga pada *feature space*. Selanjutnya proses

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

pembelajaran pada SVM dalam menemukan titik-titik *support vector* hanya bergantung pada *dot product* dari data yang sudah ditransformasikan pada ruang baru yang berdimensi lebih tinggi (Nugroho, Witarto, and Handoko, 2003).

Dasar dari kernel yaitu memetakan data pada ruang dimensi yang lebih tinggi menggunakan fungsi pemetaan ϕ, dengan perkalian fungsinya sebagai berikut:

$$(\phi(x_i).\phi(x_j)) = K(x_i, x_j)$$

Kernel $K(x_i, x_j)$ merupakan fungsi yang memenuhi kondisi berikut (Adyanti, 2018):

$$\int\int K(x_i, x_j) f(x_i) f(x_j) dx_i dx_j \geq 0, \forall f \in L^2$$

Berikut pada Tabel 2.1 merupakan kernel yang digunakan dalam SVM:

Tabel 2.1 Kernel SVM

No.	Fungsi Kernel	Formula	Parameter Optimasi
1	Linear	$K(x_i, x_j) = (x_i, x_j)$	C
2	RBF	$K(x_i, x_j) = \exp(-\gamma \lVert x_i, x_j \rVert^2 + C)$	C dan γ
3	*Sigmoid*	$K(x_i, x_j) = \tanh(\gamma(x_i, x_j) + r)$	C, γ, dan r
4	*Polynomial*	$K(x_i, x_j) = (\gamma(x_i, x_j) + r)^d$	C, γ, r, dan d

(Nanda, et al, 2018)

Dengan x_i, x_j merupakan pasangan dua data dari data latih, dan parameter C, γ, r, dan d adalah konstanta. Parameter kernel dapat dilakukan pengoptimalan, sehingga dapat memebrikan keakuratan terbaik dalam proses klasifikasi. Dengan menggunakan fungsi pemetaan $\phi(x)$, setiap bentuk $x_i.x_j$ akan dihitung

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menngunakan $K(x_i, x_j)$. Stelah x'_i dipetakan ke ruang dimensi lebih tinggi, model masalah diberikan pada persamaan berikut:

$$\text{maks } L'_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x'_i . x'_j)$$

Menjadi

$$\text{maks } L'_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (\phi(x'_i) . \phi(x'_j))$$

Bias yang sebelumnya tidak dipetakan, sehingga $\phi(x_i) = [\phi(x_i)\lambda]$. Oleh karena itu, diperoleh persamaan:

$$\begin{aligned} (\phi(x'_i) . \phi(x'_j)) &= [\phi(x_i)\lambda] \, [\phi(x_i)\lambda]^{transpos} \\ &= (\phi(x_i) . \phi(x_i) + \lambda^2) \\ &= (K(x_i, x_j) + \lambda^2) \end{aligned}$$

Bentuk masalah menjadi

$$\text{maks } L'_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (K(x_i, x_j) + \lambda^2)$$

Dengan syarat $0 \leq \alpha_i \leq C$. Permasalahan di atas juga dapat ditulis ke dalam bentuk:

$$\text{maks } L'_p D^{\alpha} = \sum_{i} \alpha_i - \left(\frac{1}{2} \alpha D \alpha^{transpos} \right)$$

Dengan komponen $D_{i,j} = y_i y_j (K(x_i, x_j) + \lambda^2)$ (Adyanti, 2018)

Terdapat algoritma *sequantial* untuk mencari α. Selanjutnya SVM yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

digunakan disebut SVM *sequantial*. Berikut langkah-langkah dalam algoritma tersebut:

a. Inisialisasi $\alpha_i = 0$

b. Komponen matriks dihitung dengan persamaan $[D]_{ij} = y_i y_j (K(x_i, x_j) + \lambda^2)$

c. Lakukan langkah 1.),2.), dan 3.) di bawah untuk $i = 1, 2, ...N$

   1.) $E_i = \sum_{j=1}^{N} \alpha_i D_{ij}$

   2.) $\delta\alpha_i = min\left\{maks\left[\gamma(11 - E_i), -\alpha_i\right], C - \alpha_i\right\}$

   3.) $\alpha_i = \alpha_i + \delta\alpha_i$

d. Kembali ke langkah c sampai nilai α mencapai konvergen $(|\delta\alpha_i| < \varepsilon)$

Pada algoritma di atas, γ adalah parameter untuk mengontrol kecepatan proses *learning*. Konvergensi dapat didefinisikan dari tingkat perubahan nilai α (Nugroho, Witarto, and Handoko, 2003).

2.8. Pengukuran Kerja Klasifikator

Sistem yang melakukan sebuah klasifikasi diharapkan dapat melakukan klasifikasi semua set data dengan benar. Salah satu cara yang dilakukan untuk mengukur kinerja suatu sistem klasifikasi yaitu dengan menggunakan *Confusion Matrix*. *Confudion matrix* berisi informasi tentang aktual dan prediksi yang dilakukan oleh sistem klasifikasi. Kinerja sistem dievaluasi menggunakan data dalam matriks (Santra and Christy, 2012). Evaluasi ini diperoleh dengan menghitung secara statistik dari nilai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN) yang ilustrasinya dapat dilihat pada Gambar 2.16.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

	Predict : No	**Predict : Yes**
Actual: No	TN	FP
Actual: Yes	FN	TP

Gambar 2.16 ***Confusion Matrix***

(Navin and Pankaja, 2016)

Dari Gambar 2.16 dapat dihasilkan berbagai perhitungan statistik nilai akurasi, spesifisitas, dan sensitivitas. Ketiga hasil tersebut dapat merepresentasikan hasil baik ataupun buruknya sistem klasifikasi yang bekerja. Hasil perhitungan ketiga nilai tersebut diperoleh melalui anggota-anggota dalam *confusion matrix* yang didapatkan dari proses validasi atau uji model yang didapatkan. Berikut merupakan keterangan dari entri *confusion matrix*:

TP (*True Positive*), merupakan suatu data positif yang benar terklasifikasikan menjadi data positif

FP (*False Positive*), merupakan sebuah data negatif yang salah terklasifikasikan menjadi data positif

FN (*False Negative*), merupakan suatu data positif yang salah terklasifikasikan menjadi data negatif

TN (*True Negative*), merupakan sebuah data negatif yang benar terklasifikasikan menjadi data negatif (Santra and Christy, 2012)

1. Akurasi

Akurasi merupakan hasil yang merepresentasikan banyaknya data yang terklasifikasi benar dengan data aktual. Oleh karena itu, semakin besar nilai

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

akurasi, maka semakin banyak data yang terklasifikasi dengan benar (Adyanti, 2018). Persamaan akurasi dapat dilihat pada Persamaan 2.41.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \tag{2.41}$$

2. Spesifisitas

Spesifisitas merupakan hasil yang merepresentasikan banyaknya data yang benar terklasifikasin pada kelas negatif. Oleh karena itu, semakin besar nilai spesifisitas, maka semakin besar sistem klasifikasi dapat mengklasifikasikan kelas negatif dengan baik (Adyanti, 2018). Persamaan spesifisitas dapat dilihat pada Persamaan 2.42.

$$Spesifisitas = \frac{TN}{FP + TN} \tag{2.42}$$

3. Sensitivitas

Sensitivitas merupakan hasil yang merepresentasikan banyaknya data yang benar terklasifikasin pada kelas positif. Oleh karena itu, semakin besar nilai sensitivitas, maka semakin besar sistem klasifikasi dapat mengklasifikasikan kelas positif dengan baik (Adyanti, 2018). Persamaan sensitivitas dapat dilihat pada Persamaan 2.43.

$$Sensitivitas = \frac{TP}{TP + FN} \tag{2.43}$$

2.9. FFA-SVM

Penggabungan metode FFA dengan SVM dimaksudkan untuk meminimalkan kesalahan klasifikasi. SVM merupakan suatu metode yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

memperkenalkan konsep kernel untuk menyelesaikan masalah pemisahan data menjadi dua kelas pada data nonlinear. Proses ini biasanya dibuat dengan memilih tingkat generalisasi yang diwakili oleh parameter-parameter seperti konstanta margin dan kernel. Oleh karena itu, perlu dilakukan pemilihan nilai parameter yang optimal untuk mendapatkan model terbaik dengan tingkat keakuratan yang tinggi (Sharma, et al., 2013).

Proses FFA dilakukan untuk mendapatkan nilai parameter dari proses SVM secara optimal. Proses ini dilakukan sebelum melakukan proses klasifikasi menggunakan SVM. Pada proses SVM dengan fungsi kernel linear dibutuhkan parameter C,fungsi kernel RBF dibutuhkan parameter C dan γ, fungsi kernel *sigmoid* dibutuhkan parameter C, γ, dan r, dan fungsi kernel *polynomial* dibutuhkan parameter C, γ, r, dan d. Parameter C yang menentukan besar penalti akibat dari kesalahan dalam proses klasifikasi. Parameter C yang tepat dapat memaksimalkan *margin* dengan baik dan menghasilkan *hyperplane* yang optimal. Parameter γ, r, dan d merupakan tingkat keragaman pada fungsi basisnya. Apabila parameter tersebut tepat, maka akan mendapatkan suatu fungsi yang akan memetakan data ke ruang dimensi yang lebih tinggi secara tepat.

Satu parameter pada SVM merupakan satu variabel pada FFA. Populasi awal pada FFA dievaluasi menggunakan fungsi objektif untuk mendapatkan nilai *fitness* dari setiap individu dengan persamaan (Suyanto, 2005):

$$f(x) = x_b + (x_a - x_b)g \tag{2.44}$$

dimana x_b dan x_a merupakan batas bawah dan batas atas dari nilai parameter, sedangkan g adalah nilai dari posisi di titik x kunang-kunang. Nilai variabel yang diperoleh akan digunakan dalam proses pengkalsifikasian pada metode SVM.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB III

METODE PENELITIAN

3.1. Jenis Penelitian

Penelitian mengenai klasifikasi tumor otak menggunakan metode FFA *optimization* dan SVM *classifier* merupakan penelitian deskriptif kuantitatif karena mengandung unsur perhitungan dan analisis dari hasil yang didapatkan. Selain itu, jika ditinjau dari aspek fungsinya, penelitian ini termasuk dalam kategori penelitian terapan. Hasil penelitian ini bertujuan untuk mempercepat proses dan membantu pihak medis dalam mendeteksi penyakit tumor otak.

3.2. Pengumpulan Data

Data yang digunakan dalam penelitian ini adalah data MRI otak dan untuk mendapatkan karakteristiknya digunakan analisis tekstur GLCM. Analisis GLCM digunakan untuk mendapatkan fitur tekstur sebagai inputan bagi SVM *classifier* untuk klasifikasi tumor otak atau kondisi normal. Namun, sebelum dilakukan klasifikasi, parameter SVM dioptimalkan dengan menggunakan FFA *optimization* agar hasil kerja SVM lebih optimal. Jumlah data citra otak dalam penelitian ini sebanyak 120 data (60 data normal dan 60 data tumor) yang diperoleh dari *kaggle dataset*.

3.3. Analisis Data

Data MRI otak yang sudah diklasifikasi oleh ahli terbagi ke dalam dua klasifikasi yaitu normal dan tumor. Data dibagi menjadi dua bagian. Bagian

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

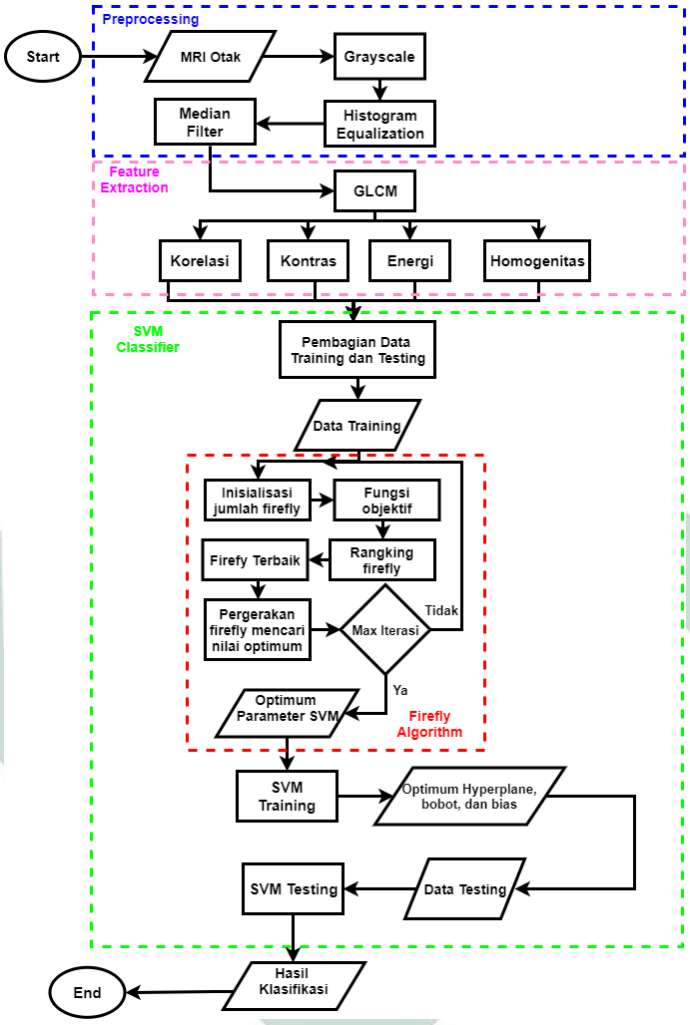
pertama sebagai data *training* dan bagian kedua sebagai data *testing* dengan menggunakan *k-fold*. Untuk mencapai tujuan dalam penelitian ini, perlu dilakukan langkah-langkah analisis data dengan menggunakan tiga tahapan, yaitu *pre-processing*, ekstraksi fitur menggunakan GLCM dan klasifikasi menggunakan FFA *optimization* dan SVM *classifier*. Langkah awal yang dilakukan adalah perbaikan kualitas citra atau biasa disebut sebagai *pre-processing*. Proses ini dilakukan karena hasil citra otak memiliki intensitas yang berbeda. Proses ini dilakukan dengan beberapa tahapan. Tahap pertama adalah merubah citra menjadi citra *grayscale* untuk mendapatkan ekstrasi fitur statistik dengan metode GLCM. Setelah diubah ke *grayscale*, maka dilakukan perbaikan citra dengan filtering untuk menghilangkan noise pada citra. Setelah itu, dilakukan perataan histogram menggunakan *histogram equalization* untuk peningkatan kontras, sehingga akan mudah diklasifikasi. Selanjutnya dilakukan analisis tekstur menggunakan GLCM untuk mendapatkan fitur tekstur yang akan digunakan sebagai input bagi SVM untuk klasifikasi tumor otak.

3.4. Pengujian dan Evaluasi

Pengujian pertama dilakukan dengan proses *image pre-processing* dengan mengambil citra otak yang selanjutnya dilakukan ekstraksi fitur menggunakan GLCM untuk mendapat analisis tekstur yang akan digunakan sebagai input bagi SVM untuk klasifikasi tumor otak atau kondisi normal. Setelah dilakukan klasifikasi menggunakan FFA dan SVM, digunakan pengukur kerja klasifikator yang meliputi akurasi, sensitifitas, dan sepsitifitas. Dalam mencapai tujuan penelitian ini, perlu langkah-langkah secara urut dan sistematis, adapun Gambar 4.10 merupakan representasi dari diagram alir *flowchart* untuk klasifikasi tumor otak menggunakan metode FFA-SVM. Gambar 4.10 adalah diagram alir *flowchart*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

untuk klasifikasi tumor otak mengguanakan FFA *Optimization* dan SVM *Classifier*.



Gambar 3.1 Diagram Alir untuk Klasifikasi Tumor Otak

Diagram alir pada Gambar 4.10 memiliki tujuan yang akan dicapai yaitu untuk mendapatkan hasil klasifikasi yang baik. Oleh karena itu perlu dilakukan langkah-langkah pengerjaan yang terstruktur. Berdasarkan diagram alir tersebut, dapat disimpulkan langkah-langkah penelitian adalah :

1. *Prerocessing*

Pada proses ini dilakukan perbaikan citra MRI otak dengan mengubah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

citra MRI dari citra RGB menjadi citra *grayscale* menggunakan Persamaan 2.1. Setelah didapatkan citra *grayscale*, dilakuakn proses filtering dengan menggunakan metode *median filtering* menggunakan Persamaan 2.6. Hasil dari proses filtering digunakan untuk proses selanjutnya, yaitu *histogram equalization* untuk perataan histogram menggunakan Persamaan 2.5. Kemudianj proses selanjutnya adalah ekstraksi fitur (*feature extraction*) menggunakan GLCM.

2. Ekstraksi Fitur

   Pada proses ini, citra akhir dari proses *histogram equalization* diproses untuk didapatkan data numerik atau statistiknya berdasarkan tekstur citra. Proses ini menggunakan metode GLCM. Fitur yang diambil sebagai data klasifikasi yaitu fitur korelasi, kontras, energi, dan homogenitas. Tahap pertama yang dilakukan adalah menentukan hubungan antar piksel referensi dan tetangga dengan sudut θ dan jarak $d_p = 1$. Selanjutnya dilakukan perhitungan jumlah *co-occurrence* dalam sebuah matriks. Setelah didaptkan matriks *co-occurrence*, dijumlahkan dengan transposenya dan dilakukan normalisasi menggunakan Persamaan 2.7. Kemudian dihitung nilai korelasi dengan Persamaan 2.11, kontras dengan Persamaan 2.9, energi dengan Persamaan 2.8, dan homogenitas dengan Persamaan 2.10. Nilai-nilai dari fitur GLCM ini akan digunakan sebagai inputan data pada proses klasifikasi.

3. Klasifikasi SVM

   Tahap ini merupakan tahap proses pengklasifikasian citra MRI otak. Pada tahap ini dilakukan proses *learning* menggunakan *k-fold* dengan $k = 5$. Selanjutnya dengan data *training* akan dilakukan proses penemuan model svm untuk proses pada *svm testing*. Proses *learning* pertama dilakukan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dengan menggunakan SVM sederhana tanpa menggunakan pengoptimasian parameter. Nilai akurasi dari proses *learning* pertama akan dilakukan pengoptimalan pada parameter SVM menggunakan FFA dalam proses *learning* kedua. Pengoptimalan parameter svm yang terdiri dari parameter konstanta *margin* dan kernel dengan metode FFA (*firefly algorithm*).

Tahap pertama yang dilakuakn adalah menginisialisasi jumlah *firefly*, maksimal iterasi pada proses FFA, dan parameter dari proses FFA, yang termasuk inisialisasi variabel untuk parameter svm yang akan dioptimalkan. Setelah dilakukan inisialisasi, proses selanjutnya adalah perhitungan nilai fungsi objektif dengan menggunakan Persamaan 2.44. Setelah itu, dilakuakn perangkingan untuk mendapatkan nilai *fitness* pertama yang merupakan *firefly* terbaik. Kemudian dilakukan pergerakan *firefly* menuju *firefly* dengan cahaya paling cerah menggunakan Persamaan 2.29. Selanjutnya dilakukan perhitungan kembali fungsi objektif sampai mendapatkan *firefly* dengan nilai *fitness* terbaik atau sampai iterasi berhenti.

Setelah mendapatkan parameter optimal yang merupakan nilai *fitness* terbaik dari proses FFA, dilakukan proses *training* untuk mendapatkan *hyperplane* optimal, bobot, dan bias pada model SVM. Proses terakhir yaitu menggunakan model svm yang telah didapatkan untuk proses *testing*. Proses *testing* akan menghasilkan hasil dari klasifikasi yang dapat digunakan sebagai perhitungan keakuratan model.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB IV

HASIL DAN PEMBAHASAN

Penelitian ini bertujuan untuk mengetahui hasil kinerja dari FFA (*Firefly Algorithm*) untuk mengoptimasi parameter SVM (*Support Vector Machine*) Classifier dalam klasifikasi penyakit tumor otak. Selain itu, penelitian ini juga akan membandingankan hasil sistem dan waktu komputasi yang diperlukan dalam proses klasifikasi menggunakan SVM-FFA dengan SVM sederhana. Simulasi pada penelitian ini menggunakan perangkat lunak dengan pemrogrman Python. Pembahasan dalam penelitian ini meliputi perbaikan citra, pemilihan model, *learning*, evaluasi dan validasi model.

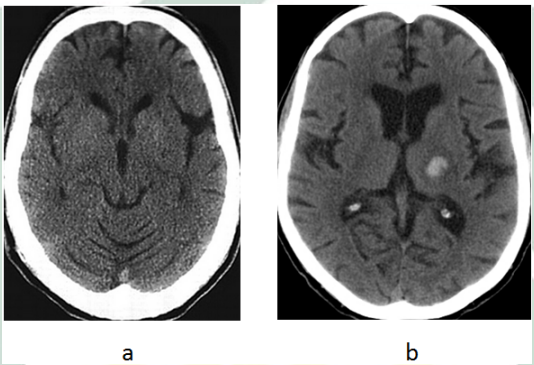
4.1. Perbaikan Citra

Tahap ini merupakan tahap pengolahan citra untuk mendapatkan hasil yang akurat saat dilakukan proses klasifikasi. Proses ini dilakukan karena citra yang didapat biasanya belum memenuhi standar pengolahan. Salah satu permasalahan pada citra MRI otak yaitu intensitas cahaya yang kurang merata serta adanya *noise* pada citra. Adanya permasalahan tersebut, maka perlu dilakukan proses konversi ke citra *grayscale* untuk menghilangkan *noise* pada citra RGB. Selanjutnya dilakukan proses *histogram equalization* untuk meratakan intensitas piksel citra. Proses terakhir yang dilakukan adalah *median filter* untuk menghilangkan *noise* jenis *salt and pepper*.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4.1.1. Data

Pada penelitian ini digunakan data MRI citra otak yang diperoleh dari *kaggle dataset* dengan total data sebanyak 120 data dalam bentuk citra RGB yang terdiri dari 60 data normal dan 60 data tumor. Citra tumor otak ditandai dengan adanya warna yang mencolok atau intensitas piksel yang lebih terang pada bagian supratentorial. Gambar 4.1 adalah sampel data citra MRI otak yang digunakan.



Gambar 4.1 a) Sampel Data Normal b) Sampel Data Tumor

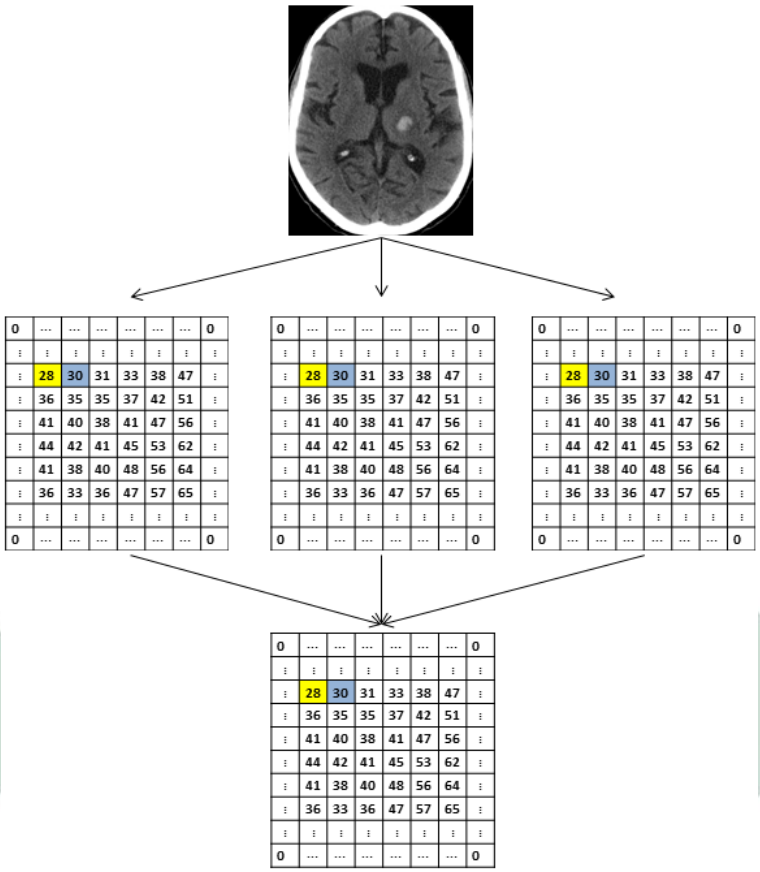
4.1.2. Konversi *Grayscale*

Proses *grayscale* dilakukan untuk menyederhanakan piksel pada citra, sehingga lebih mudah diproses pada tahap selanjutnya. Selain itu, pada citra RGB dengan piksel merah (*red*), hijau (*green*), dan biru (*blue*) memeliki *noise* yang cukup tinggi, sehingga pengkonversian ke *grayscale* diterapkan untuk mengurangi piksel *noise* pada citra. Hasil dari proses *grayscale* diuraikan pada Gambar 4.2.

Gambar 4.2 menunjukkan piksel berukuran 593×479 dari salah satu data citra otak yang terdiri dari komponen piksel *Red*, *Green*, dan *Blue*. Piksel tersebut dikonversi menjadi piksel citra *grayscale* dengan menggunakan Persamaan 2.1. Misalkan matriks piksel pada citra RGB adalah matriks A dan matriks piksel (*grayscale*) adalah matriks B. Pada contoh perhitungan, nilai perhitungan piksel

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$B(281, 238)$ ditunjukkan dengan warna kuning dan $B(281, 239)$ warna biru.



Gambar 4.2 Proses *Grayscale*

Nilai piksel pada posisi $B(i, j)$ dengan i adalah baris dan j adalah kolom, dapat diperoleh dengan:

$$B(281, 238) = (0.299(r)) + (0.587(g)) + (0.114(r))$$

$$B(281, 238) = (0.299(28)) + (0.587(28)) + (0.114(28)) = 28$$

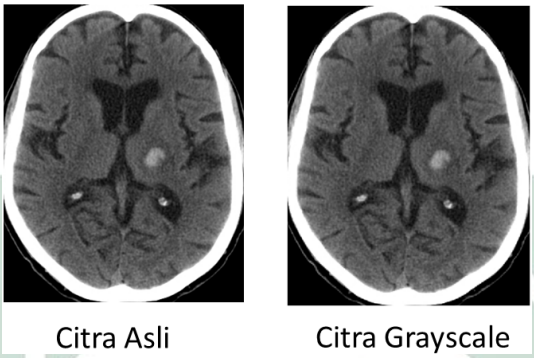
$$B(281, 239) = (0.299(r)) + (0.587(g)) + (0.114(r))$$

$$B(281, 239) = (0.299(30)) + (0.587(30)) + (0.114(30)) = 30$$

Pada citra *grayscale* warna gelap (mendekati hitam) memiliki nilai piksel yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

mendekati 0, dan warna terang (mendekati putih) memiliki nilai piksel yang mendekati 255. Karena data citra otak yang digunakan banyak, maka digunakan *syntax* atau *function rgb2gray* pada *software* Python. Hasil dari proses konversi citra RGB ke *grayscale* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Hasil Citra Proses Konversi ke *Grayscale* dengan ukuran 593×479

4.1.3. *Histogram Equalization*

Proses *histogram equalization* dilakukan untuk meratakan dan meningkatkan kontras suatu citra agar piksel keabuan pada citra MRI otak relatif sama. Berikut adalah proses *histogram equalization* dengan piksel hasil pengkonversian ke citra *graysccale* dengan ukuran 593×479:

1. Menghitung kemunculan derajat keabuan pada citra dari rentang $[0, 225]$.

0	⋯	⋯	⋯	⋯	⋯	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	28	30	31	33	38	47	⋮
⋮	36	35	35	37	42	51	⋮
⋮	41	40	38	41	47	56	⋮
⋮	44	42	41	45	53	62	⋮
⋮	41	38	40	48	56	64	⋮
⋮	36	33	36	47	57	65	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	⋯	⋯	⋯	⋯	⋯	⋯	0

⬇

DG	0	...	30	31	32	33	34	35	36	37	38	...	255
K	41812	...	1059	1009	986	938	887	924	900	933	922	...	0

Gambar 4.4 Perhitungan Jumlah Kemunculan Nilai Piksel pada Citra *Grayscale*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tahap ini merupakan tahap perhitungan jumlah kemunculan dari nilai piksel pada citra *grayscale* menggunakan Persamaan 2.11 yang ditunjukkan pada Gambar 4.4. Pada Gambar 4.4, DG adalah nilai tingkat keabuan atau piksel citra *grayscale* dan K adalah banyak kemunculan dari piksel keabuan yang sesuai dengan matriks piksel pada citra awal.

2. Menghitung probabilitas kemunculan derajat keabuan pada citra dari rentang $[0, 225]$.

Pada tahap ini dilakukan perhitungan probabilitas dari kemunculan piksel keabuan menggunakan Persamaan 2.3 yang ditunjukkan pada Gambar 4.5.

DG	0	...	30	31	32	33	34	35	36	37	38	...	255
K	41812	...	1059	1009	986	938	887	924	900	933	922	...	0
P	0.147	...	0.004	0.004	0.003	0.003	0.003	0.003	0.003	0.003	0.003	...	0

Gambar 4.5 Perhitungan Probabilitas dari Kemunculan Piksel pada Citra *Grayscale*

Pada Gambar 4.5, P adalah probabilitas dari kemunculan piksel. Berikut adalah contoh perhitungan probabilitas pada DG 18,20, dan 21.

$$
\begin{aligned}
P(r_k) &= \frac{n_k}{n} & P(r_k) &= \frac{n_k}{n} \\
p(0) &= \frac{41812}{284047} & p(30) &= \frac{1059}{284047} \\
p(0) &= 0.147201 & p(30) &= 0.003728
\end{aligned}
$$

Proses probabilitas pada tahap ini juga disebut sebagai proses normalisasi histogram dari kemunculan piksel. Oleh karena itu, ketika dijumlahkan keseluruhan, akan mengasilkan nilai piksel 1, sesuai dengan kisaran normal tingkat dominasi $[0, 1]$. Nilai n bernilai 284047 dari total seluruh nilai K (41812 + ... + 1059 + 1009 + 986 + 938 + 887 + ... + 0 = 284047)

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

3. Menghitung transformasi dari hasil probabilitas kemunculan derajat keabuan pada citra dari rentang $[0, 225]$.

DG	0	...	30	31	32	33	34	35	36	37	38	...	255
K	41812	...	1059	1009	986	938	887	924	900	933	922	...	0
P	0.147	...	0.004	0.004	0.003	0.003	0.003	0.003	0.003	0.003	0.003	...	0
T	0.147		0.189	0.192	0.195	0.199	0.202	0.205	0.208	0.212	0.215	...	1.000

Gambar 4.6 Perhitungan Transformasi Probabilitas dari Kemunculan Piksel pada Citra *Grayscale*

Pada tahap ini dilakukan transformasi dari hasil probabilitas dengan menggunakan Persamaan 2.5. Tahap ini menghasilkan rentang progresif dan kontras yang lebih baik. Gambar 4.6 adalah proses dari transformasi hasil probabilitas. Pada Gambar 4.6 hasil dari nilai T adalah penjumlahan kumulatif dari setiap piksel probabilitas. Berikut adalah contoh perhitungan nilai T pada DG 33 dan 38.

$$
\begin{aligned}
T(r_k) &= \sum_{j=0}^{k} p_r(r_j) \\
T(33) &= 0.147 + ... + 0.004 + 0.004 + 0.003 + 0.003 \\
T(33) &= 0.199
\end{aligned}
$$

4. Mengalikan hasil transformasi dengan nilai rentang maksimal, yaitu 255.

DG	0	...	30	31	32	33	34	35	36	37	38	...	255
K	41812	...	1059	1009	986	938	887	924	900	933	922	...	0
P	0.147	...	0.004	0.004	0.003	0.003	0.003	0.003	0.003	0.003	0.003	...	0
T	0.147		0.189	0.192	0.195	0.199	0.202	0.205	0.208	0.212	0.215	...	1.000
T*255	38		48	49	50	51	51	52	53	54	55		255

Gambar 4.7 Perhitungan Perkalian Hasil Transformasi dengan Nilai Rentang Maksimal pada Citra *Grayscale*

Tahap ini merupakan proses terakhir perhitungan histogram, yaitu dengan mengalikan hasil piksel transformasi dengan nilai rentang maksimal, yaitu 255.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Gambar 4.7 adalah hasil dari proses pengalian piksel transformasi dengan nilai rentang maksimal citra *grayscale*. Pada perhitungan ini, dilakukan pembulatan pada hasil perkalian. Sebagai contoh perhitungan perkalian pada DG(35), yaitu $0.205 \times 255 = 52$. Nilai 25 merupakan nilai hasil dari pembulatan terdekat.

5. Mengganti piksel dengan hasil dari proses *histogram equalization*.

   Tahap terakhir pada proses *histogram equalization* yaitu mengganti nilai piksel pada citra awal dengan nilai $T * 255$ yang telah diperoleh pada tahap sebelumnya. Hasil dari proses pergantian piksel ini dapat dilihat pada Gambar 4.8.

0	⋯	⋯	⋯	⋯	⋯	⋯	0
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	28	30	31	33	38	47	⋮
⋮	36	35	35	37	42	51	⋮
⋮	41	40	38	41	47	56	⋮
⋮	44	42	41	45	53	62	⋮
⋮	41	38	40	48	56	64	⋮
⋮	36	33	36	47	57	65	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
0	⋯	⋯	⋯	⋯	⋯	⋯	0

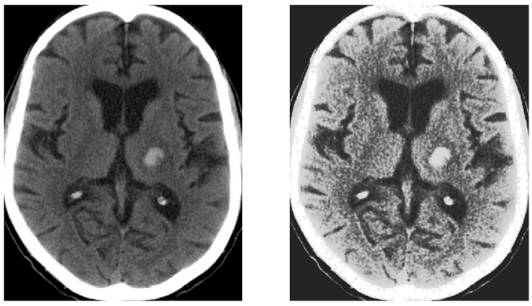
38	⋯	⋯	⋯	⋯	⋯	⋯	38
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	46	48	49	51	55	62	⋮
⋮	53	52	52	54	58	66	⋮
⋮	57	57	55	57	62	72	⋮
⋮	60	58	57	61	68	79	⋮
⋮	57	55	57	63	72	82	⋮
⋮	53	51	53	62	73	83	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	⋯	⋯	⋯	⋯	⋯	⋯	38

Gambar 4.8 Hasil Piksel dari Proses *Histogram Equalization*

Gambar 4.8 menunjukkan perubahan nilai piksel pada citra awal. Misalkan piksel citra awal dengan GD(38) memiliki nilai $T * 255$ sebesar 55, maka nilai piksel 38 diganti dengan 55 sebagai hasil dari proses *histogram equalization*. Contoh ini ditandai dengan warna kuning pada Gambar 4.8.

Proses ini juga dapat dipermudah dengan bantuan *function* pada Python menggunakan *syntax equalize_ hist*. Berikut adalah hasil akhir citra setelah dilakukan proses *histogram equalization* yang disajikan pada Gambar 4.9. Hasil citra merepresentasikan kontras atau intensitas cahaya yang lebih merata dari sebelumnya, sehingga citra terlihat lebih jelas.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id



Citra *Grayscale* Citra Hasil HE

Gambar 4.9 Hasil Citra Proses *Histogram Equalization* dengan ukuran 593×479

4.1.4. *Median Filter*

Proses *median filter* merupakan proses *filtering* yang digunakan untuk membantu mengurangi *noise* jenis *salt and pepper* pada citra. Sebelum melakukan proses *median filter*, dilakukan penentuan ukuran *mask* yang akan digunakan. Pada penelitian ini, digunakan *mask* dengan ukuran 3×3. Hasil dari proses *median filter* dapat memberikan ukuran citra yang berbeda dari citra awal. Namun, masalah ini dapat diatasi dengan menambahkan *pading* pada piksel citra agar hasil yang diperoleh tetap berukuran sama dengan citra awal dengan ukuran 593×479 seperti pada Gambar 4.10.

38	⋯	⋯	⋯	⋯	⋯	⋯	38
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	46	48	49	51	55	62	⋮
⋮	53	52	52	54	58	66	⋮
⋮	57	57	55	57	62	72	⋮
⋮	60	58	57	61	68	79	⋮
⋮	57	55	57	63	72	82	⋮
⋮	53	51	53	62	73	83	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	⋯	⋯	⋯	⋯	⋯	⋯	38

➡

38	38	⋯	⋯	⋯	⋯	⋯	⋯	38	38
38	38	⋯	⋯	⋯	⋯	⋯	⋯	38	38
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	⋮	46	48	49	51	55	62	⋮	⋮
⋮	⋮	53	52	52	54	58	66	⋮	⋮
⋮	⋮	57	57	55	57	62	72	⋮	⋮
⋮	⋮	60	58	57	61	68	79	⋮	⋮
⋮	⋮	57	55	57	63	72	82	⋮	⋮
⋮	⋮	53	51	53	62	73	83	⋮	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	38	⋯	⋯	⋯	⋯	⋯	⋯	38	38
38	38	⋯	⋯	⋯	⋯	⋯	⋯	38	38

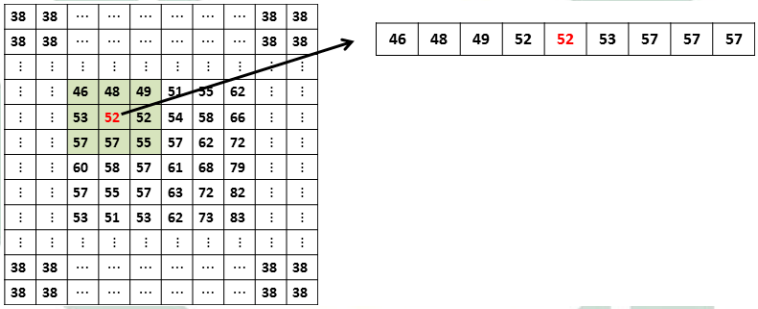
Gambar 4.10 Penambahan Pading untuk Proses *Median Filter*

Penambahan *pading* pada piksel citra dapat dilakukan dengan dua cara, penambahan piksel zeros atau penambahan berdasarkan ketetanggaan. Dalam

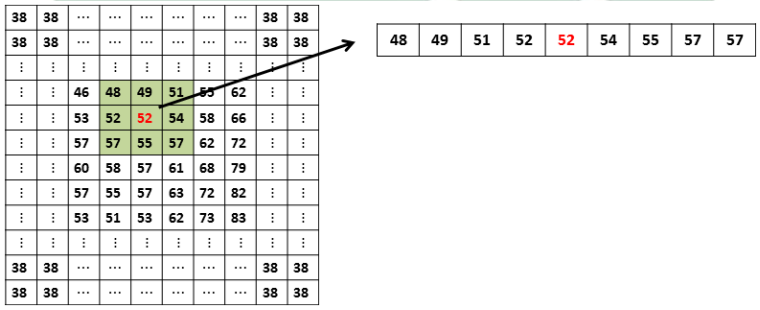
digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

penelitian ini, cara yang dipilih adalah penambahan *padding* berdasarkan ketetanggaan. Warna biru pada Gambar 4.10 merupakan *padding* yang ditambahkan. Sedangkan warna abu-abu merupakan bagian yang akan digunakan untuk contoh perhitungan proses *median filter*. Selanjutnya proses *median filter* dapat dilakukan dengan menggunakan *mask* yang sudah ditentukan sebelumnya.

Proses *median filter* dapat dihitung menggunakan Persamaan 2.6. Nilai piksel yang telah diterapkan *masking* (*sliding window*) diurutkan dari nilai terkecil ke nilai terbesar, kemudian nilai dari mediannya digunakan untuk menggantikan nilai pusat dari *sliding window*.



Gambar 4.11 Proses *Median Filter* dengan Mask ke-1



Gambar 4.12 Proses *Median Filter* dengan Mask ke-2

Pada Gambar 4.11 dan Gambar 4.12 menunjukkan proses *median filter* pada citra ukuran 593×479 dengan *mask* ke-1 dan ke-2. *Sliding window* pada gambar tersebut diberikan dengan warna hijau, merupakan piksel citra dengan *mask* yang telah ditentukan ukurannya. Pada *sliding window* terdapat piksel berwarna merah

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

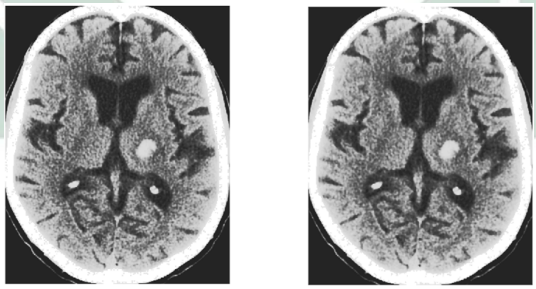
yang merupakan pusat dari *sliding window*. Pusat inilah yang nilainya akan diganti dengan nilai *median* yang didapatkan. Oleh karena itu, pada *mask* pertama, nilai pusat piksel 52 diganti dengan nilai yang tetap, yaitu 52. Karena setelah ditemukan mediannya, nilainya sama seperti nilai piksel pusat *mask* dan pada *mask* kedua, nilai pusat piksel 52 diganti dengan nilai yang juga tetap, yaitu 52 sebagai nilai mediannya. Proses ini berulang hingga pada pusat *sliding window* terakhir. Hasil akhir piksel pada proses *median filter* disajikan dalam Gambar 4.13

38	⋯	⋯	⋯	⋯	⋯	⋯	38
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	46	48	49	51	55	62	⋮
⋮	53	52	52	54	58	66	⋮
⋮	57	57	55	57	62	72	⋮
⋮	60	58	57	61	68	79	⋮
⋮	57	55	57	63	72	82	⋮
⋮	53	51	53	62	73	83	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	⋯	⋯	⋯	⋯	⋯	⋯	38

38	⋯	⋯	⋯	⋯	⋯	⋯	38
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
⋮	46	49	50	53	58	65	⋮
⋮	53	52	52	55	58	66	⋮
⋮	57	56	56	57	62	71	⋮
⋮	58	57	57	60	68	78	⋮
⋮	58	56	57	62	71	81	⋮
⋮	55	53	55	62	71	81	⋮
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
38	⋯	⋯	⋯	⋯	⋯	⋯	38

Gambar 4.13 Hasil Piksel dari Proses *Median Filter*

Berikut adalah hasil citra dari proses *median filter* pada Gambar 4.14.



Citra HE Citra Hasil *Median Filter*

Gambar 4.14 Hasil Citra dari Proses *Median Filter* dengan ukuran 593×479

Proses *median filter* membutuhkan waktu yang relatif lama karena adanya perulangan masking. Oleh sebab itu, dapat digunakan *function* pad Python untuk proses *median filter* menggunakan *syntax median* untuk proses filter dan *square* untuk bentuk dan ukuran maskingnya. Pada Gambar 4.14 citra hasil *median filter* terlihat lebih halus yang merepresentasikan bahwa noise pada citra berkurang dari

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sebelumnya.

4.1.5. Ekstraksi Fitur

Ekstraksi fitur dilakukan untuk mendapatkan informasi statistik dari citra MRI otak. Data statistik yang didapatkan dari citra dapat mempermudah dan mempercepat proses klasifikasi. Dalam sebuah citra, terdapat fitur yang dapat merepresentasikan karakteristik pada citra. Pada penelitian ini dilakukan analisis tekstur pada citra MRI otak menggunakan metode statistik orde dua, yaitu GLCM. Informasi tekstur yang didapatkan dapat membedakan dua wilayah yang berdekatan atau wilayah yang tumpang tindih. Tekstur dari hasil GLCM ini digunakan untuk membedakan bagian supratentorial dengan massa pada bagian tersebut, yang kemudian digunakan untuk deteksi otak normal dan tumor.

GLCM merupakan metode ekstraksi fitur orde dua yang menggunakan matriks kookurensi sebagai representasi dari hubungan antara kelompok dua piksel dalam citra asli. Hubungan ini didasarkan pada berbagai arah orientasi dan jarak spasial. Pada penelitian ini, digunakan jarak $d = 1$ piksel dengan sudut orientasi yang meliputi 0°, 45°, 90°, dan 135°. Tahapan untuk mendapatkan matriks kookurensi dapat dilhat pada Bab II subsubbab Ekstraksi Fitur Citra Tumor Otak. Berikut adalah proses mendapatkan fitur dengan metode GLCM:

1. Membuat area kerja dari matriks citra.

$$A = \begin{pmatrix} 38 & 38 & \cdots & 38 \\ 38 & 38 & \cdots & 38 \\ \vdots & \vdots & \ddots & \vdots \\ 38 & 38 & \cdots & 38 \end{pmatrix}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Misalkan A adalah citra hasil proses *median filter*. Matriks citra ini yang digunakan dalam contoh perhitungan ekstraksi fitur. Sebelum membuat area kerja matriks, dilakukan proses normalisasi ke dalam 8 piksel dengan menggunakan Persamaan 4.1:

$$A_{i,j} = \frac{(A_{i,j} - Min)(NMax - Nmin)}{Max - Min} + NMin \tag{4.1}$$

Dimana $A_{i,j}$ merupakan piksel keabuan citra pada baris ke-i dan kolom ke-j, Min dan Max merupakan nilai piksel terkecil dan terbesar dari piksel citra, $NMin$ dan $NMax$ merupakan nilai terkecil dan terbesar dari rentang piksel baru. Berikut adalah hasil dari normalisasi piksel citra.

$$A = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix}$$

Matriks hasil normalisasi tersebut yang akan digunakan dalam proses GLCM. Selanjutnya, dilakukan pembuatan area kerja untuk proses mendapatkan matriks *co-occurence*. Area kerja yang dibuat berukuran sesuai dengan level dari tingkat keabuan citra. Dalam penelitian ini, level keabuan yang digunakan adalah 8. Hasil area kerja untuk citra MRI otak ditunjukkan pada Tabel 4.1 dengan contoh perhitungan dari proses normalisasi pada $A_{1,1}$:

$$\begin{aligned} A_{i,j} &= \frac{(A_{i,j} - Min)(NMax - Nmin)}{Max - Min} + NMin \\ A_{1,1} &= \frac{(38 - 38)(8 - 1)}{255 - 38} + 1 \\ A_{1,1} &= 1 \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.1 Area Kerja Matriks dengan ukuran matriks 8×8

t \ r	1	2	3	4	5	6	7	8
1	(1,1)	(1,2)	(1,3)	(1,4)	(1,5)	(1,6)	(1,7)	(1,8)
2	(2,1)	(2,2)	(2,3)	(2,4)	(2,5)	(2,6)	(2,7)	(2,8)
3	(3,1)	(3,2)	(3,3)	(3,4)	(3,5)	(3,6)	(3,7)	(3,8)
4	(4,1)	(4,2)	(4,3)	(4,4)	(4,5)	(4,6)	(4,7)	(4,8)
5	(5,1)	(5,2)	(5,3)	(5,4)	(5,5)	(5,6)	(5,7)	(5,8)
6	(6,1)	(6,2)	(6,3)	(6,4)	(6,5)	(6,6)	(6,7)	(6,8)
7	(7,1)	(7,2)	(7,3)	(7,4)	(7,5)	(7,6)	(7,7)	(7,8)
8	(8,1)	(8,2)	(8,3)	(8,4)	(8,5)	(8,6)	(8,7)	(8,8)

2. Menentukan hubungan spasial antar piksel.

Tabel 4.2 Nilai Hubungan Spasial Antar Piksel

t \ r	1	2	3	4	5	6	7	8
1	69267	2340	132	87	36	52	32	0
2	2336	21546	3688	213	30	78	80	0
3	129	3683	19485	5361	250	32	66	0
4	68	239	5358	22137	5544	240	78	0
5	44	28	261	5559	19861	4156	66	0
6	65	54	32	246	4177	23697	1817	0
7	44	80	55	68	82	1832	24287	1104
8	0	0	0	0	0	0	1178	32074

Hubungan spasial antar piksel ditentukan dengan menentukan ukuran jarak piksel d_p dan sudut orientasi θ. Table 4.2 merupakan hasil hubungan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

spasial antar piksel referensi (r) dan tetangga (t). Pada contoh perhitungan ini digunakan $d_p = 1$ dan $\theta = 0^o$. Nilai $d_p = 1$ berarti nilai tersebut dihitung dengan melihat piksel yang berjarak 1 dari piksel referensi dengan arah sebesar 0° yang mengarah ke sebelah kanan.

3. Menghitung nilai probabilitas

Tabel 4.3 Nilai Probabilitas Matriks *Co-Occurence*

t \ r	1	2	3	4	5	6	7	8
1	0.2444	0.0083	0.0005	0.0003	0.0001	0.0002	0.0001	0.0000
2	0.0082	0.0760	0.0130	0.0008	0.0001	0.0003	0.0003	0.0000
3	0.0005	0.0130	0.0687	0.0189	0.0009	0.0001	0.0002	0.0000
4	0.0002	0.0008	0.0189	0.0781	0.0196	0.0008	0.0003	0.0000
5	0.0002	0.0001	0.0009	0.0196	0.0701	0.0147	0.0002	0.0000
6	0.0002	0.0002	0.0001	0.0009	0.0147	0.0836	0.0064	0.0000
7	0.0002	0.0003	0.0002	0.0002	0.0003	0.0065	0.0857	0.0039
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0042	0.1132

Tahap selanjutnya yaitu menghitung probabilitas pada matriks *co-occurence* yang telah didapatkan dengan menggunakan Persamaan 2.7. Hasil probabilitas matriks *co-occurence* pada citra MRI otak secara lengkap dapat dilihat pada Tabel 4.3. Berikut adalah contoh proses perhitungan probabilitas pada matriks *co-occurence*:

$$P(i,j) = \frac{f(i,j)}{\sum_i \sum_j f(i,j)}$$

$$P(4,3) = \frac{5358}{283454} = 0.0189$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

4. Menghitung Ekstraksi Fitur

Nilai probabilitas pada matriks *co-occurence* digunakan untuk menghitung ekstraksi fitur pada citra MRI otak. Perhitungan ekstraksi fitur dilakukan untuk mendapatkan nilai tekstur secara statistik pada citra. Proses ekstraksi fitur ini juga dapat memepermudah proses klasifikasi, karena informasi citra yang digunakan diubah dalam bentuk statistik. Pada penelitian ini, ekstraksi fitur yang digunakan adalah kontras, korelasi, energi, dan homogenitas. Ekstraksi fitur kontras dapat dihitung menggunakan Persamaan 2.9, korelasi menggunakan Persamaan 2.11, energi menggunakan Persamaan 2.8, dan homogenitas menggunakan Persamaan 2.10.

a. Kontras

$$\begin{aligned}
\text{Kontras} &= \sum_{i=1}^{N}\sum_{j=1}^{N}(i-j)^2 P(i,j) \\
&= (|1-1|^2 \times 0.2444) + (|1-2|^2 \times 0.0083) + ... \\
&\quad +(|4-3|^2 \times 0.0189) + (|4-4|^2 \times 0.0781) + ... \\
&\quad +(|8-7|^2 \times 0.0042) + (|8-8|^2 \times 0.1132) \\
&= 0.2623
\end{aligned}$$

b. Korelasi

Sebelum menghitung nilai korelasi pada citra MRI otak, perlu dilakukan perhitungan nilai *mean* baris, *mean* kolom, standar deviasi baris, dan standar deviasi kolom. Hasil perhitungan μ_i, μ_j, σ_i, dan σ_j dapat dilihat pada Tabel 4.4 hingga Tabel 4.9.

Nilai *mean* baris (μ_i) dihitung dengan contoh perhitungan seperti berikut:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.4 Nilai *Mean* Baris

Baris ke-i	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	0.254	0.197	0.307	0.475	0.529	0.637	0.680	0.939
***Mean* Baris**	**4.018**							

Contoh perhitungan pada μ_4

$$\begin{aligned}
\mu_i &= \sum_{i=1}^{N}\sum_{j=1}^{N} i(P(i,j)) \\
\mu_4 &= 4(0.0002 + 0.0008 + 0.0189 + 0.0781 + 0.0196 + 0.0008 \\
&\quad +0.0003 + 0.0000) \\
\mu_4 &= 0.475
\end{aligned}$$

Nilai *mean* kolom (μ_j) dihitung dengan contoh perhitungan seperti berikut:

Tabel 4.5 Nilai *Mean* Kolom

Kolom ke-j	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**
	0.254	0.197	0.307	0.475	0.529	0.637	0.682	0.936
***Mean* Kolom**	**4.017**							

Contoh perhitungan pada μ_4

$$\begin{aligned}
\mu_j &= \sum_{i=1}^{N}\sum_{j=1}^{N} j(P(i,j)) \\
\mu_4 &= 4(0.0003 + 0.0008 + 0.0189 + 0.0781 + 0.0196 + 0.0009 \\
&\quad +0.0002 + 0.0000) \\
\mu_4 &= 0.475
\end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Nilai standar deviasi baris (σ_i) dihitung dengan memisalkan $1,1$ sebagai representasi nilai dari baris ke satu pada kolom ke satu.

Tabel 4.6 Nilai Standar Deviasi Baris

Baris ke-i	$\mathbf{P(i,j)(i-\mu_i)^2}$
$1,1$	2.2254
$1,2$	0.0752
$\vdots$	$\vdots$
$7,2$	0.0025
$7,3$	0.0017
$\vdots$	$\vdots$
$8,8$	1.7944
Standar Deviasi Baris	**2.4624**

Berikut adalah contoh perhitungan $P(i,j)(i-\mu_i)^2$:

$$\begin{aligned} i,j &= P(i,j)(i-\mu_i)^2 \\ 7,2 &= 0.0003(7-4.018)^2 \\ 7,2 &= 0.0025 \end{aligned} \qquad \begin{aligned} i,j &= P(i,j)(i-\mu_i)^2 \\ 8,8 &= 00.1132(8-4.018)^2 \\ 8,8 &= 1.7944 \end{aligned}$$

Sehingga didapatkan nilai standar deviasi σ_i:

$$\begin{aligned} \sigma_i &= \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)(i-\mu_i)^2} \\ \sigma_i &= \sqrt{2.2254+0.0752+...+0.0025+0.0017+...+1.7944} \\ \sigma_i &= 2.4624 \end{aligned}$$

Nilai standar deviasi kolom (σ_j) dihitung dengan cara yang sama.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.7 Nilai Standar Deviasi Kolom

Kolom ke-j	$\mathbf{P(i,j)(j-\mu_j)^2}$
$1,1$	2.2246
$1,2$	0.0336
$\vdots$	$\vdots$
$5,7$	0.0021
$6,7$	0.0570
$\vdots$	$\vdots$
$8,8$	1.7950
Standar Deviasi Kolom	**2.4619**

Perhitungan standar deviasi kolom tidak jauh berbeda dengan perhitungan standar deviasi baris, yaitu dengan menghitung terlebih dahulu nilai $P(i,j)(i-\mu_i)^2$ seperti berikut:

$$\begin{aligned} i,j &= P(i,j)(j-\mu_j)^2 & i,j &= P(i,j)(j-\mu_j)^2 \\ 5,7 &= 0.0002(7-4.017)^2 & 8,8 &= 0.1132(8-4.017)^2 \\ 5,7 &= 0.0021 & 8,8 &= 1.7950 \end{aligned}$$

dengan $5,7$ menunjukkan nilai dari baris ke lima pada kolom ke tujuh. Dengan melihat Tabel 4.7, semua nilai $P(i,j)(i-\mu_i)^2$ dihitung dan kemudian menjumlahkan smeua nilai. Oleh karena itu didapatkan nilai standar deviasi σ_j:

$$\begin{aligned} \sigma_j &= \sqrt{\sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)(i-\mu_i)^2} \\ \sigma_j &= \sqrt{2.2246+0.0336+...+0.0021+0.0570+...+1.7950} \\ \sigma_j &= 2.4619 \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Nilai μ_i, μ_j, σ_i, dan σ_j yang telah didapatkan digunakan untuk menghitung nilai korelasi. Nilai korelasi dapat dihitung dengan proses perhitungan seperti berikut:

$$\begin{aligned}
\text{Korelasi} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{(i-\mu_i)(j-\mu_j)P(i,j)}{\sigma_i\sigma_j} \\
&= \frac{(1-4.018)(1-4.017)0.2444}{(2.4624)(2.4619)} + \frac{(1-4.018)(2-4.017)0.0083}{(2.4624)(2.4619)} \\
&\quad +...+\frac{(5-4.018)(7-4.017)0.0002}{(2.4624)(2.4619)}+... \\
&\quad +\frac{(8-4.018)(8-4.017)0.1132}{(2.4624)(2.4619)} \\
\text{Korelasi} &= 0.3408+0.0077+...+0.0001+0.2749 \\
&= 0.9784
\end{aligned}$$

c. Energi

Tabel 4.8 Nilai Energi

	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**
1	0.0597	0.0001	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
2	0.0001	0.0058	0.0002	0.0000	0.0000	0.0000	0.0000	0.0000
3	0.0000	0.0002	0.0047	0.0004	0.0000	0.0000	0.0000	0.0000
4	0.0000	0.0000	0.0004	0.0061	0.0004	0.0000	0.0000	0.0000
5	0.0000	0.0000	0.0000	0.0004	0.0049	0.0002	0.0000	0.0000
6	0.0000	0.0000	0.0000	0.0000	0.0002	0.0070	0.0000	0.0000
7	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0073	0.0000
8	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0128
Energi	**0.1109**							

Hasil perhitungan ekstraksi fitur energi dari citra MRI otak dinyatakan

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

dalam Tabel 4.8 beserta dengan contoh perhitungannya.

$$\begin{aligned}\text{Energi} &= \sum_{i=1}^{N}\sum_{j=1}^{N} P(i,j)^2 \\ &= P(1,1)^2 + ... + P(4,6)^2 + ... + P(8,8)^2 \\ &= 0.2444^2 + ... + 0.0008^2 + ... + 0.1132^2 = 0.1109\end{aligned}$$

Nilai-nilai dari Tabel 4.8 merupakan hasil dari perhitungan $P(i,j)^2$. Kemudian nilai-nilai tersebut dijumlahkan untuk didapatkan nilai energi sebesar 0.1109.

d. Homogenitas

Tabel 4.9 Nilai Homogenitas

$(\mathbf{i},\mathbf{j})$	$(\mathbf{i}-\mathbf{j})^2$	$\mathbf{P}(\mathbf{i},\mathbf{j})$	$\frac{\mathbf{P}(\mathbf{i},\mathbf{j})}{\mathbf{1}+(\mathbf{i}-\mathbf{j})^2}$
$(1,1)$	0	0.2444	0.2444
$(1,2)$	1	0.0083	0.0041
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(4,5)$	1	0.0196	0.0098
$(4,6)$	4	0.0008	0.0002
$\vdots$	$\vdots$	$\vdots$	$\vdots$
$(8,8)$	0	0.1132	0.1132
Homogenitas			**0.9062**

Hasil perhitungan nilai homogenitas pada citra MRI otak dinyatakan dalam Tabel 4.9. Nilai-nilai dari Tabel 4.9 merupakan hasil dari perhitungan $\frac{\mathbf{P}(\mathbf{i},\mathbf{j})}{\mathbf{1}+(\mathbf{i}-\mathbf{j})^2}$. Kemudian nilai-nilai tersebut dijumlahkan untuk didapatkan nilai energi sebesar 0.9062. Berikut contoh perhitungannya:

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$
\begin{aligned}
\text{Homogenitas} &= \sum_{i=1}^{N}\sum_{j=1}^{N}\frac{P(i,j)}{1+(i-j)^2} \\
&= \frac{0.2444}{1+(1-1)^2}+\frac{0.0083}{1+(1-2)^2}...+\frac{0.0196}{1+(4-5)^2} \\
&\quad +\frac{0.0008}{1+(4-6)^2}+...+\frac{0.1132}{1+(8-8)^2} \\
&= 0.2444+0.0041+...+0.0098+0.0002+...+0.1132 \\
&= 0.9062
\end{aligned}
$$

Karena data yang digunakan dalam penelitian ini cukup banyak dan memiliki ukuran matriks piksel yang besar, maka perhitungan dapat dipermudah dengan bantuan *function* pada Python, yaitu dengan *syntax greycomatrix* untuk mendapatkan matriks *co-occurence* dan *greycoprops* untuk mendapatkan fitur kontras, koreasi, energi, dan homogenitas. Dalam penelitian ini akan dilakukan perbandingan pada setiap sudut orientasi GLCM sebagai data input untuk proses klasifikasi. Berdasarkan 120 data MRI otak yang digunakan, akan ditampilkan hasil ekstraksi fitur dengan sudut orientasi yang berbeda. Data penelitian yang telah diekstraksi fitur dengan menggunakan GLCM sudut orientasi $\theta = 0^o$ dan jarak piksel $d_p = 1$ ditunjukkan pada Tabel 4.10. 1cmData penelitian yang telah diekstraksi fitur dengan menggunakan GLCM sudut orientasi $\theta = 45^o$ dan jarak piksel $d_p = 1$ ditunjukkan pada Tabel 4.11. Data penelitian yang telah diekstraksi fitur dengan menggunakan GLCM sudut orientasi $\theta = 90^o$ dan jarak piksel $d_P = 1$ ditunjukkan pada Tabel 4.12. Data penelitian yang telah diekstraksi fitur dengan menggunakan GLCM sudut orientasi $\theta = 135^o$ dan jarak piksel $d_p = 1$ ditunjukkan pada Tabel 4.13.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

a. Uji coba dengan sudut orientasi $\theta = 0^o$ dan jarak piksel $d_p = 1$.

Tabel 4.10 Ekstraksi fitur GLCM sudut orientasi $\theta = 0^o$ dan jarak piksel $d_p = 1$

Citra ke-	Kontras	Korelasi	Energi	Homogenitas
1	0.26079	0.96813	0.10229	0.90535
2	0.27630	0.96479	0.09765	0.89790
3	0.35464	0.95587	0.09455	0.88062
4	0.20197	0.97494	0.10618	0.92069
⋮	⋮	⋮	⋮	⋮
117	0.19716	0.98196	0.09766	0.92957
118	0.22140	0.97294	0.10122	0.91329
119	0.21239	0.97321	0.10726	0.92492
120	0.23398	0.97914	0.09406	0.91207

b. Uji coba dengan sudut orientasi $\theta = 45^o$ dan jarak piksel $d_p = 1$.

Tabel 4.11 Ekstraksi fitur GLCM sudut orientasi $\theta = 45^o$ dan jarak piksel $d_p = 1$

Citra ke-	Kontras	Korelasi	Energi	Homogenitas
1	0.38631	0.95271	0.09372	0.87735
2	0.42767	0.94539	0.08797	0.86548
3	0.49453	0.93835	0.08728	0.85468
4	0.28085	0.96510	0.09791	0.89770
⋮	⋮	⋮	⋮	⋮
117	0.26558	0.97566	0.09165	0.91127
118	0.30692	0.96241	0.09396	0.89156
119	0.31595	0.96000	0.09766	0.89877
120	0.35561	0.96825	0.08773	0.89062

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

c. Uji coba dengan sudut orientasi $\theta = 90^o$ dan jarak piksel $d_p = 1$.

Tabel 4.12 Ekstraksi fitur GLCM sudut orientasi $\theta = 90^o$ dan jarak piksel $d_p = 1$

Citra ke-	Kontras	Korelasi	Energi	Homogenitas
1	0.24967	0.96950	0.10237	0.90528
2	0.28228	0.96404	0.09773	0.89735
3	0.29017	0.96392	0.09882	0.89539
4	0.17280	0.97857	0.10941	0.92913
⋮	⋮	⋮	⋮	⋮
117	0.15700	0.98563	0.10023	0.93762
118	0.18281	0.97766	0.10713	0.92860
119	0.21250	0.97320	0.10687	0.92388
120	0.23746	0.97883	0.09613	0.91874

d. Uji coba dengan sudut orientasi $\theta = 135^o$ dan jarak piksel $d_p = 1$.

Tabel 4.13 Ekstraksi fitur GLCM sudut orientasi $\theta = 135^o$ dan jarak piksel $d_p = 1$

Citra ke-	Kontras	Korelasi	Energi	Homogenitas
1	0.38466	0.95291	0.09327	0.87574
2	0.41928	0.94646	0.08826	0.86651
3	0.48141	0.93998	0.08713	0.85480
4	0.28075	0.96511	0.09840	0.89894
⋮	⋮	⋮	⋮	⋮
117	0.27015	0.97524	0.09221	0.91292
118	0.29802	0.96350	0.09468	0.89382
119	0.30516	0.96137	0.09756	0.89884
120	0.35957	0.96790	0.08571	0.88324

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Percobaan ekstraksi fitur dengan metode GLCM pada jarak piksel $d_p = 1$ dan beberapa sudut orientasi yang berbeda, menghasilkan nilai fitur yang berbeda pula. Oleh sebab itu akan dilakukan percobaan dari setiap sudut orientasi pada proses pemilihan model klasifikasi.

4.2. Pemilihan Model dan Proses *Learning*

Penelitian ini menggunakan metode klasifikasi SVM dengan parameter input menggunakan hasil ekstraksi fitur GLCM. Pada SVM sendiri juga terdapat parameter-parameter lain untuk membantu menemukan *hyperplane* terbaik seperti parameter C, γ, r, dan d. Parameter-parameter tersebut merupakan parameter kernel yang ditentukan sendiri nilainya oleh peneliti. Pada tahap pertama dilakukan klasifikasi citra MRI otak menggunakan SVM dengan parameter *default*. Kemduian hasil dari klasifikasi SVM dengan parameter *default* akan dibandingkan dengan parameter yang diperoleh dari hasil pengoptimasian menggunakan FFA.

4.2.1. Klasifikasi SVM

Proses klasifikasi menggunakan SVM dilakukan dengan tahap pelatihan dan pengujian terhadap citra MRI otak. Tahap pelatihan merupakan tahap untuk mendapatkan model sistem, sedangkan pengujian merupakan tahap untuk menguji keakuratan sistem yang telah didaptkan pada proses pelatihan. Dalam penelitian ini digunakan 120 data MRI otak yang telah diproses dengan metode ekstraksi fitur GLCM untuk mendapatkan data statistik dari citra MRI otak. Data statistik tersebut merupakan data fitur kontras, korelasi, energi dan homogenitas. Empat fitur yang didapatkan dari setiap citra dapat merepresentasikan bentuk citra dan membedakan citra satu dengan yang lainnya. Fitur-fitur tersebut yang digunakan sebagai parameter input untuk proses SVM.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Sebelum melakukan proses *learning* menggunakan SVM, data citra yang telah diekstraksi dibagi menjadi data pelatihan dan data uji. Dalam penelitian ini, pembagian data dilakukan dengan menggunakan *k-fold* $= 5$, sehingga diperoleh pembagian data *training* sebanyak 96 data yang terdiri dari 48 data normal dan 48 data tumor dan data *testing* sebanyak 24 data yang terdiri dari 12 data normal dan 12 data tumor. Pada proses klasifikasi, data dipisahkan secara biner, yaitu kelas normal dan kelas tumor dengan pendekatan kernel linear, kernel RBF, kernel *polynomial*, dan kernel *sigmoid*. Berikut akan dijelaskan sampel data perhitungan proses klasifikasi SVM menggunakan kernel *polynomial* dengan sampel data dari ekstraksi fitur GLCM pada sudut orientasi $\theta = 45^o$ sebanyak 6 data. Data klasifikasi disajikan dalam bentuk tabel beserta nilai labelnya sebagai berikut:

Tabel 4.14 Sampel Data untuk Klasifikasi SVM

Kontras(x_1)	**Korelasi(x_2)**	**Energi(x_3)**	**Homogenitas(x_4)**	**Y**	**Label**
0.386	0.953	0.094	0.877	Normal	0
0.428	0.94539	0.088	0.866	Normal	0
0.494	0.938	0.087	0.855	Normal	0
0.307	0.962	0.094	0.892	Tumor	1
0.316	0.960	0.098	0.899	Tumor	1
0.356	0.968	0.088	0.891	Tumor	1

Pada proses klasifikasi SVM, dibutuhkan beberapa parameter input untuk membantu proses menemukan model *hyperplane* terbaik. Parameter input tersebut diantaranya, yaitu:

Sampel data *training* : $x = x_1, x_2, ..., x_n$

Label data *training* : $y = y_1, y_2, ..., y_n \subset \pm 1$

Jenis dan Parameter Kernel : *Polynomial* dan $\gamma = 1.8$, $r = 0.0$, $d = 3$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Parameter input kernel yang dicontohkan di atas merupakan nilai yang didapatkan dari *default software* pemrograman pada *Python*. Setelah dilakukan input parameter untuk proses klasifikasi dengan metode SVM, dilakukan perhitungan matriks kernel dengan memisalkan matriks data *training* X. Selanjutnya untuk mendapatkan matriks kernel, dilakukan perkalian matriks *training* dengan transposenya.

$$X \times X^T$$

$$\begin{pmatrix} 0.386 & 0.953 & \cdots & 0.877 \\ 0.428 & 0.945 & \cdots & 0.866 \\ \vdots & \vdots & \ddots & \vdots \\ 0.356 & 0.968 & \cdots & 0.891 \end{pmatrix} \times \begin{pmatrix} 0.386 & 0.428 & \cdots & 0.356 \\ 0.953 & 0.945 & \cdots & 0.968 \\ \vdots & \vdots & \ddots & \vdots \\ 0.877 & 0.866 & \cdots & 0.891 \end{pmatrix}$$

Hasil perhitungan matriks di atas dapat digunakan untuk menghitung nilai matriks kernel K dengan contoh perhitungan per entri baris dan kolom sebagai berikut:

$x_1.x_1^T$

$= [\ 0.386 \quad 0.953 \quad 0.094 \quad 0.877\ ][\ 0.386 \quad 0.953 \quad 0.094 \quad 0.877\ ] = 1.8354$

$$\begin{aligned} K(x_i, x_j) &= (\gamma\,(x_i, x_j) + r)^d \\ K(1,1) &= (1.8\,(1.8354) + 0.0)^3 = 36.0597 \end{aligned}$$

$x_1.x_2^T$

$= [\ 0.386 \quad 0.953 \quad 0.094 \quad 0.877\ ][\ 0.428 \quad 0.94539 \quad 0.088 \quad 0.866\ ] = 1.8335$

$$\begin{aligned} K(x_i, x_j) &= (\gamma\,(x_i, x_j) + r)^d \\ K(1,2) &= (1.8\,(1.8335) + 0.0)^3 = 35.9450 \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

$$\vdots$$

$x_6.x_6^T$

$= [\ 0.356 \quad 0.968 \quad 0.088 \quad 0.891\ ][\ 0.356 \quad 0.968 \quad 0.088 \quad 0.891\ ]$

$= 1.8649$

$$\begin{aligned} K(x_i, x_j) &= (\gamma\,(x_i, x_j) + r)^d \\ K(6,6) &= (1.8\,(1.8649) + 0.0)^3 \\ &= 37.8235 \end{aligned}$$

Dari perhitungan di atas, diperoleh matriks K, yaitu:

$$K = \begin{pmatrix} 36.0597 & 35.9450 & 36.5115 & 35.5355 & 35.9965 & 36.8928 \\ 35.9450 & 35.9443 & 36.6864 & 35.2170 & 35.6919 & 36.6864 \\ 36.5115 & 36.6864 & 37.7195 & 35.4525 & 35.9613 & 37.1222 \\ 35.5355 & 35.2170 & 35.4525 & 35.3991 & 35.8216 & 36.5252 \\ 35.9965 & 35.6919 & 35.9613 & 35.8216 & 36.2564 & 36.9803 \\ 36.8928 & 36.6864 & 37.1222 & 36.5252 & 36.9803 & 37.8235 \end{pmatrix}$$

Hasil matriks kernel K yang telah didapatkan akan digunakan untuk mencari nilai pengali *lagrange* menggunakan Persamaan 2.40, yaitu:

$$\text{maks}\ \ L_p = \sum_{i=1}^{N} \alpha_i - \frac{1}{2} \sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_i y_i \alpha_j y_j (x_i'.x_j')$$

Dimana perkalian dari $x_i'.x_j'$ merupakan hasil dari matriks kernel K dan y adalah label data $\{-1, -1, -1, 1, 1, 1\}$. Dimana -1 merupakan data normal dan 1 adalah data tumor. Persamaan 2.40 merupakan persamaan dualitas yang sulit diselesaikan secara manual. Oleh karena itu, dibutuhkan bantuan *software* dalam

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

menyelesaikan persamaan tersebut. Setelah mendapatkan nilai pengali *lagrange* (a), didapatkan nilai w dan b. Akan tetapi nilai-nilai tersebut tidak dapat ditampilkan secara tegas karena sulitnya perhitungan. Ketiga nilai tersebut merupakan pembentuk model SVM yang digunakan untuk proses klasifikasi. Berikut akan disajikan hasil klasifikasi citra MRI otak menggunakan SVM berdasarkan kernel dalam Tabel 4.15 dengan Acu(akurasi), Spes(spesifisitas), dan Sens(sensivisitas).

Tabel 4.15 Hasil Klasifikasi Citra MRI Otak Berdasarkan Kernel pada SVM

Data	Kernel	Acu	Spes	Sens
GLCM 0°	Linear	70.83%	41.67%	100.00%
	RBF	79.17%	58.33%	100.00%
	Sigmoid	70.83%	41.67%	100.00%
	Polynomial	79.17%	58.33%	100.00%
GLCM 45°	Linear	79.17%	58.33%	100.00%
	RBF	79.17%	58.33%	100.00%
	Sigmoid	79.17%	66.67%	91.67%
	Polynomial	79.17%	58.33%	100.00%
GLCM 90°	Linear	66.67%	33.33%	100.00%
	RBF	79.17%	58.33%	100.00%
	Sigmoid	66.67%	33.33%	100.00%
	Polynomial	79.17%	58.33%	100.00%
GLCM 135°	Linear	79.17%	58.33%	100.00%
	RBF	79.17%	58.33%	100.00%
	Sigmoid	79.17%	58.33%	100.00%
	Polynomial	79.17%	58.33%	100.00%

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pada Tabel 4.15, proses *learning* yang dilakukan didasarkan pada data GLCM dengan empat sudut dan empat kernel pada SVM yang terdiri dari kernel linear, RBF, *sigmoid*, dan *polynomial*. Pada kernel linear, parameter *default* yang digunakan adalah parameter $C = 1$. Pada kernel RBF, parameter *default* yang digunakan adalah parameter $C = 1$ dan $\gamma = 1.8$. Pada kernel *sigmoid*, parameter *default* yang digunakan adalah parameter $C = 1$, $\gamma = 1.8$, dan $r = 0.0$. Pada kernel *polynomial*, parameter *default* yang digunakan adalah parameter $C = 1$, $\gamma = 1.8$, $r = 0.0$, dan $d = 3$. Hasil *learning* terbaik dari proses klasifikasi menggunakan SVM yaitu pada akurasi sebesar 79.17% dengan sepesifisitas sebesar 58.33%, dan sensitivitas sebesar 100.00%. Perolehan dari klasifikasi SVM dengan parameter *default* hanya memberikan akurasi 79.17%. Secara penilaian, nilai tersebut termasuk dalam kategori cukup. Oleh karena itu, dalam penelitian ini dilakukan pengoptimasian parameter SVM untuk mendapatkan *hyperplane* yang terbaik.

Hyperplane terbaik dan optimal akan memebrikan nilai akurasi sistem yang tinggi. Jika akurasi tinggi, maka sistem dapat mengklasifikasikan data sesuai dengan kelasnya secara baik. Pengoptimasian parameter input pada SVM dilakukan dengan menggunakan metode *firefly optimization*. Hasil optimasi dari *firefly* yang akan digunakan untuk proses *learning* untuk menemukan model terbaik. Proses ini disebut sebagai proses klasifikasi FFA-SVM.

4.2.2. *Firefly Algorithm*

Algoritma *firefly* (FFA) merupakan algoritma optimasi yang terinspirasi dari perilaku kunang-kunang. Optimasi ini dilakukan untuk menentukan kombinasi dari nilai parameter input SVM agar hasil dari nilai akurasi sistem yang digunakan besar. Proses dari pengoptimalan parameter SVM dilakukan sebelum melakukan proses

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

klasifikasi. Berikut adalah contoh tahapan hasil proses pengoptimalan parameter C pada SVM.

1. Inisialisasi parameter FFA

   Pada proses FFA, perlu dilakukan inisialisasi terlebih dahulu parameter-parameetr dalam proses FFA, diantaranya yaitu jumlah populasi *firefly*, dimensi, *max* iterasi, parameter α, parameter β, parameter γ, batas atas, dan batas bawah nilai parameter yang akan dioptimasi. Berikut adalah nilai-nilai inisialisasi proses FFA yang disajikan pada Tabel 4.16. Berdasar Tabel 4.16, populasi yang digunakan sebanyak 15 *firefly*.

Tabel 4.16 Inisialisasi Parameter *Firefly*

Parameter	Nilai
Populasi *firefly*	7
Dimensi	1
Max iterasi	100
α	0.2
β_0	1.0
γ	1.0
Batas bawah	100
Batas atas	2000

2. Menentukan nilai setiap populasi *firefly*

   Setelah proses inisialisasi seperti pada Tabel 4.16, dilakukan penentuan nilai populasi *firefly* secara acak. Hal ini dilakukan agar *firefly* menempati titik secara acak. Hasil dari pengacakan nilai populasi *firefly* disajikan pada Tabel 4.17.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Tabel 4.17 Nilai Acak Populasi *Firefly*

Firefly	1	2	3	4	5	6	7
x	777.32	187.29	1526.40	289.53	1606.57	900.90	1505.65
f(x)/I	77655	18710	152488	28925	160496	90000	150415

x menunjukkan posisi *firefly* di titik x, dan $f(x)$ menunjukkan nilai intensitas cahaya (I) pada *firefly* yang dapat dihitung menggunakan Persamaan 2.44 sebagai berikut:

$$\begin{aligned} f(x) &= x_b + (x_a - x_b)g \\ f(x_1) &= 100 + (1000 - 100)777.32 = 77655 \\ f(x) &= x_b + (x_a - x_b)g \\ f(x_2) &= 100 + (1000 - 100)187.29 = 18710 \end{aligned}$$

3. Menyortir nilai intensitas cahaya pada *firefly* dari kecil ke besar.

Tabel 4.18 Hasil Penyortiran Nilai Intensitas Cahaya *Firefly*

Sortir *I*	18710	28925	77655	90000	150415	152488	160496
Firefly	2	4	1	6	7	3	5
x	187.29	289.53	777.32	900.90	1505.65	1526.40	1606.57

Penyortiran nilai intensitas cahaya *firefly* dilakukan untuk mendapatkan nilai fbest atau *firefly* terbaik. Dalam penelitian ini, kasus yang diselesaikan adalah memaksimalkan nilai akurasi yang dicapai oleh sistem klasifikasi dengan mengoptimalkan nilai parameter SVM. Oleh karena itu, *firefly* dipilih berdasarkan *firefly* dengan tingkat intensitas cahaya terbesar. Nilai *x* dari *firefly*

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

tersebut yang akan digunakan sebagai input nilai C pada parameter SVM. Tabel 4.18 adalah hasil dari penyortiran nilai intensitas cahaya *firefly*. Tabel 4.18 menunjukkan bahwa nilai minimal dari intensitas cahaya *firefly* dimiliki oleh *firefly* ke 1 dengan nilai intensitas cahaya sebesar 12044.

4. Menghitung nilai delta dan perubahan nilai α.

   Nilai α merupakan nilai α sebelumnya. Nilai ini dihitung setiap kali mulai perulangan dari maksimum perulangan/iterasi yang telah ditetapkan. Nilai delta merupakan pengurangan nilai pengacakan yang dapat dihitung dengan Persamaan 4.2.

$$delta = 1 - \frac{\left(\frac{10^{-4}}{0.9}\right)^1}{\left(\frac{iterasi}{populasi}\right)} \tag{4.2}$$

   Maka, diperoleh nilai $delta$ sebagai berikut:

$$\begin{aligned} delta &= 1 - \frac{\left(\frac{10^{-4}}{0.9}\right)^1}{\left(\frac{iterasi}{populasi}\right)} \\ delta &= 1 - \frac{\left(\frac{10^{-4}}{0.9}\right)^1}{\left(\frac{100}{7}\right)} \\ &= 0.999992222 \end{aligned}$$

   Kemudian perubahan nilai α dapat dihitung menggunakan Persamaan 4.3.

$$\alpha_{new} = (1 - delta) \times \alpha \tag{4.3}$$

   Berikut nilai α_{new} yang didapat dengan:

$$\begin{aligned} \alpha_{new} &= (1 - delta) \times \alpha \\ \alpha_{new} &= (1 - 0.999992222) \times 0.2 = 1.55556 \times 10^{(-6)} \end{aligned}$$

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

5. Memindahkan *firefly* menuju *firefly* dengan tingkat intensitas cahaya yang lebih terang.

Pada tahap ini, *firefly* yang memiliki intensitas lebih rendah akan dipindahkan menuju *firefly* dengan tingkat intensitas cahaya yang lebih terang. Jika terdapat kondisi tersebut, maka akan dilakukan proses perhitungan jarak antar kedua *firefly* dan kemudian dihitung perubahan posisinya. Tabel 4.19 adalah contoh dari proses pergerakan *firefly* yang telah di sortir tingkat intensitas cahayanya.

Tabel 4.19 Pergerakan *Firefly* ke-1

I_x	I_y	*r*	Posisi Baru
18710	18710	-	-

Pada Tabel 4.19 menunjukkan bahwa *firefly* tidak mengalami pergerakan karena dibandingkan terhadap dirinya sendiri. Oleh karena itu tidak ada perbedaan intensitas cahaya pada *firefly* tersebut. Selanjutnya akan dilakukan perbandingan terhadap *firefly* lainnya yang disajikan dalam Tabel 4.20.

Tabel 4.20 Pergerakan *Firefly* ke-2

I_x	I_y	*r*	Posisi Baru
18710	18710	-	-
	28925	102.25	100

Pada Tabel 4.20 nilai intensitas *firefly x* lebih kecil dari *firefly y*, sehingga dilakukan perhitungan jarak antar kedua *firefly* dan pergerakannya menggunakan Persamaan 2.27 dan Persamaan 2.29. Berikut adalah contoh

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

perhitungannya.

$$
\begin{aligned}
r_{i,j} &= \sqrt{x_{i,k} - x_{j,k})^2} \\
r_{1,2} &= \sqrt{(187.29 - 289.53)^2} \\
&= 102.25 \\
x_i &= x_i(1 - \beta_o e^{\gamma r_{i,j}^2}) + xo_j(\beta_o e^{\gamma r_{i,j}^2}) + \alpha \left(rand - \frac{1}{2} \right) \\
x_1 &= 187.29(1 - 1 \times e^{1.0(102.25)^2}) + 187.29(1 \times e^{1.0(102.25)^2}) + 1.55556 \times 10^{(-6)} \left(8 - \frac{1}{2} \right) \\
&= -\infty = 100
\end{aligned}
$$

Karena nilai x kurang dari batas bawah, maka secara otomatis menjadi dirubah sebesar batas bawah. Dalam perhitungan di atas dimisalkan nilai *rand* nya adalah 8. Setelah *firefly* dipindahkan, maka didapatkan nilai *firefly* terbaru seperti pada Tabel 4.21

Tabel 4.21 *Firefly* Baru 1

Sortir *I*	18710	28925	77655	90000	150415	152488	160496
Firefly	2	4	1	6	7	3	5
x	100	289.53	777.32	900.90	1505.65	1526.40	1606.57

Kemudian menghitung kembali perpindahan *firefly* seperti pada perhitungan sebelumnya. Hasil pergerakan *firefly* ke-3 disajikan pada Tabel 4.22.

Tabel 4.22 Pergerakan *Firefly* ke-3

I_x	I_y	***r***	**Posisi Baru**
18710	18710	-	-
	28925	102.25	100
	77655	677.32	100

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Hasil perhitungan jarak antar kedua *firefly* dan pergerakannya menggunakan adalah contoh perhitungannya.

$$
\begin{aligned}
r_{i,j} &= \sqrt{x_{i,k} - x_{j,k})^2} \\
r_{1,3} &= \sqrt{(100 - 777.32)^2} \\
&= 677.32 \\
x_i &= x_i(1 - \beta_o e^{\gamma r_{i,j}^2}) + xo_j(\beta_o e^{\gamma r_{i,j}^2}) + \alpha \left(rand - \frac{1}{2} \right) \\
x_1 &= 100(1 - 1 \times e^{1.0(677.32)^2}) + 1526.40(1 \times e^{1.0(677.32)^2}) + 1.55556 \times 10^{(-6)} \left(8 - \frac{1}{2} \right) \\
&= -\infty = 100
\end{aligned}
$$

Setelah *firefly* dipindahkan, maka didapatkan nilai *firefly* terbaru seperti pada Tabel 4.23

Tabel 4.23 *Firefly* Baru 2

Sortir *I*	18710	28925	77655	90000	150415	152488	160496
Firefly	2	4	1	6	7	3	5
x	100	289.53	777.32	900.90	1505.65	1526.40	1606.57

Proses pergerakan *firefly* dilakukan terus menerus sampai pada *firefly* terakhir dan sampai pada batas maksimum iterasi yang telah ditetapkan sebelumnya untuk mendapatkan nilai optimal dari fungsi objektif yang diharapkan.

4.2.3. FFA-SVM

Penelitian ini menggunakan sudut orientasi 0°, 45°, 90°, dan 135° yang diterapkan pada proses ekstraksi fitur GLCM, mengoptimasi nilai parameter input SVM, dan menerapkan kernel linear, RBF, *Sigmoid*, dan *Polynomial* untuk proses

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

SVM. Pada penelitian ini dilakukan beberapa kali pengoptimasian parameter dengan percobaan jumlah iterasi berdasarkan sudut orientasi GLCM dan macam-macam kernel. Pada penelitian ini, pengoptimasian parameter digunakan kasus maksimalisasi, atau dipilih nilai cahaya yang paling terang sebagai *firefly* terbaik. Dalam penelitian ini, dilakukan pengoptimasian parameter C dengan batas $100 - 2000$, parameter γ dengan batas $1 - 20$, parameter d dengan batas $0.00001 - 0.0005$, dan parameter r dengan batas $3 - 7$. Hasil percobaan optimasi parameter SVM dengan *firefly* berdasarkan banyaknya iterasi disajikan dalam Tabel 4.24 hingga Tabel 4.27.

Tabel 4.24 Nilai Parameter C pada Kernel Linear dan Sudut Orientasi GLCM

Data	**n Iterasi**	**C**	Akurasi
GLCM 0°	35	1925.532	83.33
	50	1828.664	83.33
	100	1863.234	83.33
GLCM 45°	35	1910.687	83.33
	50	1670.061	83.33
	100	1821.212	83.33
GLCM 90°	35	1916.315	83.33
	50	1814.127	83.33
	100	1561.366	83.33
GLCM 135°	35	1967.817	83.33
	50	1948.562	83.33
	100	1992.511	83.33

Tabel 4.24 menunjukkan bahwa percobaan dilakukan dengan tiga iterasi, dimana iterasi tersebut terdiri dari 35 iterasi, 50 iterasi, dan 100 iterasi. Dari hasil

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

yang diperoleh, semua nilai akurasi pada percobaan C diperoleh sebessar 83.33%. Hal ini terjadi karena nilai maksimal dari *firefly* mendekati nilai batas atas, sehingga semua akurasi yang didapatkan pada setiap percobaan data yang berdasarkan pada jumlah iterasi besarnya sama. Selanjutnya akan disajikan hasil percobaan optimasi parameter SVM menggunakan kernel (*Radial Basis Function*) RBF dengan semua sudut orientasi GLCM pada Tabel 4.25.

Tabel 4.25 Nilai Parameter C dan γ pada Kernel RBF dan Sudut Orientasi GLCM

Data	n Iterasi	C	γ	Akurasi
GLCM 0°	35	1771.744	19.367	79.17
	50	1834.330	19.688	79.17
	100	1595.364	19.574	79.17
GLCM 45°	35	1896.143	19.635	75.00
	50	1930.012	19.538	79.17
	100	1937.791	16.806	83.33
GLCM 90°	35	1897.086	19.953	87.50
	50	1506.966	17.223	91.67
	100	1874.654	19.191	95.83
GLCM 135°	35	1657.644	18.679	83.33
	50	1991.722	18.009	83.33
	100	1697.011	19.376	83.33

Tabel 4.25 menunjukkan bahwa pada data GLCM 0° dan GLCM 135° nilai akurasi yang diperoleh sama, meskipun dengan percobaan iterasi yang berbeda. Sedangkan pada data GLCM 45° dan GLCM 90° nilai akurasi yang diperoleh berbeda dari setiap percobaan iterasi. Percobaan tersebut menunjukkan bahwa semakin banyak iterasi, maka semakin baik nilai akurasi yang diperoleh. Secara

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

kesuluruhan dapat dikatakan bahwa hasil akurasi tidak dipengaruhi pada jumlah iterasi saja, melainkan juga pada data yang digunakan untuk proses *training* dan *testing*. Nilai akurasi tertinggi dengan kernel RBF diperoleh sebesar 95.83. Akurasi tersebut diperoleh pada proses *learning* menggunakan GLCM dengan sudut orientasi sebesar 90°. Pada akurasi ini, jumlah iterasi yang digunakan sebanyak 100 iterasi dengan nilai parameter $C = 1874.654$ dan $\gamma = 19.191$. Selanjutnya akan disajikan hasil percobaan optimasi parameter SVM menggunakan kernel *Sigmoid* dengan semua sudut orientasi GLCM pada Tabel 4.26.

Tabel 4.26 Nilai Parameter C, γ, dan r pada Kernel *Sigmoid* dan Sudut Orientasi GLCM

Data	n Iterasi	C	γ	r	Akurasi
GLCM 0°	35	1847.282	19.067	0.0005	50.00
	50	1792.077	19.567	0.0005	50.00
	100	1906.999	19.915	0.0005	50.00
GLCM 45°	35	1781.045	19.586	0.0005	50.00
	50	1989.806	19.368	0.0005	50.00
	100	1945.669	18.722	0.0005	50.00
GLCM 90°	35	1498.203	19.955	0.0005	50.00
	50	1888.085	16.750	0.0005	50.00
	100	1968.331	16.996	0.0005	50.00
GLCM 135°	35	1855.582	19.466	0.0005	50.00
	50	1874.559	17.823	0.0005	50.00
	100	1991.522	19.171	0.0005	50.00

Tabel 4.26 menunjukkan bahwa hasil percobaan pada kernel *sigmoid* tidak jauh berbeda dengan hasil percobaan pada kernel linear. Semua nilai akurasi yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

diperoleh besarnya sama. Akan tetapi, pada yang diperoleh hanya sebesar 50.00. Akurasi ini menunjukkan bahwa sistem tidak dapat mengklasifikasi dengan baik semua data positif atau semua data negatif. Pada proses *learning*, kombinasi parameter C, γ, dan r tidak dapat menemukan *hyperplane* yang tepat. Selain itu, nilai yang digunakan merupakan nilai dari intensitas cahaya yang maksimal. Oleh karena itu, semua nilai mendekati batas atas yang telah ditentukan. Selanjutnya akan disajikan hasil percobaan optimasi parameter SVM menggunakan kernel *Polynomial* dengan sudut orientasi GLCM pada Tabel 4.27.

Tabel 4.27 Nilai Parameter C, γ, r, dan d pada Kernel *Polynomial* dan Sudut Orientasi GLCM

Data	n Iterasi	C	γ	r	d	Akurasi
GLCM 0°	35	1964.467	19.859	0.0005	6.992	66.67
	50	1839.557	19.093	0.0005	6.684	75.00
	100	1996.870	19.824	0.0005	6.968	79.17
GLCM 45°	35	1952.011	19.701	0.0005	6.911	83.33
	50	1941.914	18.247	0.0005	6.896	70.83
	100	1970.044	19.976	0.0005	6.568	83.33
GLCM 90°	35	1980.323	19.895	0.0005	6.592	83.33
	50	1770.286	19.257	0.0005	6.838	91.67
	100	1796.125	19.765	0.0005	6.956	95.83
GLCM 135°	35	1965.514	18.934	0.0005	6.024	83.33
	50	1828.732	17.522	0.0005	6.850	75.00
	100	1936.169	19.454	0.0005	6.671	83.33

Tabel 4.27 menunjukkan bahwa pada data GLCM 45° dan GLCM 135° nilai akurasi yang diperoleh tidak dipengaruhi oleh banyaknya iterasi yang

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

digunakan. Berbeda dengan percobaan pada kernel RBF, percobaan ini menunjukkan jika semakin banyak nilai iterasi yang ditentukan, maka nilai akurasi yang didapatkan tidak pasti semakin besar. Sedangkan pada data GLCM 90° dan GLCM 0° akurasinya terus meningkat dengan bertambahnya batas iterasi yang ditentukan. Oleh karena itu dapat dikatakan bahwa nilai akurasi yang diperoleh tergantung pada kombinasi parameter C, γ, dan r, dan d. Nilai akurasi tertinggi dengan kernel *polynomial* diperoleh sebesar 95.83. Akurasi tersebut diperoleh pada proses *learning* menggunakan GLCM dengan sudut orientasi sebesar 90°. Pada akurasi ini, nilai iterasi yang digunakan sebesar 100 iterasi dengan nilai parameter $C = 1796.125$, $\gamma = 19.765$, $r = 0.0005$ dan $d = 6.956$.

Percobaan dari Tabel 4.24 hingga Tabel 4.27 di atas merepresentasikan hasil dari percobaan optimasi parameter SVM dengan metode *firefly* dengan beberapa macam kernel dan sudut-sudut orientasi pada GLCM. Pada tabel 4.24 dan 4.26 menunjukkan nilai akurasi yang sama dari seluruh percobaan, tabel 4.25 menujukkan nilai akurasi yang semakin meningkat dengan semakin banyaknya iterasi yang digunakan, dan tabel 4.27 menunjukkan bahwa nilai akurasi belum tentu meningkat dengan banyaknya iterasi yang digunakan. Dari hasil keempat tabel tersebut, dapat disimpulkan bahwa nilai akurasi dari sistem dengan pengoptimasian parameter tidak bergantung pada banyaknya iterasi. Nilai akurasi yang diperoleh oleh sistem tergantung pada data yang digunakan, nilai acak *firefly* yang mempengaruhi nilai intensitas cahaya yang maksimal. Oleh karena itu, semua nilai mendekati batas atas yang telah ditentukan. Selain itu, kombinasi dari semua nilai input parameter juga mempengaruhi model dari nilai parameter yang didapatkan.

Ditinjau dari dari kernel, dua kernel yang memberikan hasil akurasi baik

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

adalah kernel RBF dengan akurasi sebesar 95.83% dan kernel *Polynomial* dengan akurasi yang sama, sebesar 95.83%. Sedangkan jika ditinjau dari sudut GLCM, sudut yang memberikan hasil terbaik adalah sudut GLCM sebesar 90°. Tahap selanjutnya akan dilakukan validasi model dari kedua hasil terbaik berdasarkan percobaan kernel dan sudut orientasi pada GLCM.

4.3. Evaluasi dan Validasi Model

4.3.1. Confusion Matrix

Subbab sebelumnya telah dipaparkan hasil dari percobaan optimasi untuk parameter SVM. Dari hasil-hasil terbaik yang diperoleh, akan dievaluasi dan validasi modelnya berdasarkan *confusion matrix*. Hal ini dilakukan untuk mendapatkan model terbaik berdasarkan analisa hasil *confusion matrix*. Analisa *confusion matrix* yaitu analisa akurasi, spesifisitas, dan sensitifitas menggunakan Persamaan 2.41, Persamaan 2.42, dan Persamaan 2.43. Hasil dari *confusion matrix* berdasarkan kernel terbaik disajikan pada Tabel 4.28 sampai Tabel4.29.

Tabel 4.28 Hasil *Confusion Matrix* dari Hasil Terbaik Kernel RBF

Percobaan			**Kelas Prediksi**	
RBF,GLCM 90°			**Negatif**	**Positif**
	Kelas Sebenarnya	**Negatif**	12	0
		Positif	1	11

Tabel 4.29 Hasil *Confusion Matrix* dari Hasil Terbaik Kernel *Polynomial*

Percobaan			**Kelas Prediksi**	
Polynomial,GLCM 90°			**Negatif**	**Positif**
	Kelas Sebenarnya	**Negatif**	1	11
		Positif	0	12

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Hasil *confusion matrix* di atas digunakan untuk menganalisa nilai akurasi, spesifisitas, dan sensitivitas dengan contoh perhitungan pada percobaan RF,GLCM 90° sebagai berikut:

$$\begin{aligned} Akurasi &= \frac{TP+TN}{TP+FN+FP+TN} \\ Akurasi &= \frac{11+12}{11+1+0+12} \\ Akurasi &= 95.83\% \end{aligned}$$

Nilai akurasi yang diperoleh merepresentasikan banyaknya suatu data yang terklasifikasi benar dengan data aktual oleh sistem yang telah dibuat. Selanjutnya dilakukan perhitungan nilai spesifisitas sebagai berikut:

$$\begin{aligned} Spesifisitas &= \frac{TN}{FP+TN} \\ Spesifisitas &= \frac{12}{0+12} \\ Spesifisitas &= 100.00\% \end{aligned}$$

Nilai spesifisitas yang diperoleh merepresentasikan banyaknya data yang benar terklasifikasi di kelas negatif. Oleh karena itu secara tidak langsung spesifisitas merupakan nilai yang mengukur kinerja sistem dalam mengklasifikasikan kelas negatif. Selanjutnya dilakukan perhitungan nilai sensitivitas sebagai berikut:

$$\begin{aligned} Sensitivitas &= \frac{TP}{TP+FN} \\ Sensitivitas &= \frac{11}{11+1} \\ Sensitivitas &= 91.67\% \end{aligned}$$

Nilai sensitivitas yang diperoleh merepresentasikan banyaknya data yang benar

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

terklasifikasi di kelas positif. Oleh karena itu secara tidak langsung sensitivitas merupakan nilai yang mengukur kinerja sistem dalam mengklasifikasi kelas positif. Dari contoh perhitungan diatas, berikut disajikan pada Tabel 4.30 hasil dari nilai akurasi, spesifisitas, dan sensitivitas dari semua hasil *confusion matrix*.

Tabel 4.30 Hasil Analisa *Confusion Matrix*

Percobaan	Akurasi	Spesifisitas	Sensitivitas
RBF,GLCM 90°	95.83%	100.00%	91.67%
Polynomial,GLCM 90°	95.83%	91.67%	100.00%

Secara *confusion matrix*, pada kernel *polynomial* ataupun RBF terdapat 1 data yang tidak terklasifikasi dengan tepat. Akan tetapi, pada *polynomial* data yang tidak terklasifikasi dengan tepat adalah data dari kelas negatif yang terklasifikasi positif. Sedangkan pada RBF, data yang tidak terklasifikasi dengan tepat adalah data dari kelas positif yang terklasifikasi negatif. Jika ditinjau berdasarkan keadaan pasien, maka sistem jauh lebih baik jika mampu memberikan kemampuan mendiagnosa penyakit secara tepat dengan akurasi sempurna. Oleh karena itu, pasien yang terdiagnosa secara tepat sedang sakit akan segera tertangani dengan baik.

Dari Tabel 4.30 dapat dipilih model sistem yang terbaik untuk klasifikasi MRI tumor otak menggunakan metde FFA-SVM dengan melihat nilai akurasi yang tinggi dan nilai sensitivitas yang tinggi. Karena jika nilai akurasi tinggi, maka sistem dapat mengklasifikasikan data aktual dengan benar. Selain itu jika nilai sensitivitas tinggi, maka sistem dapat memgklasifikasikan kelas positif dengan baik. Oleh karena itu kesalahan dalam mendiagnosa pasien tumor yang didiagnosa normal akan semakin berkurang. Dari hasil akurasi dan sensitivitas pada Tabel 4.30, hasil evaluasi terbaik yaitu pada akurasi sebesar 95.83% dan sensitivitas

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sebesar 100.00%. Oleh karena itu, dapat disimpulkan bahwa hasil terbaik yaitu pada sistem yang menggunakan kernel *polynomial* dengan data dari ekstraksi fitur GLCM pada arah orientasi $\theta = 90^o$ dan jarak piksel $d_p = 1$. Parameter yang digunakan adalah parameter hasil optimasi dengan FFA yang menghasilkan pasangan nilai $C = 1796.125$, $\gamma = 19.765$, $r = 0.0005$ dan $d = 6.956$

4.3.2. Evaluasi Waktu

Hasil evaluasi dan validasi model berdasarkan *confusion matrix* yang telah dilakukan dapat diketahui sistem yang terbaik untuk klasifikasi MRI tumor otak berdasarkan akurasi, spesifisitas, dan sensitivitas. Selanjutnya dilakukan evaluasi waktu untuk mengetahui kecepatan waktu komputasi dari kinerja klasifikasi SVM yang parameternya didapatkan dengan cara optimasi menggunakan metode FFA. Pada penelitian ini, dilakukan evaluasi waktu dengan cara membandingkan waktu komputasi pada kedua hasil terbaik FFA-SVM.

Tabel 4.31 Hasil Evaluasi Waktu Komputasi

Percobaan	*C*	γ	*r*	*d*	Waktu
RBF,GLCM 90°	1874.654	19.191	-	-	0.016
Polynomial,GLCM 90°	1796.125	19.765	0.0005	6.956	4.784

Berdasarkan hasil dari Tabel 4.31, waktu tercepat pada FFA-SVM sebesar 0.016 detik pada kernel RBF. Sedangkan pada kernel *polynomial*, waktu komputasi sebesar 4.784. Jika dibandingkan dengan FFA-SVM kernel RBF, waktu komputasi hanya berselisih sebesar 4.768 detik. Oleh karena itu dapat disimpulkan bahwa dalam penelitian ini didapatkan hasil terbaik dari FFA-SVM dengan akurasi sebesar 95.83%, spesifisitas sebesar 91.67%, dan sensitivitas sebesar 100.00% dengan menggunakan data GLCM 90° pada kernel *polynomial*. Hal ini terjadi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

karena terdapat satu data normal yang terklasifikasi tumor. Hasil ini didapatkan dengan pengoptimalan nilai parameter SVM, sehingga didapatkan nilai $C = 1796.125$, $\gamma = 19.765$, $r = 0.0005$ dan $d = 6.956$. Sedangkan waktu komputasi tercepat didapatkan pada FFA-SVM kernel RBF sebesar 0.016 detik, lebih cepat 4.768 detik dari FFA-SVM kernel *polynomial*. Jika dibandingkan dengan SVM parameter *default* hasil akurasi yang diperoleh memiliki perbedaan atau selisih yang cukup jauh. Oleh karena itu, hal ini membuktikan bahwa metode FFA dapat mengoptimasi parameter input pada SVM.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

BAB V

PENUTUP

5.1. Simpulan

Berdasarkan hasil dari implementasi dan percobaan menggunakan FFA-SVM untuk mengklasifikasikan citra MRI otak menjadi kelas normal dan tumor, maka diperoleh simpulan dan saran sebagai berikut:

1. Berdasarkan uji coba yang dilakukan untuk mengklasifikasi citra MRI otak, sistem dengan *pre-processing* menggunakan *histogram equalization* dan *median filter* dapat digunakan sebagai perbaikan citra MRI otak. Proses klasifikasi SVM dengan kombinasi ekstraksi fitur GLCM empat sudut orientasi dan empat kernel SVM, telah menunjukkan bahwa parameter *default* $C = 1$, $\gamma = 1.8$, $r = 0.0$ dan $d = 3$ hanya dapat memberikan akurasi sebesar 79.17%, sepesifisitas sebesar 58.33%, dan sensitivitas sebesar 100.00%.

2. Sistem klasifikasi menggunakan FFA-SVM, dengan kombinasi ekstraksi fitur GLCM empat sudut orientasi dan empat kernel SVM berdasarkan pada percobaan banyaknya iterasi pada FFA telah menunjukkan bahwa sudut orientasi terbaik adalah 90° dengan jarak 1 piksel. Sedangkan penggunaan kernel terbaik yaitu kernel *polynomial* dan kernel RBF. Berdasarkan hasil *confusion matrix*, kernel *polynomial* merupakan kernel terbaik dengan hasil akurasi sebesar 95.83%, sepesifisitas sebesar 91.67%, dan sensitivitas sebesar 100.00%. Nilai kombinasi parameter yang didapatkan pada kernel *polynomial* adalah $C = 1796.125$, $\gamma = 19.765$, $r = 0.0005$ dan $d = 6.956$ pada iterasi

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

sebanyak 100 iterasi. Jika berdasarkan waktu komputasi, kernel RBF merupakan kernel terbaik dengan waktu komputasi sebesar 0.016 detik, lebih cepat 4.768 detik dari FFA-SVM kernel *polynomial*. Nilai kombinasi parameter yang didapatkan pada kernel RBF adalah $C = 1874.654$ dan $\gamma = 19.191$ pada iterasi sebanyak 100 iterasi.

5.2. Saran

Penelitian mengenai klasifikasi tumor otak menggunakan FFA-SVM masih memiliki banyak kekurangan. Oleh karena itu peneliti menyarakan beberapa hal untuk perbaikan dan pengembangan penelitian selanjutnya, yaitu dilakukan proses perbaikan citra dengan metode lain, seperti *skull removal* untuk menghilangkan batas putih pada citra otak. Oleh karena itu, area yang didapatkan hanya area yang diamati. Hal ini memungkinkan untuk mendapatkan hasil sistem yang jauh lebih baik.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

DAFTAR PUSTAKA

Adewumi, O. A., and Akinyelu, A. A. (2016, Juli 20). A Hybrid Firefly and Support Vector Machine Classifier for Ehishing Email Detection. *Kybernetes*, 45(6), 977-994.

Adinegoro, A., Atmaja, R. D., dan Purnamasari, R. (2015, Agustus). Deteksi Tumor Otak dengan Ektrasi Ciri dan Feature Selection mengunakan Linear. *e-Proceeding of Engineering*, 02(2), 2532.

Adyanti, D. A.(2018). *Analisis Citra Dental Panoramic Radiograph (DPR) Pada Tulang Mandibula Untuk Deteksi Osteoporosis Menggunakan Metode Glcm - Svm Multiclass (Gray Level Co-Occurrence Matrix - Support Vector Machine Multiclass)*. Undergraduate Thesis. UIN Sunan Ampel Surabaya.

Ahmad, U. (2005). *Pengelolahan Citra Digital dan Teknik Pemrogramannya*. Yogyakarta: Graha Ilmu.

Ali, Nadhirah., Othman, M. Azlishah., Husain, Mohd Nor., and Misran, M. Harrisa. (2014, October). A Review Of Firefly Algorithm. *ARPN Journal of Engineering and Applied Sciences*, 1(10), 1732-1736.

Al-Shammari, E. T., Keivani, A., Shamshirband, S., Mostafaeipour, A., Yee, P. L., Petkovi, D., et al. (2015). Prediction of Heat Load in District Heating Systems by Support Vector Systems by Support Vector. *Elsevier*, 266-273.

Ardiansyah, D. (2018, Juli). Pengenalan Dini Faktor Resiko dan Gejala Dini Tumor Otak. *MIMBAR Rumah Sakit. dr. Soetomo Surabaya*, pp. 2-3.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Arif, R. B., Khan, M. M., and Siddique, M. A. (2018, November 23). Digital Image Enhancement in Matlab: An Overview on Histogram Equalization and Specification. *International Conference on Innovation in Engineering and Technology (ICIET)*

Arora, S., and Singh, S. (2013, May). The Firefly Optimization Algorithm: Convergence Analysis and Parameter Selection. *International Journal of Computer Applications*, 69(3), 48-52.

Bengio, Y., dan Grandvalet, Y. (2004). No unbiased estimator of the variance of k-fold cross-validation. *Journal of machine learning research*, 5(Sep), 1089-1105.

Benson, Deepa, Lajish, and Rajamani, K. (2016, September). Brain Tumor Segmentation from MR Brain Images using Improved Fuzzy c-Means Clustering and Watershed Algorithm. *IEEE*, 187-192.

Cancer Support Community. (2013). *Frankly Speaking About Cancer: Brain Tumors*. Cancer Support Community Press.

Crum, C. P., Lester, S. C., dan Cotran, R. S. (2007). *Sistem Genitalia Perempuan dan Payudara* (7 ed., Vol. 2). Jakarta: EGCss Pre.

Salamah, E. N. (2018). *Identifikasi Tumor pada Otak Menggunakan Discrete Wavelet Transform (DWT) dan Artificial Neural Network (ANN)*. Undergraduate Thesis. UIN Sunan Ampel Surabaya.

Estlin, E., and Lowis, S. L. (2005). *Central Nervous System Tumours of Childhood*. London: Mac Keith Press.

Foeady, A. Z., Novitasari, D. C., and Asyhar, A. H. (2018, October). Automated

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Diagnosis System of Diabetic Retinopathy Using GLCM Method and SVM Classifier. *Proceeding of EECSI*, 154-160.

Fletcher, T. (2008). *Support Vector Machines Explained*. University College London Press.

Gadkari, D. (2000). *Image Quality Analysis Using GLCM*. B.S.E.E. University of Pune.

Hall-Beyer, M. (2017). *GLCM Texture: A Tutorial*. Canada: University of Calgary.

Harafani, H., dan Wahono, R. S. (2015, Desember). Optimasi Parameter pada SVM Berbasis Algoritma Genetika untuk Estimasi Kebakaran Hutan. *Journal of Intelligent System*, 01(2), 82-90.

Heranurweni, S., Destyningtias, B., dan Nugroho, A. K. (2018). Klasifikasi Pola Image pada Pasien Tumor Otak Berbasis Jaringan Syaraf Tiruan (Studi Kasus Penanganan Kuratif Pasien Tumor Otak). *eLEKTRIKAL*, 10(2), 37-40.

Jakkula, V. (2006). *Tutorial on support vector machine (SVM)*. School of EECS, Washington State University.

Johari, N. F., Zain, A. M., Mustaffa, N. H., and Udin, A. (2013). Firefly Algorithm for Optimization Problem. *Applied Mechanics and Materials*, 421, 512-517.

Karuna, M., and Joshi, A. (2013, Oktober). Automatic Detection and Severity of Brain Tumors Using GUI in Matlab. *International Journal of Research in Engineering and Technology*, 02(10), 586-594.

Kesehatan, P. D. (2015). *Situasi Penyakit Kanker*. Jakarta: Kementrian Kesehatan RI.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Kumari, N., and Saxena, S. (2018). Review of Brain Tumor Segmentation and Classification. *IEEE*, 1-6.

Lang, R., Zhao, L., and Jia, K. (2016). Brain Tumor Image Segmentation Based On Convolution Neural Network. *IEEE*, 1402-1406.

Larroza, A., Bodí, V., and Moratal, D. (2016). Texture Analysis in Magnetic Resonance Imaging: Review and Considerations for Future Application. *INTECH*, 85-88.

Mathew, R., and Anto, D. B. (2017, Juli). Tumor Detection and Classification of MRI Brain Image Using Wavelet Transform and SVM. *IEEE*, 75-78.

Maulana, I., Andono, P. N., Studi, P., Informatika, T., Komputer, F. I., dan Nuswantoro, U. D. (2016). Analisa Perbandingan Adaptif Median Filter Dan Median Filter Dalam Reduksi Noise Salt and Pepper. 2(2), 157–166.

Mohanaiah, P., Sathyanarayana, P., and GuruKumar, L. (2013, May). Image Texture Feature Extraction Using GLCM Approach. *International Journal of Scientific and Research Publications*, 3(5), 1-5.

Murthy, D., and G.Sadashivappa. (2014). Brain Tumor Segmentation Using Thresholding,Morphological Operations And Extraction Of,Features Of Tumor. *IEEE*, 1-6.

Nanda, M. A., Seminar, K. B., Nandika, D., and Maddu, A. (2018). A Comparison Study of Kernel Functions in the Support Vector Machine and Its Application for Termite Detection. *Information*, 9(5), 1-14.

McLean, Radha. *The Essential Guide to Brain Tumors*. San Francisco, California: Modern Litho-Print Company.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Novitasari, D. C., Lubab, A., Sawiji, A., and Asyhar, A. H. (2019). Application of Feature Extraction for Breast Cancer using One Order Statistic, GLCM, GLRLM, and GLDM. *Advances in Science, Technology and Engineering Systems Journal*, 4(4), 115-120.

Navin, M., and Pankaja. (2016, December). Performance Analysis of Text Classification Algorithms Using Confusion Matrix. *International Journal of Engineering and Technical Research (IJETR)*, 6(4), 75-78.

Nugroho, A. S., Witarto, A. B., and Handoko, D. (2003). Support Vector Machine (Teori dan Aplikasinya dalam Bioinformatika). *Proceeding of Indonesian Scientific Meeting in Central Japan*. Gifu-Japan: IlmuKomputer.Com.

Oemiati, R., Rahajeng, E., dan Kristanto, A. Y. (2011). Prevalensi Tumor dan Beberapa Faktor yang Mempengaruhinya di Indonesia. *Buletin Peneliti Kesehatan*, 39(4), 190-204.

Parveen, and singh, A. (2015). Detection of Brain Tumor in MRI Images, using Combination of Fuzzy C-Means and SVM. *International Conference on Signal Processing and Integrated Networks (SPIN)*, 98-102.

Patel, O., Maravi, Y. P., and Sharma, S. (2013, October). A Comparative Study Of Histogram Equalization Based Image Enhancement Techniques For Brightness Preservation And Contrast Enhancement. *Signal and Image Processing : An International Journal (SIPIJ)*, 4(5), 11-25.

Perdana, A. (2017, April). Analisis Komparasi Genetic Algorithm dan Firefly Algorithm pada Permasalahan Bin Packing Problem. *Jurnal dan Penelitian Teknik Informatika*, 1(2), 1-6.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

Pragathi, J., Patil, H. T. (2013). Segmentation Method for ROI Detection in Mammographic Images using Wiener Filter and Kittler's Method. *International Journal of Computer Applications*, 27–33.

Saleh, E. (2016). *Neoplasma*. Yogyakarta: Universitas Muhammadiyah Yogyakarta.

Santi, C. N. (2011, Januari). Mengubah Citra Berwarna Menjadi GrayScale dan Citra biner. *Jurnal Teknologi Informasi Dinamik*, 16(1), 14-19.

Santra, A. K., and Christy, C. J. (2012, January). Genetic Algorithm and Confusion Matrix for Document Clustering. *IJCSI International Journal of Computer Science*, 9(1), 322-328.

Satrio, E. P. (2016). *Klasifikasi Tenun Menggunakan Metode K-Nearest Neighbour Berdasarkan Gray Level Co-Occurrence Matrices (GLCM)*. Semarang: Universitas Dian Nuswantoro.

Sebastian, B., Unnikrishnan, A., and Balakrishnan, K. (2012, April). Grey Level Co-Occurrence Matrices:Generalisation and Some New Features. *International Journal of Computer Science, Engineering and Information Technology (IJCSEIT)*, 2(2), 151-157.

Sehgal, A., Goel, S., Mangipudi, P., Mehra, A., and Tyagi, D. (2016, Juni). Automatic Brain Tumor Segmentation And Extraction In MR Images. *IEEE*, 104-107.

Sharma, A., Zaidi, A., Singh, R., Jain, S., and Sahoo, A. (2013). Optimization of SVM Classifier Using Firefly Algorithm. *IEEE*, 198-202.

Singh, R. P., and Dixit, M. (2015). Histogram Equalization: A Strong Technique

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id

for Image Enhancement. *International Journal of Signal Processing, Image Processing and Pattern Recognition*, 8(8), 345-352.

Sinha, T. (2018, May 1). Tumors : Benign and Malignant. *Cancer Therapy and Oncology International Journal*, 10(3), 1-3.

Suhag, S., and Saini, L. M. (2015, Juli). Automatic Detection of Brain Tumor by Image Processing in Matlab. *International Journal of Advances in Science Engineering and Technology*, 03(3), 114-117.

Suyanto. 2005. *Algoritma Genetika dalam MATLAB*. Penerbit ANDI. Yogyakarta.

Tim Dosen. (2016). *Pengolahan Citra*. Jakarta: Universitas Mercu Buana.

Virupakshappa, and Amarapur, D. B. (2017). An Automated Approach for Brain Tumor Identification using ANN Classifier. *IEEE*.

Wahyudi, E., Triyanto, D., dan Ruslianto, I. (2017). Identifikasi Teks Dokumen Menggunakan Metode Profile Projection dan Template Matching. *Jurnal Coding, Sistem Komputer Untan*, 03(2), 1-10.

Wong, T. T. (2015). Performance evaluation of classification algorithms by k-fold and leave-one-out cross validation. Pattern Recognition, 48(9), 2839-2846.

Yang, X. S. (2009). Firefly Algorithms for Multimodal Optimization. *Lecture in Computer Science*, 5792, 169-178.

Yang, Xin-She and He, Xingshi. (2013). Firefly Algorithm: Recent Advances and Applications. *Int. J. Swarm Intelligence*, 1(01), 36-50.

Zulpe, N., dan Pawar, V. (2012). Glcm Textural Features For Brain Tumor Classification. *International Journal Of Computer Science*, 9(3), 354–359.

digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id digilib.uinsby.ac.id