

**DETEKSI STROKE BERBASIS CITRA *COMPUTED TOMOGRAPHY*
SCAN MENGGUNAKAN *GRAY LEVEL CO-OCCURRENCE*
MATRIX DAN *MULTI LAYER PERCEPTRON***

SKRIPSI

Oleh :
QONITA FARIQOINI
NIM. 19650127



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

**DETEKSI STROKE BERBASIS CITRA *COMPUTED TOMOGRAPHY*
SCAN MENGGUNAKAN *GRAY LEVEL CO-OCCURRENCE*
MATRIX DAN *MULTI LAYER PERCEPTRON***

SKRIPSI

Diajukan kepada:
Universitas Islam Negeri Maulana Malik Ibrahim Malang
Untuk memenuhi Salah Satu Persyaratan dalam
Memperoleh Gelar Sarjana Komputer (S.Kom)

Oleh :
QONITA FARIQOINI
NIM. 19650127

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2024**

HALAMAN PERSETUJUAN


**DETEKSI STROKE BERBASIS CITRA *COMPUTED TOMOGRAPHY*
SCAN MENGGUNAKAN *GRAY LEVEL CO-OCCURRENCE*
MATRIX DAN *MULTI LAYER PERCEPTRON***

SKRIPSI


Oleh :
QONITA FARIQOINI
NIM. 19650127

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 13 Desember 2024

Pembimbing I,



Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Pembimbing II,


Dr. Cahyo Crysdian, M.Cs
NIP. 19740424 200901 1 008

Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang




Dr. Fachrul Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

HALAMAN PENGESAHAN

DETEKSI STROKE BERBASIS CITRA *COMPUTED TOMOGRAPHY* *SCAN* MENGGUNAKAN *GRAY LEVEL CO-OCCURRENCE* *MATRIX* DAN *MULTI LAYER PERCEPTRON*

SKRIPSI

Oleh:
QONITA FARIQOINI
NIM. 19650127

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 17 Desember 2024

Susunan Dewan Penguji

Ketua Penguji : Dr. M. Amin Hariyadi, M.T
NIP. 19670018 200501 1 001

Anggota Penguji I : Ajib Hanani, M.T
NIP. 19840731 202321 1 013

Anggota Penguji II : Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Anggota Penguji III : Dr. Cahyo Crysdiyan, M.Cs
NIP. 19740424 200901 1 008

(
(
(
(



Mengetahui,
Ketua Program Studi Teknik Informatika
Fakultas Sains dan Teknologi
Universitas Islam Negeri Maulana Malik Ibrahim Malang



Dr. Iqbal Kurniawan, M.MT, IPU
NIP. 19771020 200912 1 001

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Qonita Fariqoini
NIM : 19650127
Fakultas : Sains dan Teknologi
Program Studi : Teknik Informatika
Judul Skripsi : Deteksi Stroke Berbasis Citra *Computed Tomography Scan* Menggunakan *Gray Level Co-occurrence Matrix* dan *Multi Layer Perceptron*

Menyatakan dengan sebenarnya bahwa Skripsi yang saya tulis ini benar-benar hasil karya saya sendiri, bukan merupakan pengambilalihan data, tulisan atau pikiran orang lain yang saya akui hasil tulisan atau pikiran saya sendiri, kecuali dengan mencantumkan sumber cuplikan pada daftar pustaka.

Apabila di kemudian hari terbukti atau dapat dibuktikan Skripsi ini hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 20 Desember 2024
Yang Membuat Pernyataan,



Qonita Fariqoini
NIM.19650127

MOTTO

“Barangsiapa yang menempuh jalan untuk mencari ilmu, maka Allah akan memudahkan baginya jalan menuju surga” – (HR. Muslim)

HALAMAN PERSEMBAHAN

Puji Syukur atas kehadiran Allah *Subhanahu wa ta'ala*
Shalawat serta salam kepada Rasulullah *Shallallahu 'alaihi wasallam*

Penulis mempersembahkan karya ini kepada orang tua, dosen, adik, sahabat, dan semua pihak yang telah memberikan motivasi, doa, dan dukungan sehingga skripsi ini dapat selesai dengan baik.

KATA PENGANTAR

Dengan mengucapkan syukur kehadiran Allah SWT, atas rahmat, karunia, dan hidayah-Nya, penulis dapat menyelesaikan skripsi ini yang berjudul “Deteksi Stroke Berbasis Citra *Computed Tomography Scan* Menggunakan *Gray Level Co-occurrence Matrix* dan *Multi Layer Perceptron*”.

Penulis mengucapkan terima kasih sebesar-besarnya kepada semua pihak yang membantu menyusun skripsi dalam bentuk apa pun. Oleh karena itu penulis mengucapkan terima kasih kepada:

1. Prof. Dr. H.M. Zainuddin, MA selaku rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang
2. Prof. Dr. Hj. Sri Harini, M.Si selaku dekan Fakultas Sains dan Teknologi UIN Maulana Malik Ibrahim Malang
3. Dr. Ir. Fachrul Kurniawan, M.MT, IPU selaku Ketua Program Studi Teknik Informatika UIN Maulana Malik Ibrahim Malang
4. Dr. Irwan Budi Santoso, M.Kom dan Dr. Cahyo Crysdiyan, M.Cs sebagai Pembimbing I dan Pembimbing II yang telah memberikan ilmu dalam menyusun skripsi
5. Dr. M. Amin Hariyadi, M.T dan Ajib Hanani, M.T sebagai Penguji I dan Penguji II yang telah memberikan kritik serta saran baik dalam menyusun skripsi
6. Seluruh Dosen dan Sivitas Akademika Teknik Informatika UIN Malang yang telah memberikan arahan dan ilmunya

7. Orang tua dan adik-adik penulis yang telah memberikan dukungan dan semangat dalam menyelesaikan skripsi
8. Teman-teman Angkatan ALIEN 19 dan khususnya teman-teman Kelas E yang telah memberikan dukungan secara langsung maupun tidak langsung
9. Teman-teman yang tidak mampu saya sebut satu persatu yang telah memberikan dukungan baik secara langsung maupun tidak langsung.
10. Penulis sendiri yang telah berjuang dalam penulisan skripsi ini dengan seluruh kemampuannya.

Skripsi yang telah ditulis ini tidak luput dari kesalahan dan masih jauh dari kata sempurna. Penulis akan senang jika ada kritik dan saran baik dari pihak mana pun. Semoga skripsi ini memberikan manfaat untuk penulis maupun pembaca.

Wassalamualaikum Warahmatullahi Wabarakatuh

Malang, 20 Desember 2024

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR	viii
DAFTAR ISI	x
DAFTAR TABEL	xi
DAFTAR GAMBAR	xii
ABSTRAK	xiii
ABSTRACT	xiv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan masalah.....	6
1.3 Batasan Masalah	6
1.4 Tujuan Penelitian	6
1.5 Manfaat Penelitian	6
1.6 Sistematika Penulisan	7
BAB II STUDI PUSTAKA	9
2.1 Deteksi Stroke.....	9
2.2 Stroke.....	14
2.3 <i>Gray Level Co-occurrence Matrix (GLCM)</i>	17
2.4 <i>Multi Layer Perceptron</i>	19
BAB III DESAIN DAN IMPLEMENTASI	21
3.1 Pengumpulan Data.....	21
3.2 Desain Sistem	22
3.2.1 Preprocessing.....	23
3.2.2 Ekstraksi Fitur Gray Level Co-occurrence Matrix (GLCM)	25
3.2.3 Multi Layer Perceptron (MLP)	34
BAB IV UJI COBA DAN PEMBAHASAN	43
4.1 Skenario Uji Coba.....	43
4.2 Hasil Uji Coba Jarak	49
4.3 Hasil Uji Coba Jumlah Hidden Layer	51
4.4 Hasil Uji Coba Jumlah Neuron	54
4.5 Pembahasan	58
BAB V PENUTUP	66
5.1 Kesimpulan	66
5.2 Saran	66
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR TABEL

Tabel 4.1 Rasio pembagian data penelitian	48
Tabel 4.2 Hasil confusion matrix dengan jarak 1	49
Tabel 4.3 Nilai precision, recall, dan f1-score dengan jarak 1	49
Tabel 4.4 Hasil confusion matrix dengan jarak 2	49
Tabel 4.5 Nilai precision, recall, dan f1-score dengan jarak 2	50
Tabel 4.6 Hasil confusion matrix dengan kombinasi jarak 1 dan jarak 2	50
Tabel 4.7 Nilai precision, recall, dan f1-score dengan jarak 1 dan 2	50
Tabel 4.8 Hasil confusion matrix uji coba jarak	51
Tabel 4.9 Hasil confusion matrix dengan satu hidden layer	52
Tabel 4.10 Nilai precision, recall, dan f1-score dengan satu hidden layer	52
Tabel 4.11 Hasil confusion matrix dengan dua hidden layer	53
Tabel 4.12 Nilai precision, recall, dan f1-score dengan dua hidden layer	53
Tabel 4.13 Hasil confusion matrix dengan tiga hidden layer	53
Tabel 4.14 Nilai precision, recall, dan f1-score dengan tiga hidden layer	53
Tabel 4.15 Hasil confusion matrix uji coba jumlah hidden layer	54
Tabel 4.16 Hasil confusion matrix dengan 32 neuron	55
Tabel 4.17 Nilai precision, recall, dan f1-score dengan 32 neuron	55
Tabel 4.18 Hasil confusion matrix dengan 64 neuron	56
Tabel 4.19 Nilai precision, recall, dan f1-score dengan 64 neuron	56
Tabel 4.20 Hasil confusion matrix dengan 128 neuron	56
Tabel 4.21 Nilai precision, recall, dan f1-score dengan 128 neuron	56
Tabel 4.22 Hasil confusion matrix dengan 256 neuron	57
Tabel 4.23 Nilai precision, recall, dan f1-score dengan 256 neuron	57
Tabel 4. 24 Hasil confusion matrix uji coba jumlah neuron	57

DAFTAR GAMBAR

Gambar 2.1	Arsitektur MLP.....	20
Gambar 3.1	Contoh citra otak normal dan stroke	22
Gambar 3.2	Desain Sistem	22
Gambar 3.3	Alur ekstraksi fitur GLCM	26
Gambar 3.4	Alur pembentukan matriks GLCM	27
Gambar 3.5	Alur perhitungan normalisasi matriks.....	28
Gambar 3.6	Alur perhitungan ekstraksi fitur GLCM.....	29
Gambar 3.7	Arsitektur MLP.....	35
Gambar 4.1	Alur proses training MLP algoritma backpropagation	44
Gambar 4.2	Alur proses testing MLP algoritma backpropagation.....	45
Gambar 4.3	Hasil uji coba dengan variasi jarak	58
Gambar 4.4	Hasil uji coba dengan variasi jumlah hidden layer.....	60
Gambar 4.5	Hasil uji coba dengan variasi jumlah neuron	61

ABSTRAK

Fariqoini, Qonita. 2024. **Deteksi Stroke Berbasis Citra *Computed Tomography Scan* Menggunakan *Gray Level Co-Occurrence Matrix* dan *Multi Layer Perceptron*.**
Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso, M.Kom (II) Dr. Cahyo Crysodian, M.Cs.

Kata Kunci: Citra *CT scan* , GLCM, MLP, Stroke.

Stroke adalah penyebab kematian kedua terbanyak di dunia dan sering menyebabkan kecacatan pada orang dewasa. Penyakit ini tiap tahunnya akan mengalami peningkatan karena gejala yang ditimbulkan oleh penyakit stroke sifatnya mendadak dan sering terlambat disadari. Deteksi dini terhadap stroke melalui penggunaan teknologi citra medis, khususnya *CT scan*, berguna dalam diagnosis awal untuk kelangsungan hidup pasien. Penelitian ini bertujuan untuk menganalisis performa *Gray Level Co-occurrence Matrix* (GLCM) dan model *Multi Layer Perceptron* (MLP) dalam mendeteksi stroke berdasarkan citra *CT scan* otak. Metode GLCM digunakan untuk ekstraksi fitur tekstur dengan menguji pengaruh variasi parameter jarak dan sudut terhadap kualitas fitur yang dihasilkan. Selanjutnya, model MLP dengan algoritma *backpropagation* digunakan untuk deteksi, dengan mengoptimalkan jumlah *hidden layer* dan jumlah neuron untuk peningkatan performa. Hasil penelitian menunjukkan bahwa variasi parameter pada GLCM dan konfigurasi arsitektur MLP berpengaruh signifikan terhadap performa deteksi stroke, menghasilkan akurasi sebesar 91,13%, presisi sebesar 89,83%, *recall* sebesar 91,15%, dan *f1-score* sebesar 90,48%. Penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan metode berbasis citra untuk diagnosis stroke secara lebih akurat.

ABSTRACT

Fariqoini, Qonita. 2024. **Computed Tomography Scan Image-Based Stroke Detection Using Gray Level Co-Occurrence Matrix and Multilayer Perceptron.** Informatics Engineering Department Faculty of Science and Technology Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisor: (I) Dr. Irwan Budi Santoso, M.Kom (II) Dr. Cahyo Crysdiان, M.Cs.

Stroke is the second leading cause of death worldwide, and it leads to disability among adult survivors. It is increasing annually since its symptoms come suddenly and unpredictably. The earlier detection of stroke can use medical image technology, particularly CT scans, to provide an early diagnosis of a patient's life. The research aims to analyze the performance of the Gray Level Co-occurrence Matrix (GLCM) and Multilayer Perceptron Model (MLP) in detecting stroke based on the brain scan CT image. The GLCM method performs texture feature extraction by testing the influence of distance and angle parameter variation on the generated feature quality. Then, the MLP model with a backpropagation algorithm is used for detection by optimizing the number of hidden layers and neurons to increase the performance. The research result shows that the parameter variation of the GLCM and MLP architectural configuration has a significant influence on the performance of stroke detection, generating an accuracy of 91.13%, precision of 89.83%, recall of 91.15%, and f1-score of 90.48%. The research is expected to contribute to the image-based method development for stroke diagnosis accurately.

Keywords: CT scan image, Gray Level Co-occurrence Matrix, Multilayer Perceptron, Stroke.

مستخلص البحث

فريقيين، قاننة . 2024 . الكشف عن السكتة الدماغية بناء على صور التصوير المقطعي المحوسب باستخدام مصفوفة الحدوث المتزامن على المستوى الرمادي والمستقبلات متعددة الطبقات. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. إيوان بودي سانتوسو، الماجستير. المشرف الثاني: د. جاهيو كريسدیان، الماجستير.

الكلمات الرئيسية: صور تصوير مقطعي محوسب، مصفوفة حدوث متزامن على مستوى رمادي، مستقبلات متعددة طبقات، سكتة دماغية.

السكتة الدماغية هي السبب الرئيسي الثاني للوفاة في العالم وغالبا ما تسبب الإعاقة لدى البالغين. سيزداد هذا المرض كل عام لأن الأعراض التي تسببها السكتة الدماغية مفاجئة وغالبا ما تكون متأخرة جدا لتحقيقها. يعد الكشف المبكر عن السكتة الدماغية من خلال استخدام تقنية التصوير الطبي -وتحديدا التصوير المقطعي المحوسب- مفيدا في التشخيص المبكر لبقاء المريض. هدف هذا البحث إلى تحليل أداء مصفوفة الحدوث المتزامن للمستوى الرمادي (GLCM) ونموذج المستقبلات متعددة الطبقات (MLP) في الكشف عن السكتة الدماغية بناء على صور التصوير المقطعي المحوسب للدماغ. تستخدم طريقة GLCM لاستخراج ميزة النسيج عن طريق اختبار تأثير التباين في معلمات المسافة والزاوية على جودة الميزات الناتجة. علاوة على ذلك، يتم استخدام نموذج MLP مع خوارزمية الانتشار الخلفي للكشف، خلال تحسين عدد الطبقات المخفية وعدد الخلايا العصبية لتحسين الأداء. أظهرت النتائج أن تباين المعلمات في تكوين بنية GLCM و MLP كان له تأثير كبير على أداء الكشف عن السكتة الدماغية، مما أدى إلى دقة 91.13٪، ودقة 89.83٪، واستدعاء 91.15٪، ودرجة ف 1 بنسبة 90.48٪.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Stroke adalah penyebab kematian kedua terbanyak di dunia dan sering menyebabkan kecacatan pada orang dewasa (Geneva, 2023). Definisi stroke menurut WHO adalah gangguan neurologis fokal atau global yang muncul tiba-tiba dan berasal dari masalah pembuluh darah, berlangsung lebih dari 24 jam atau menyebabkan kematian. Data *Global Stroke Fact Sheet 2022* dari *World Stroke Organization* (WSO) menyatakan ada lebih dari 12,2 juta kasus stroke baru setiap tahun, dan lebih dari 101 juta orang saat ini hidup dengan menderita stroke. Setiap tahun, 6,5 juta orang meninggal akibat stroke.

Dari banyaknya kasus stroke yang terjadi tidak menutup kemungkinan di tiap tahunnya akan mengalami peningkatan karena gejala yang ditimbulkan oleh penyakit stroke sifatnya mendadak dan sering terlambat disadari. Masih banyak pihak yang terlambat akan kesadaran kondisi tubuhnya dan mendapati kondisi serius ketika melakukan pemeriksaan (Amraeni, 2021). Dibutuhkan ketepatan dalam mendiagnosis penyakit tersebut sehingga dapat mengenali lebih awal dan mendapatkan penanganan yang tepat. Untuk dapat menegakkan diagnosis pada kasus stroke dapat didukung dengan teknologi sebagaimana perkembangan teknologi saat ini yang dapat memberikan pengaruh besar terhadap kehidupan termasuk di ranah kesehatan (Cholik, 2021).

Teknologi yang dapat digunakan yaitu mengidentifikasi citra melalui data citra *Computed Tomography Scanning* (CT scan) otak seorang pasien. *Computed Tomography Scanning* (CT scan) menjadi penunjang diagnosis yang mempunyai aplikasi universal dalam pemeriksaan organ tubuh seperti tulang, otot, tenggorokan, rongga perut hingga susunan syaraf pusat (Sanubary, 2018). Dalam pengenalan pola citra CT scan tersebut digunakan metode jaringan syaraf tiruan. Karakteristik dari jaringan syaraf tiruan dapat dikatakan mirip dengan jaringan syaraf biologis yang terdapat pada otak manusia. Jaringan syaraf tiruan salah satu metode yang baik untuk pengenalan pola-pola yang cukup kompleks. Salah satu algoritma jaringan syaraf tiruan yang dapat digunakan yaitu *Multi Layer Perceptron* (MLP). *Multi Layer Perceptron* (MLP) dikenal handal dengan proses pembelajaran yang dapat dilakukan secara terarah. Pada *Multi Layer Perceptron* dilakukan pengupdatean bobot balik (*backpropagation*). Bobot tersebut akan berpengaruh pada hasil prediksi yang tepat dengan menetapkan bobot secara optimal.

Penelitian yang telah dilakukan menggunakan *Multi Layer Perceptron* (MLP) dalam citra medis menunjukkan beberapa keunggulan yang signifikan. Studi Mataija & Nemeč (2021) menunjukkan bahwa MLP mampu mengatasi tantangan keanekaragaman dataset dengan jumlah 253 citra yang meningkatkan akurasi klasifikasi 78,66% menjadi 95% pada data yang terbatas. Penelitian Sharma et al., (2022) menyoroti efektivitas MLP dalam mengatasi kelemahan waktu respon yang lebih lama terhadap pengklasifikasian citra motorik, memberikan hasil akurasi yang signifikan hingga mencapai 92,5% dalam skenario

kasus terbaik, dengan tiap epoch diselesaikan dalam waktu kurang dari 7 detik dengan pembelajaran pola yang efisien. Sementara itu, penelitian Chakraborty et al., (2020) menegaskan bahwa penggunaan MLP memberikan kinerja terbaik dalam mengklasifikasikan penyakit Parkinson pada pemindaian MRI dengan akurasi keseluruhan 95,3%, presisi keseluruhan 97,28% dan skor F1 mencapai 94%. Berdasarkan hasil tersebut penelitian ini menggunakan MLP sebagai salah satu algoritma dari jaringan syaraf tiruan dengan keunggulan meningkatkan akurasi klasifikasi, mengatasi batasan data, dan memberikan hasil yang optimal dalam penanganan berbagai kondisi medis. Penggunaan *Gray Level Co-occurrence Matrix* (GLCM) juga mendukung penelitian ini untuk mengekstraksi ciri karena metode ini termasuk metode ekstraksi ciri paling baik (Nithya & Santhi, 2011). GLCM dianggap sebagai metode yang mampu melakukan klasifikasi dan identifikasi yang cukup efektif karena memberikan informasi rinci dari suatu citra terkait hal tekstur (Listia et al., 2014).

Stroke termasuk penyakit yang hampir belum dapat disembuhkan di dunia medis. Dalam kehidupan masyarakat masih banyak terjadi keterlambatan penanganan karena tidak dilakukannya deteksi dini yang dilatarbelakangi dengan kurangnya kesadaran tiap individu terhadap kesehatan dirinya. Perspektif islam mengenai kesehatan dapat dikaitkan dengan penerapan Maqosid Syariah yang membahas tentang upaya manusia untuk dapat memahami tujuan Allah menerapkan syariat atau hukum-hukum yang terdiri dari usaha untuk menjaga agama, akal, jiwa, keturunan, dan harta (Kasdi, 2014). Dari lima prinsip dasar kehidupan manusia tersebut, penyakit termasuk upaya pencegahan maupun

pengobatan, dan bidang kesehatan lainnya dikaitkan dengan usaha untuk menjaga akal, jiwa dan keturunan. Sebagaimana firman Allah dalam QS. Al-Maidah ayat 32 yang menekankan terhadap pentingnya menjaga kehidupan.

مِنْ أَجْلِ ذَلِكَ كَتَبْنَا عَلَىٰ بَنِي إِسْرَائِيلَ أَنَّهُ ۖ مَنْ قَتَلَ نَفْسًا بِغَيْرِ نَفْسٍ أَوْ فَسَادٍ فِي الْأَرْضِ فَكَأَنَّمَا قَتَلَ النَّاسَ جَمِيعًا ۚ وَمَنْ أَحْيَاهَا فَكَأَنَّمَا أَحْيَا النَّاسَ جَمِيعًا ۚ وَلَقَدْ جَاءَتْهُمْ رُسُلُنَا بِالْبَيِّنَاتِ ثُمَّ إِنَّ كَثِيرًا مِّنْهُمْ بَعْدَ ذَلِكَ فِي الْأَرْضِ لَمُسْرِفُونَ

“Oleh karena itu, Kami menetapkan (suatu hukum) bagi Bani Israil bahwa siapa yang membunuh seseorang bukan karena (orang yang dibunuh itu) telah membunuh orang lain atau karena telah berbuat kerusakan di bumi, maka seakan-akan dia telah membunuh semua manusia.211) Sebaliknya, siapa yang memelihara kehidupan seorang manusia, dia seakan-akan telah memelihara kehidupan semua manusia. Sungguh, rasul-rasul Kami benar-benar telah datang kepada mereka dengan (membawa) keterangan-keterangan yang jelas. Kemudian, sesungguhnya banyak di antara mereka setelah itu melampaui batas di bumi.” (QS. Al-Maidah : 32).

Penyakit juga datangnya dari Allah Subhanahu Wa ta’ala dan hanya kepada-Nyalah, manusia berhak meminta kesembuhan tersebut sebagaimana yang telah dijelaskan dalam QS. Yunus ayat 57

يَا أَيُّهَا النَّاسُ قَدْ جَاءَتْكُمْ مَوْعِظَةٌ مِّن رَّبِّكُمْ وَشِفَاءٌ لِّمَا فِي الصُّدُورِ وَهُدًى وَرَحْمَةٌ لِّلْمُؤْمِنِينَ

“Wahai manusia, sungguh telah datang kepadamu pelajaran (Al-Qur’an) dari Tuhanmu, penyembuh bagi sesuatu (penyakit) yang terdapat dalam dada, dan petunjuk serta rahmat bagi orang-orang mukmin.” (QS. Yunus : 57).

Berdasarkan Tafsir Al-Madinah Al-Munawwarah menambahkan bahwa Al-Qur'an adalah "cahaya yang melenyapkan kegelapan jahiliyah," yang dapat dianalogikan dengan cara ilmu pengetahuan dan teknologi membantu menghapus kegelapan ketidakpastian dan ketidaktahuan dalam diagnosis medis. Dalam konteks ini, penggunaan teknologi canggih dalam kedokteran tidak hanya membantu dalam diagnosis yang lebih akurat tetapi juga mengurangi kecemasan

pasien dan meningkatkan kemungkinan pengobatan yang berhasil, membawa "cahaya" ke dalam kegelapan yang disebabkan oleh penyakit. Selain itu, "rahmat yang besar bagi orang-orang yang beriman" dalam tafsir ini dapat diinterpretasikan sebagai manfaat besar yang diberikan oleh penelitian Anda kepada masyarakat, khususnya bagi mereka yang berada dalam pencarian solusi melalui kepercayaan pada kemajuan ilmiah untuk pengobatan penyakit serius seperti stroke.

Konsep yang diberikan oleh Leavell and Clark (1958) juga membagi usaha pencegahan sakit dalam 5 tahap yaitu promosi kesehatan, perlindungan khusus, deteksi dini dan terapi segera, pencegahan kecacatan, dan pemulihan (Azhar et al., 2020). Dalam hal ini deteksi dini dan terapi termasuk ke dalam fase patologis. Upaya deteksi dini ini sangat penting dilakukan mengingat banyaknya kejadian yang bisa terlewat bahkan saat diagnosis sudah dalam tahapan lanjutan terutama untuk penyakit tidak menular seperti stroke yang dapat berlangsung lama serta gejala dan tanda-tandanya yang tidak nyata.

Penelitian ini memanfaatkan data citra *Computed Tomography* (CT scan) berasal dari situs resmi *Kaggle* yang akan menghasilkan dua kelas berbeda dari adanya pembagian data citra yang sudah tersedia ke dalam kelas stroke dan normal. Sebelum melangkah dalam mendeteksi penyakit stroke diperlukan adanya *preprocessing* dan juga penerapan metode untuk ekstraksi ciri terhadap data citra CT scan yang sudah tersedia.

Berdasarkan penjabaran di atas, metode *Multi Layer Perceptron* diterapkan pada penelitian dalam mendeteksi ada atau tidaknya stroke pada pasien melalui

pengenalan pola terhadap citra CT scan otak ditunjang dengan ekstraksi ciri *Gray Level Co-occurrence Matrix* untuk mengenali tekstur sehingga didapat hasil yang optimal dalam mengidentifikasi atau mendeteksi penyakit stroke melalui citra CT scan otak.

1.2 Pernyataan masalah

Bagaimana pengaruh dan performa *Gray Level Co-occurrence Matrix* (GLCM) dan *Multi Layer Perceptron* (MLP) dalam mendeteksi citra CT scan otak untuk membedakan antara otak yang normal dan otak yang terkena stroke?

1.3 Batasan Masalah

Batasan-batasan masalah yang diterapkan pada penelitian ini adalah sebagai berikut:

1. Citra yang digunakan pada penelitian ini merupakan citra normal dan stroke dengan format (*.jpg).
2. Data CT scan didapatkan dari *Kaggle* yang diakses pada tanggal 27 Juli 2023.

1.4 Tujuan Penelitian

Menganalisis pengaruh *Gray Level Co-occurrence Matrix* (GLCM) dan mengetahui bagaimana performa *Gray Level Co-occurrence Matrix* (GLCM) serta *Multi Layer Perceptron* (MLP) yang diterapkan untuk mendeteksi penyakit stroke yang berbasis citra CT scan otak.

1.5 Manfaat Penelitian

Menganalisis pengaruh *Gray Level Co-occurrence Matrix* (GLCM) dan mengetahui bagaimana performa *Gray Level Co-occurrence Matrix* (GLCM) serta *Multi Layer Perceptron* (MLP) yang diterapkan untuk mendeteksi penyakit stroke yang berbasis citra CT scan otak.

1.6 Sistematika Penulisan

Penelitian ini disusun dengan sistematika penulisan sebagai berikut.

A. BAB I PENDAHULUAN

Bab ini sebagai pembuka dari penelitian yang menjabarkan secara umum terkait hal yang melatarbelakangi penelitian, tujuan dan manfaat serta batasan lingkup penelitian.

B. BAB II TINJAUAN PUSTAKA

Bab ini membahas terkait penelitian terdahulu yang relevan dengan topic penelitian ini, termasuk penjabaran secara teoritis perihal konsep yang digunakan pada penelitian ini.

C. BAB III DESAIN DAN IMPLEMENTASI

Bab ini menjelaskan rancangan penelitian, metodologi, implementasi tahapan-tahapan penelitian serta teknik pengambilan dan analisis data.

D. BAB IV UJI COBA DAN PEMBAHASAN

Bab ini berisi penjabaran dari hasil uji coba sistem, hasil penerapan tahapan-tahapan penelitian serta hasil analisis pengujian yang telah dievaluasi dan dikaji ulang dari teori yang digunakan.

E. BAB V KESIMPULAN DAN SARAN

Bab ini berisi pernyataan berupa simpulan yang didapat dari hasil uji coba dan analisis penelitian serta saran-saran yang berguna untuk pengembangan lanjutan terkait penelitian serupa.

BAB II

STUDI PUSTAKA

2.1 Deteksi Stroke

Stroke adalah salah satu jenis gangguan kesehatan dengan kategori mematikan setelah penyakit jantung dan kanker (Lingga, 2013). Rizaldi (2016) mengatakan bahwa gejala penyakit ini dipengaruhi oleh bagian otak yang terganggu, antara lain kelumpuhan otot dan gangguan refleks, hilangnya kemampuan melihat seperti penglihatan yang buram, kelelahan, serta rasa berat pada kepala dan mengalami vertigo, kesemutan atau sensasi sentuhan berkurang serta keseimbangan tubuh terganggu. Stroke termasuk salah satu penyakit yang sering tidak disadari oleh masyarakat, sehingga ketika gejalanya muncul, kondisi penderita sudah berada pada tahap yang serius. Penyakit stroke disebabkan oleh pecahnya pembuluh darah atau terjadinya penyumbatan pada pembuluh darah (Rachman, 2021). Teknologi pendeteksian objek dapat mendukung dalam pengenalan atau penanganan suatu objek (Li et al., 2020). Deteksi dapat diartikan sebagai proses pemeriksaan terhadap objek dengan menggunakan metode dan teknik tertentu (Apriani, dkk, 2020). Objek yang dimaksud adalah penyakit stroke, yang dapat dikenali melalui citra medis, yaitu CT scan. Deteksi dini stroke dapat meningkatkan akurasi hasil diagnosis serta mempercepat dan mempermudah penanganan medis yang lebih tepat.

Subudhi et al., (2019) melakukan diagnosis terhadap stroke iskemik akut menggunakan dataset MRI otak yang terdiri dari 192 slice, yang terbagi menjadi tiga jenis stroke. Penelitian ini menerapkan Expectation-Maximization (EM)

untuk segmentasi area yang terkena dampak stroke, dilanjutkan dengan Fractional-Order Darwinian Particle Swarm Optimization (FODPSO) untuk memisahkan bagian-bagian yang tidak terkena dampak. Citra otak yang telah disegmentasi kemudian diklasifikasikan menggunakan dua model, yaitu Support Vector Machine (SVM) dan Random Forest (RF). Hasil penelitian menunjukkan bahwa RF memiliki akurasi yang lebih tinggi (93,4%) dibandingkan SVM yang hanya mencapai 92%.

Penelitian yang dilakukan oleh Muliantara & Widiartha, (2011) mengimplementasikan sebuah model untuk memprediksi anotasi pada citra. Penentuan fitur citra dilakukan dengan menggunakan metode color quantization dan multi-level wavelet transform. Prediksi anotasi citra dilakukan dengan menerapkan metode Multi-Layer Perceptron (MLP). Untuk mengevaluasi kinerja model yang diimplementasikan, digunakan dataset citra sebanyak 453. Hasil penelitian menunjukkan bahwa tingkat akurasi prediksi anotasi menggunakan MLP mencapai 81%.

Pada tahun 2019 Usha & Perumal, (2019) melakukan penelitian pengklasifikasian gambar otak yang berbasis citra MRI. Citra yang digunakan berupa dataset gambar otak MRI dengan format DICOM yang didapatkan dari Pemerintah Tirunelveli, Rumah Sakit Tamil Nadu sejumlah 40 citra normal dan 40 citra abnormal dengan masing-masing 20 citra normal dan abnormal telah dilakukan pengujian. Citra tersebut diklasifikasikan menjadi dua kategori yaitu tumor dan non-tumor. Pengklasifikasian didukung dengan pendekatan alternative GLCM berbasis HTT yang terdiri dari tiga tahapan yaitu teknik transformasi

hierarki, ekstraksi fitur tekstur, dan klasifikasi dan membandingkan dengan metode ekstraksi fitur GLCM konvensional. Hasil yang didapatkan memberikan akurasi 80%. Kinerja metode penelitian yang digunakan juga dibandingkan dengan berbagai fungsi kernel.

Peng et al., (2020) melakukan penelitian untuk memprediksi stroke berdasarkan data fisiologis dasar, riwayat penyakit, dan lingkungan hidup pasien menggunakan Artificial Neural Network (ANN). Penyakit stroke yang didiagnosis meliputi stroke iskemik, stroke hemoragik, dan pendarahan subarachnoid dengan total 43.400 data training. Dataset ini dibagi menjadi tiga subset: 70% untuk training, 15% untuk pengujian, dan 15% untuk validasi. Penelitian ini menggunakan dua algoritma dalam metode ANN, yaitu Levenberg Marquardt dan Scaled Conjugate Gradient, yang berhasil meningkatkan akurasi prediksi stroke hingga 98% pada kondisi 1000-fold cross-validation, dengan deviasi error terbaik sebesar 1%. Penelitian ini juga menunjukkan bahwa jumlah hidden node memengaruhi performa model namun tidak terlalu berpengaruh terhadap akurasi pada klasifikasi.

Hussain & Khunteta, (2020) melakukan klasifikasi dan segmentasi tumor otak dengan tujuan memisahkan jaringan tumor otak dari citra MRI otak. Data citra MRI yang digunakan diambil dari situs www.kaggle.com dan www.figshare.com, dengan resolusi 256x256 piksel dengan tingkat skala abu-abu 8 bit. Sebanyak 36 gambar digunakan sebagai data pelatihan, sementara 26 gambar lainnya digunakan sebagai data uji. Segmentasi tumor otak diterapkan menggunakan metode segmentasi watershed. Sebelum tahap segmentasi,

dilakukan median filtering dan skull stripping dalam proses preprocessing, karena citra MRI sering mengandung noise, dan intensitas tengkorak yang hampir serupa dengan area tumor otak. Setelah segmentasi, dilakukan ekstraksi fitur dengan empat fitur GLCM, yaitu kontras, korelasi, energi, dan homogenitas. Selanjutnya, citra diklasifikasikan menggunakan Support Vector Machine (SVM). Akurasi rata-rata yang diperoleh mencapai 93,05%, dengan usulan penerapan enam model pembelajaran SVM, yaitu SVM Linear, SVM Quadratic, SVM Cubic, SVM Fine-Gaussian, SVM Medium Gaussian, dan SVM Coarse-Gaussian.

Hokkinen et al., (2021) melakukan penelitian untuk memprediksi volume infark akhir menggunakan CTA dan membandingkan hasilnya dengan perangkat lunak berbasis CTP. Metode yang digunakan dalam penelitian ini adalah Convolutional Neural Network (CNN). Data pasien stroke yang digunakan berasal dari Rumah Sakit Universitas Helsinki pada periode Januari 2018 hingga Juni 2019, dengan total 83 pasien yang dianalisis. Metode CNN menghasilkan korelasi yang baik dengan nilai $r = 0,89$, sensitivitas 1,00, dan spesifisitas 0,94.

Rachman, (2021) melakukan diagnosis terhadap pasien dengan gejala stroke menggunakan sistem pakar yang dikembangkan oleh penulis. Sistem pakar ini menerapkan metode Case-Based Reasoning dan algoritma Probabilistic Symmetric untuk membantu pengambilan keputusan berdasarkan gejala yang dirasakan pasien. Pengujian sistem pakar ini dilakukan melalui tampilan pertanyaan terkait gejala yang dirasakan oleh pasien, di mana gejala yang dipilih akan mengarahkan pada diagnosis penyakit stroke. Terdapat empat kategori stroke dalam sistem pakar ini, yaitu Stroke Iskemik, Stroke Hemoragik, Stroke Ringan,

dan Stroke Brain Stem. Algoritma K-Nearest Neighbor (KNN) digunakan untuk menentukan kemiripan gejala stroke, dan hasil perhitungan KNN digunakan untuk perhitungan disimilaritas pada algoritma Probabilistic Symmetric dengan hasil 0,1667. Perhitungan kemiripan menghasilkan tingkat akurasi akhir sebesar 66,66%.

Kemudian Tazin et al., (2021) melakukan deteksi dan prediksi penyakit stroke menggunakan empat metode yang berbeda, yaitu Decision Tree (DT), Logistic Regression (LR), Voting Classifier, dan Random Forest Classifier (RF). Metode RF menunjukkan tingkat akurasi yang lebih tinggi dibandingkan dengan metode lainnya, yaitu 96%. Data yang digunakan diperoleh dari Stroke Prediction Dataset yang bersifat open-access. Penelitian ini juga menunjukkan hasil akurasi yang lebih baik dibandingkan dengan penelitian sebelumnya yang menggunakan dataset yang sama.

Zuriati & Qomariyah, (2023) melakukan klasifikasi penyakit stroke dengan menggunakan metode K-Nearest Neighbor (KNN). Data yang digunakan terdiri dari 5110 record yang dibagi menjadi tiga perbandingan data training dan testing: 90%:10%, 80%:20%, dan 70%:30%. Nilai K yang digunakan adalah K=3 dan K=5 untuk memperoleh akurasi tertinggi. Sebelum tahap pengujian, dilakukan preprocessing pada data, yang menunjukkan adanya 201 data yang hilang pada 11 atribut dataset. Pada pengujian dengan K=3, akurasi tertinggi diperoleh pada perbandingan 70%:30% sebesar 89,17%. Sedangkan pada K=5, akurasi terbaik diperoleh pada perbandingan 90%:10% sebesar 93,54%. Dengan demikian,

akurasi terbaik tercapai pada K=5 dan perbandingan data training dan testing 90%:10%.

2.2 Stroke

Stroke menjadi penyebab utama kematian di daerah perkotaan (Kompas, 3 Desember 2008). Berdasarkan data dari hasil Riset Kesehatan Dasar (Riskesdas) tahun 2018 yang menyatakan bahwa prevalensi stroke berdasarkan diagnosis dokter pada penduduk dengan umur lebih dari 15 tahun yaitu sebesar 10,9% dengan angka mencapai 713.783. Sedangkan prevalensi berdasarkan tempat tinggal yaitu sebesar 12,6% di perkotaan dengan jumlah mencapai 394.850 dan 8,8% di pedesaan dengan jumlah 318.933.

Menurut World Health Organization (WHO), stroke merupakan kumpulan gejala dan tanda klinis yang disebabkan oleh gangguan fungsi otak, baik secara fokal maupun global, yang terjadi secara mendadak dan berkembang dengan cepat. Gejala-gejala tersebut berlangsung selama 24 jam atau lebih, atau dapat berujung pada kematian. Gangguan fungsi syaraf pada stroke dapat menimbulkan berbagai gejala, antara lain kelumpuhan pada wajah atau anggota badan, perubahan kesadaran, kesulitan berbicara atau bicara yang tidak jelas (pelo), gangguan penglihatan, dan lainnya. (Riskesdas, 2018).

Stroke dapat terjadi saat pasokan darah ke bagian otak terhambat karena adanya bekuan darah maupun terhenti diakibatkan oleh pembuluh darah yang pecah. Hal tersebut berakibat pada terhambatnya suplai oksigen maupun nutrisi bagi jaringan sel-sel saraf otak bahkan dapat berhenti sekaligus. Ketika hal tersebut terjadi dalam beberapa menit maka sel-sel di dalam otak pun akan mati.

Sel-sel otak yang mati menyebabkan berbagai fungsi otak terganggu sehingga muncul beberapa gejala stroke seperti gangguan gerak, gangguan intelektual, gangguan penglihatan, serta terdapat beberapa kelumpuhan pada salah satu sisi.

Terdapat dua metode dalam diagnosis stroke yang umum dilakukan yaitu melalui *Computed Tomography Scanning* (CT scan) dan *Magnetic Resonance Imaging* (MRI). Pada alat yang digunakan untuk CT scan disuntikkan zat warna ke dalam pembuluh darah (vena) sehingga sinar X dapat menciptakan gambaran berupa 3 dimensi dari pembuluh darah otak dan leher pasien. Dari gambaran tersebut dapat diketahui adanya kecacatan pembuluh darah maupun penyempitan pada arteri. Sedangkan pada MRI akan didapatkan gambaran 3 dimensi otak dengan menggunakan medan magnet yang kuat sehingga mampu mendeteksi adanya kerusakan jaringan yang biasa disebabkan oleh stroke jenis iskemik.

Berdasarkan penyebabnya, stroke dibagi menjadi dua tipe yaitu stroke iskemik dan stroke hemoragik. Stroke iskemik terjadi ketika terdapat gumpalan darah yang terbawa bersamaan dengan aliran darah menuju otak dan terhenti di otak sehingga suplai oksigen dan zat gizi menuju sel-sel saraf otak terhambat. Hal tersebut mengakibatkan otak kekurangan oksigen maupun darah. Stroke jenis iskemik ini menjadi tipe stroke yang umum terjadi dengan presentase mencapai 80% dari semua kasus stroke yang ada. Sedangkan stroke hemoragik dapat terjadi ketika pembuluh darah di otak pecah sehingga darah mengalir ke jaringan neural di sekitar pembuluh darah (*Romawi Press*). Stroke ini juga dapat disebabkan oleh adanya hipertensi yang terjadi bertahun-tahun dan berdampak pada kerapuhan

dinding pembuluh darah. Terjadi kelainan komponen darah juga dapat menyebabkan stroke jenis hemoragik ini.

Berdasarkan penyebab terjadinya, stroke iskemik terdiri dari *thrombosis* stroke dan *embolic* stroke. Pada *thrombotic* stroke terbentuk sumbatan (*thrombus*) yang menghalangi aliran darah pada salah satu arteri yang berperan dalam suplai darah ke otak. *Thrombus* yang terbentuk dapat berupa suatu bekuan darah, minyak, gelembung udara, sel-sel tumor ataupun kombinasi dari sumbatan-sumbatan tersebut. *Thrombus* dan *embolic* dikatakan mirip karena faktor penyebab yang sama namun dibedakan pada tempatnya. *Embolic* stroke dapat dikatakan *thrombus* yang telah mengalami perpindahan tempat dari suatu pembuluh yang besar menuju pembuluh lainnya yang lebih kecil. Jenis iskemik yang lainnya yaitu terjadi karena pelepasan *neurotransmitter eksitatorik* yang paling menonjol di otak. Ketika pembuluh darah ke otak tersumbat, sel-sel otak akan mengalami *deprivasi* darah yang berakibat pada lepasnya *reseptor glutamate* (berupa NMDA), kemudian memicu reseptor glutamat pada selaput neuron untuk aktif secara berlebihan dan terjadi perpindahan Na^+ dan Ca^{2+} ke neuron pos sinaptik sehingga terjadi penyebaran toksik ke neuron lain. Penyebaran toksis tersebut yang mengakibatkan kematian sel-sel otak dan terjadi stroke.

Berdasarkan penyebabnya, stroke hemoragik dapat dibagi menjadi dua jenis yaitu *cerebral edema* dan *subarachnoid hemorrhage*. *Cerebral edema* atau pendarahan serebral terjadi karena adanya tonjolan yang pecah. Dari tonjolan tersebut dapat diketahui bahwa terdapat kelemahan lapisan pembuluh darah arteri yang merupakan tanda kelainan pembuluh darah arteri otak yang disebut

aneurisma. Sedangkan *subarachnoid hemorrhage* terjadi saat darah dari pembuluh darah abnormal yang pecah terkonsentrasi di bawah selaput *arachnoid*. Hal tersebut juga dapat memicu kemunculan *aneurisma*.

2.3 *Gray Level Co-occurrence Matrix (GLCM)*

Gray Level Co-occurrence Matrix (GLCM) merupakan matriks frekuensi yang terdiri dari pemisahan dua piksel suatu vektor pada citra. Distribusi frekuensi dalam matriks tersebut dipengaruhi oleh derajat sudut (θ) dan jarak (d) antara dua piksel (x_1, y_1) dan (x_2, y_2) . *Gray Level Co-occurrence Matrix* juga dikenal dengan matriks ketergantungan tingkat abu-abu spasial (SGLD) $P d \theta (i, j)$ yang dapat dihitung untuk beragam orientasi dan jarak. *Gray Level Co-occurrence Matrix* diusulkan pertama kali oleh (Haralick et al., 1973) dengan 6 fitur utama dalam penggalan ciri tekstur untuk merepresentasikan pola spasial yaitu *angular second moment (ASM)* atau *energy*, *entropy*, *dissimilarity*, *contrast (inertia)*, *correlation*, *inverse different moment (IDM)* atau *homogeneity*, dan *autocorrelation*. Berikut merupakan 6 fitur ekstraksi ciri GLCM yang mengacu pada Situmorang et al. (2019).

a. *Angular Second Moment (ASM)*

Angular Second Moment menggambarkan keseragaman distribusi intensitas piksel dalam citra. Semakin tinggi nilai energi, semakin homogen distribusi intensitas pikselnya. Fitur ini menunjukkan tingkat tekstur kasar atau halus dalam citra.

b. *Contrast (inertia)*

Contrast menggambarkan tingkat perbedaan intensitas piksel antara pasangan piksel yang berdekatan. Semakin besar nilai kontras, semakin tinggi perbedaan intensitas piksel dalam citra. Fitur ini membantu dalam mendeteksi batas antara objek atau pola yang berbeda dalam citra.

c. *Correlation*

Correlation menggambarkan ketergantungan linear antara intensitas piksel pada jarak dan arah tertentu. Nilai korelasi mendekati 1 menunjukkan adanya hubungan linear positif antara intensitas piksel, sementara nilai mendekati -1 menunjukkan adanya hubungan linear negatif. Fitur ini berguna untuk mengidentifikasi hubungan spasial antara piksel dalam citra.

d. *Dissimilarity*

Dissimilarity menggambarkan tingkat ketidakmiripan suatu tekstur. Nilai *dissimilarity* akan besar jika acak dan akan bernilai kecil jika seragam.

e. *Entropy*

Entropy menggambarkan tingkat ketidakpastian atau keragaman dalam distribusi intensitas piksel pada citra. Jika nilai *entropy* tinggi, maka distribusi intensitas pikselnya akan kompleks dan acak.

f. *Homogeneity* atau *Inverse Difference Moment (IDM)*

Homogeneity menggambarkan tingkat keseragaman intensitas piksel pada jarak dan arah tertentu. Nilai ini sangat sensitif terhadap nilai disekitar diagonal utama. Nilai *homogeneity* yang mendekati 1 menandakan bahwa intensitas piksel

yang berdekatan cenderung mempunyai perbedaan yang kecil. Oleh karena itu, citra dapat memiliki tekstur yang halus.

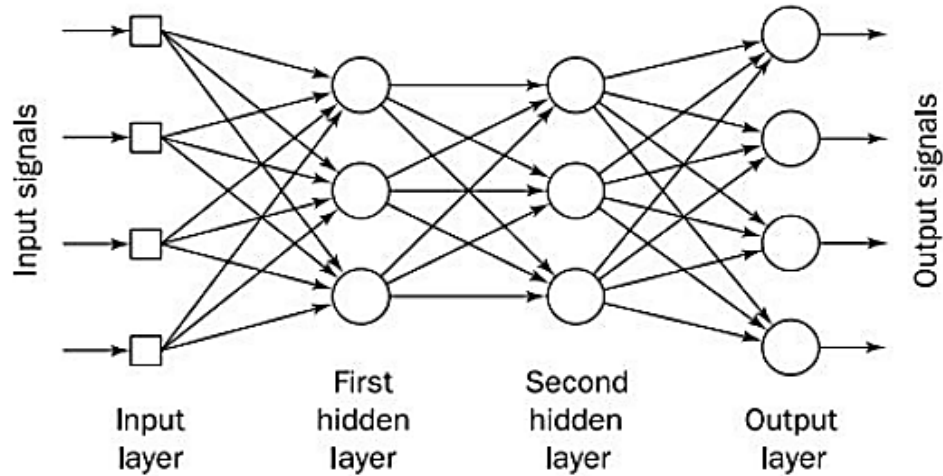
Secara sederhana proses dari pembuatan *matrix* GLCM diawali dengan pembuatan *framework matrix*. Dari *framework* tersebut dapat dibuat *co-occurrence matrix* dilanjutkan dengan pembuatan *symmetric matrix* yaitu penjumlahan antara *co-occurrence matrix* dengan *transpose matrix* tersebut. Proses selanjutnya dilakukan normalisasi *matrix* yang akan menghasilkan nilai *matrix* dalam rentang 0 sampai 1.

2.4 *Multi Layer Perceptron*

Multi Layer Perceptron (MLP) merupakan algoritma pada pembelajaran terawasi untuk melakukan klasifikasi dua jenis menggunakan fungsi yang menentukan suatu data termasuk dalam suatu kelompok atau tidak. Algoritma ini digunakan untuk mempelajari pola-pola yang kompleks dan non-linear pada data input. Karakteristik yang dimiliki oleh *Multi Layer Perceptron* yaitu keunggulan dalam penentuan nilai bobot yang lebih baik. Metode *Multi Layer Perceptron* (MLP) digunakan untuk berbagai macam tugas dalam bidang machine learning terutama dalam masalah yang melibatkan klasifikasi data ke dalam beberapa kelas atau kategori.

Multi Layer Perceptron merupakan jenis dari model perceptron yang dikemukakan oleh Rosenblatt pada tahun 1950 (Ramchoun et al., 2016). *Multi Layer Perceptron* ini termasuk model jaringan syaraf tiruan berupa system komputasi otak karena terinspirasi dari kerja otak manusia dengan memberikan

input, mengolah, kemudian menghasilkan output yang diperoleh dari proses pembelajaran sistemnya (Ewees et al., 2020).



Gambar 2.1 Arsitektur MLP

Sebuah MLP seperti pada Gambar 2.1 memiliki i layer dengan masing-masing layer berisi neuron (lingkaran). MLP menerima *input* data satu dimensi dan meneruskan data tersebut pada jaringan hingga menghasilkan *output*. Arsitektur metode ini umumnya terdiri dari beberapa lapisan yang memiliki proses masing-masing. Lapisan pertama adalah *input layer* yang menerima masukan data dari sistem bisa berupa data numerik, citra, ataupun teks. Lapisan kedua merupakan *hidden layer* suatu lapisan yang setiap unitnya tersembunyi dan hasil prosesnya tidak bisa langsung diamati atau dibaca. Lapisan terakhir merupakan *output layer* dimana lapisan ini berisi hasil proses yang dipecahkan oleh *Multi Layer Perceptron* (Amrizal & Aini, 2013).

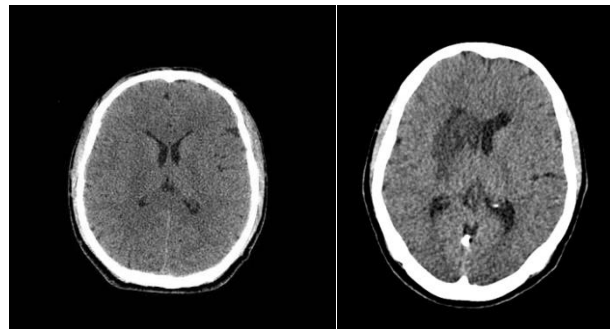
BAB III

DESAIN DAN IMPLEMENTASI

Bab ini menjabarkan tentang perancangan sistem serta pengimplementasian terhadap penelitian. Dalam penelitian ini terdiri dari tahapan pemenuhan kebutuhan sistem yang dirancang untuk memecahkan dan memberi solusi terhadap permasalahan terkait Deteksi Stroke Berbasis Citra *Computed Tomography Scan* menggunakan *Gray Level Co-occurrence Matrix* dan *Multi Layer Perceptron*.

3.1 Pengumpulan Data

Data yang digunakan dalam penelitian ini berasal dari sumber resmi Kaggle, yang tersedia di "<https://www.kaggle.com/datasets/chelebi/strokedataset>" dan dapat diakses secara terbuka oleh publik. Dataset terdiri dari dua kelompok, yaitu data pelatihan (*training*) dan data pengujian (*testing*) untuk citra *CT scan* otak. Citra tersebut dikategorikan ke dalam dua kategori, yaitu stroke dan normal. Gambar 3.1 digunakan sebagai contoh representatif dari data citra untuk kedua kategori tersebut yang akan digunakan sebagai sampel dalam tahap proses selanjutnya.

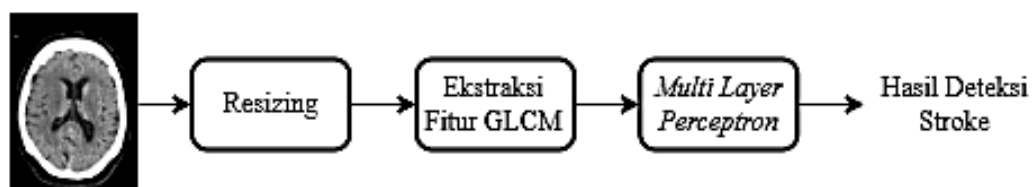


Gambar 3.1 Contoh citra otak normal dan stroke

Dataset pada data latih terdiri dari 5830 citra yang terbagi ke dalam dua kelas yaitu 3322 citra CT scan otak normal dan 2508 citra CT scan otak stroke. Sementara itu, dataset pada data uji terdiri dari 1494 citra yang terbagi ke dalam dua kelas dengan jumlah citra CT scan otak normal sebanyak 783 dan citra CT scan otak stroke sebanyak 711. Data citra ini hanya dalam format file ‘.jpg’. Dataset tersebut selanjutnya akan digunakan sebagai input untuk melatih dan menguji model deteksi dengan menerapkan metode *Multi Layer Perceptron* (MLP).

3.2 Desain Sistem

Dalam penelitian ini dirancang sebuah sistem yang melibatkan beberapa proses sebagaimana dijabarkan pada Gambar 3.2 berikut.



Gambar 3.2 Desain Sistem

Sistem yang dirancang akan dikembangkan berdasarkan pada alur desain yang ditunjukkan pada Gambar 3.2. Sistem ini akan diimplementasikan

menggunakan bahasa pemrograman Python, dengan dataset citra CT scan otak sebagai data input. Sebelumnya data akan terbagi menjadi data latih dan data uji dimana data latih diperuntukkan sebagai pembelajaran mesin, sedangkan data uji diperuntukkan sebagai penguji cara kerja sistem. Data yang telah dikelompokkan akan dilanjutkan pada pemrosesan *resizing* dan *Gray Level Co-occurrence Matrix* (GLCM). Tahapan *resizing* pada citra berguna agar ukuran citra menjadi lebih minim untuk mempermudah dan meringankan proses komputasi yang berjalan. Tahapan ekstraksi fitur yaitu penerapan metode *Gray Level Co-occurrence Matrix* (GLCM) yang hasil ekstraksinya akan digunakan sebagai input dalam model deteksi. Hasil ekstraksi fitur yang berupa data numerik tersebut menjadi inputan pada proses pelatihan model dan akan dilakukan pemisahan label dari fitur-fiturnya untuk digunakan sebagai perbandingan terhadap hasil prediksi. Pada label normal dan stroke diberikan perlakuan label encoded dengan library untuk menjadikan label teks tersebut menjadi label numerik yaitu 0 untuk normal dan 1 untuk stroke. Split data juga dilakukan pada data input tersebut dengan *train_test_split* dengan perbandingan rasio data 70:30, 80:20, dan 90:10. Tahap berikutnya adalah pengembangan model deteksi dengan menggunakan metode *Multi Layer Perceptron* (MLP). Dengan penerapan *Multi Layer Perceptron* (MLP), fitur-fitur yang telah didapat dari tahapan ekstraksi fitur akan dianalisis untuk memperoleh informasi keberadaan maupun ketiadaan stroke guna dilaporkan sebagai informasi kesehatan pasien berdasarkan CT Scan.

3.2.1 Preprocessing

Data yang sudah dikumpulkan sebelumnya perlu melewati tahap prapemrosesan (*preprocessing*) sebelum dapat digunakan untuk tahap selanjutnya. *Preprocessing* dilakukan untuk mempersiapkan data agar memenuhi kebutuhan penelitian. Dalam penelitian ini, *preprocessing* mencakup proses *resizing*. Data citra pada penelitian ini menjadi input yang akan diteruskan melewati proses *resizing* tersebut. Teknik *resizing* gambar adalah menyesuaikan ukuran gambar dengan membuat gambar lebih kecil atau lebih besar, serta dapat mengubah rasio aspeknya. *Resizing* diperlukan karena citra masih memiliki ukuran piksel yang tidak seragam sehingga perlu penyamaan dengan ukuran piksel kecil agar memudahkan proses komputasi (Rezka et al., 2017). Pada proses ini besar ukuran piksel suatu citra akan diubah menjadi ukuran 128 x 128 piksel.

Pada proses *resizing* ini digunakan metode interpolasi bilinear yang mengacu pada penelitian yang dilakukan Parsania & Virparia, (2015) dengan penggunaan library. Interpolasi bilinear memperkirakan nilai piksel yang baru dari garis lurus antar dua nilai piksel yang paling dekat. Perhitungan dimulai dengan meletakkan piksel dalam koordinat (x, y) yang berada diantara empat piksel tetangga dengan koordinat (x_1, y_1) , (x_2, y_1) , (x_1, y_2) , (x_2, y_2) . Interpolasi bilinear akan mengambil rata-rata bobot dari 4 piksel tetangga sehingga didapatkan nilai akhir interpolasi. Jika jarak piksel keseluruhan diketahui sama, maka nilai interpolasi bisa didapatkan dari rata-rata jarak piksel tersebut. Metode ini melakukan interpolasi dari dua arah yaitu horizontal dan vertikal yang dianggap dapat memberikan hasil yang lebih baik dengan waktu komputasi yang lebih singkat.

Dengan menentukan kesamaan ukuran piksel yang berlaku pada semua citra, langkah ini memastikan bahwa semua data citra yang akan digunakan memiliki dimensi konsisten yang diperlukan untuk melatih dan menguji model pembelajaran sistem dengan ketentuan ukuran masukan atau input tetap.

3.2.2 Ekstraksi Fitur Gray Level Co-occurrence Matrix (GLCM)

Gray Level Co-occurrence Matrix (GLCM) adalah *array* dua dimensi yang setiap dimensi *array*-nya sama dengan jumlah level *grayscale* pada citra. Proses GLCM ini adalah salah satu metode yang efektif dalam menggambarkan dan mengidentifikasi sifat tekstur dalam citra. Metode ekstraksi fitur tekstur ini memetakan citra masukan ke dalam tabel yang menghitung jumlah kemunculan sepasang nilai piksel dalam jarak dan arah tertentu. Nilai sudut bervariasi pada rentang 0° , 45° , 90° , dan 135° dan variasi jarak dapat dipilih jarak sebesar 1 dan 2.

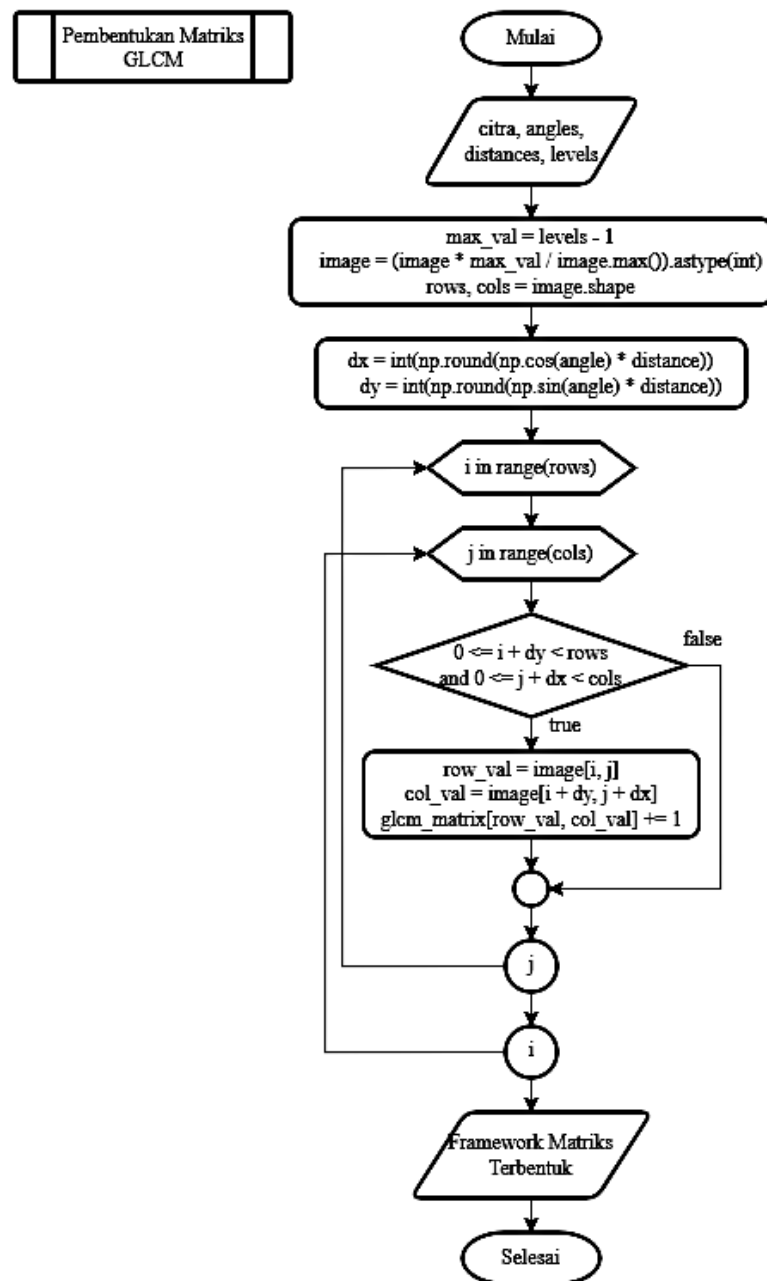
Langkah-langkah penerapan ekstraksi fitur GLCM dimulai dengan pembentukan matriks GLCM dengan citra CT scan otak sebagai inputannya. Setelah didapatkan bentuk dari framework matrix akan dilakukan penghitungan normalisasi matriks. Nilai normalisasi tersebut akan diteruskan untuk mendapatkan keenam nilai ekstraksi fitur dari *asm*, *contrast*, *correlation*, *dissimilarity*, *entropy*, dan *homogeneity*. Tahapan-tahapan tersebut dijabarkan seperti pada Gambar 3.3.



Gambar 3.3 Alur ekstraksi fitur GLCM

Pada proses awal ekstraksi fitur GLCM ini citra skala abu-abu diambil sebagai input proses GLCM. Dilakukan inisialisasi parameter dengan cara menentukan level skala keabuan, nilai jarak dan sudut untuk menghitung matriks GLCM. Jarak yang dapat digunakan bisa bernilai 1, 2, dan seterusnya sesuai kebutuhan namun pada penelitian ini digunakan jarak 1 dan 2. Terdapat 4 arah sudut yang digunakan dalam fitur tekstur ini yaitu, 0° , 45° , 90° , dan 135° . Tahapan selanjutnya yaitu melakukan penghitungan matriks GLCM dengan menskalakan intensitas citra terlebih dahulu ke rentang level *grayscale* yang sudah diinisialisasi. Kemudian menentukan pergeseran piksel (dx , dy) berdasarkan jarak dan sudut yang telah ditentukan. Dilakukan iterasi di setiap indeks baris dan kolom untuk piksel citranya, setelahnya dapat ditambahkan pasangan nilai piksel dari iterasi tersebut ke dalam matriks $p(i, j|d, \theta)$ dengan i

adalah tingkat keabuan piksel referensi, j adalah tingkat kabuan piksel tetangga, d merupakan jarak antar piksel dan θ merupakan sudut hubungan spasial antar piksel. Terbentuklah *framework* matrix beserta nilainya. Proses pembentukan matriks GLCM tersebut dijabarkan pada Gambar 3.4.



Gambar 3.4 Alur pembentukan matriks GLCM

Setelah terbentuk framework matriksnya, matriks GLCM tadi dinormalisasi untuk mendapatkan distribusi probabilitas. Sebelum itu pastikan matriks GLCM tersebut simetris dengan cara menjumlahkan matriks GLCM dengan matriks transposenya seperti pada persamaan 3.1.

$$glcm_symmetric = glcm_matrix + glcm_matrix.T \quad (3.1)$$

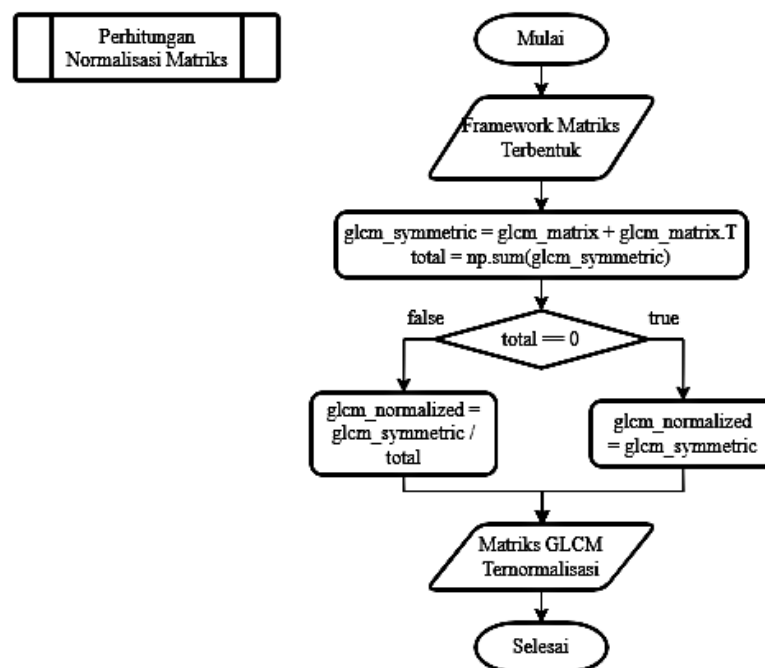
Keterangan:

$glcm_matrix.T$ = transpose dari matriks glcm

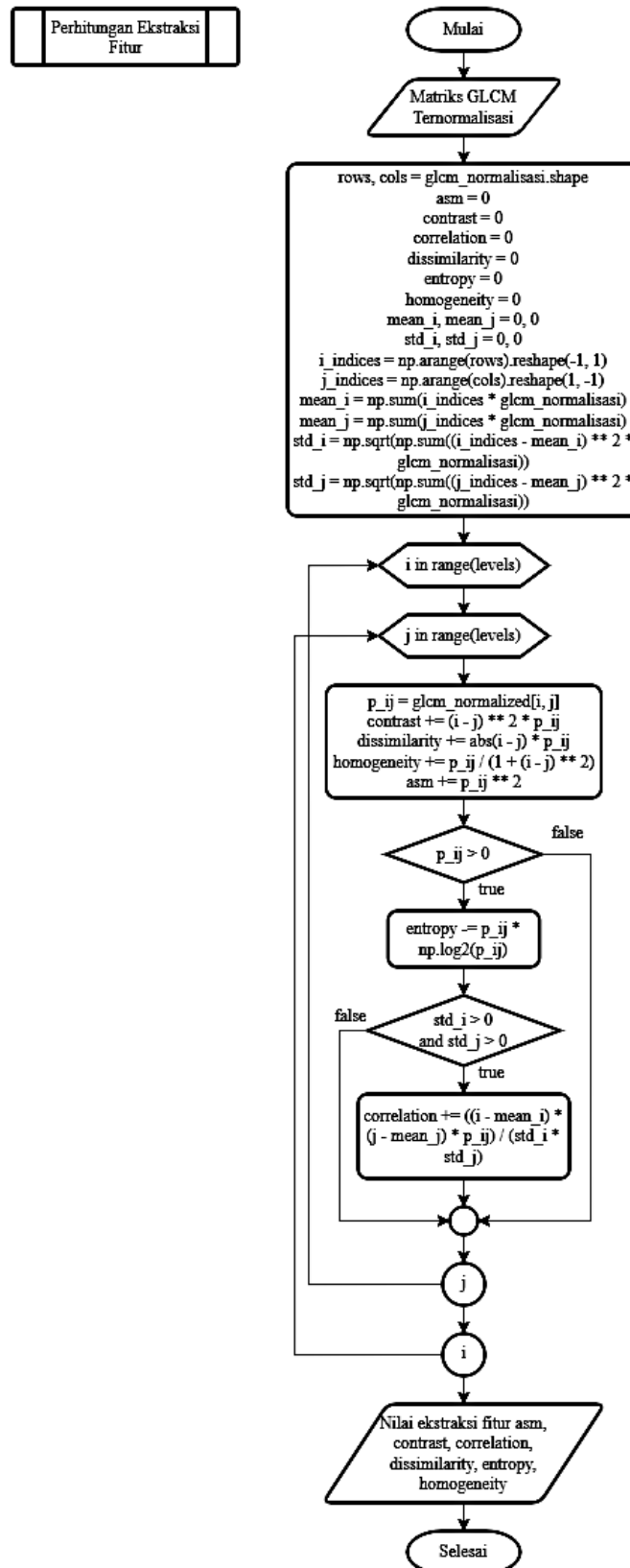
Setelah matriks GLCM simetris diperoleh, langkah berikutnya adalah melakukan normalisasi dengan menghitung total jumlah elemen dalam matriks. Setiap elemen matriks kemudian dibagi dengan total jumlah elemen tersebut untuk mendapatkan nilai normalisasi yang dapat dilihat pada persamaan 3.2.

$$normalisasi = \frac{\text{nilai matrix GLCM}}{\sum_i^N \text{nilai matrix GLCM}} \quad (3.2)$$

Tahapan perhitungan normalisasi matriks tersebut dapat dilihat pada Gambar 3.5.



Gambar 3.5 Alur perhitungan normalisasi matriks



Gambar 3.6 Alur perhitungan ekstraksi fitur GLCM

Langkah terakhir dalam proses ekstraksi fitur GLCM yaitu melakukan penghitungan terhadap beberapa statistik, yaitu ASM, contrast, correlation, dissimilarity, entropy, dan homogeneity. Penghitungan sejumlah statistik tersebut dapat dilihat pada persamaan 3.3 sampai persamaan 3.8 dan dijabarkan pada Gambar 3.6. Berikut merupakan penghitungan 6 fitur ekstraksi ciri GLCM yang mengacu pada Situmorang et al. (2019).

a. *Angular Second Moment (ASM)*

$$asm = \sum_{i,j} p(i,j)^2 \quad (3.3)$$

Keterangan:

$p(i,j)$ = probabilitas kemunculan pasangan piksel

Asm menggambarkan keseragaman distribusi intensitas piksel dalam citra. Semakin tinggi nilai energi, semakin homogen distribusi intensitas pikselnya. Fitur ini menunjukkan tingkat tekstur kasar atau halus dalam citra. Perhitungan ekstraksi fitur asm dilakukan dengan mengkuadratkan nilai probabilitas kemunculan pasangan piksel baris dan kolomnya. Hasil yang didapatkan untuk masing-masing indeks baris dan kolom akan dijumlahkan seluruhnya sebagai nilai akhir ekstraksi fitur. Perhitungan ASM tersebut dapat dilihat pada persamaan 3.3

b. *Contrast (inertia)*

$$contrast = \sum_{i,j} (i - j)^2 \cdot p(i,j) \quad (3.4)$$

Keterangan:

$p(i,j)$ = probabilitas kemunculan pasangan piksel

$(i - j)$ = pengurangan indeks baris ke-i dan kolom ke-j

Contrast menggambarkan tingkat perbedaan intensitas piksel antara pasangan piksel yang berdekatan. Semakin besar nilai kontras, semakin tinggi

perbedaan intensitas piksel dalam citra. Fitur ini membantu dalam mendeteksi batas antara objek atau pola yang berbeda dalam citra. Perhitungan ekstraksi fitur contrast dilakukan dengan langkah awal mengkuadratkan nilai dari pengurangan baris dan kolom. Kemudian hasil pengkuadratan tersebut akan dikalikan dengan nilai probabilitas kemunculan pasangan piksel i, j . Hasil yang didapatkan untuk masing-masing indeks baris dan kolom akan dijumlahkan seluruhnya sebagai nilai akhir ekstraksi fitur. Perhitungan tersebut ditunjukkan pada persamaan 3.4.

c. *Correlation*

$$correlation = \sum_{i,j} \frac{(i - \mu_x) \cdot (j - \mu_y) \cdot p(i,j)}{\sqrt{\delta_x \delta_y}}$$

$$\mu_x = \sum_{i,j} i \cdot p(i,j)$$

$$\mu_y = \sum_{i,j} j \cdot p(i,j)$$

(3.5)

$$\delta_x = \sum_{i,j} (i - \mu_x)^2 \cdot p(i,j)$$

$$\delta_y = \sum_{i,j} (j - \mu_x)^2 \cdot p(i,j)$$

Keterangan:

$p(i,j)$ = probabilitas kemunculan pasangan piksel

i = indeks baris ke- i

j = indeks kolom ke- j

μ_x = rata-rata untuk nilai elemen baris

μ_y = rata-rata untuk nilai elemen kolom

δ_x = standar deviasi untuk elemen baris

δ_y = standar deviasi untuk elemen kolom

Correlation menggambarkan ketergantungan linear antara intensitas piksel pada jarak dan arah tertentu. Nilai korelasi mendekati 1 menunjukkan adanya hubungan linear positif antara intensitas piksel, sementara nilai mendekati -1

menunjukkan adanya hubungan linear negatif. Fitur ini berguna untuk mengidentifikasi hubungan spasial antara piksel dalam citra. Perhitungan ekstraksi fitur correlation dilakukan dengan langkah awal mencari nilai rata-rata dan standar deviasi dari indeks baris dan kolom. Indeks baris dan kolom akan dikurangi dengan masing-masing nilai rata-ratanya kemudian dikalikan bersamaan dengan probabilitas kemunculan pasangan pikselnya. Langkah terakhir yaitu hasil perkalian dengan probabilitas sebelumnya akan dibagi dengan akar dari perkalian standar deviasi masing-masing baris dan kolom. Hasil yang didapatkan untuk masing-masing indeks baris dan kolom akan dijumlahkan seluruhnya sebagai nilai akhir ekstraksi fitur. Perhitungan correlation tersebut dapat dilihat pada persamaan 3.5

d. *Dissimilarity*

$$dissimilarity = \sum_{i,j} |i - j| \cdot p(i,j) \quad (3.6)$$

Keterangan:

$p(i,j)$ = probabilitas kemunculan pasangan piksel

$(i - j)$ = pengurangan indeks baris ke-i dan kolom ke-j

Dissimilarity menggambarkan tingkat ketidakmiripan suatu tekstur. Nilai *dissimilarity* akan besar jika acak dan akan bernilai kecil jika seragam. Perhitungan ekstraksi fitur *dissimilarity* dilakukan dengan cara mengurangi indeks baris dan kolom kemudian dikalikan dengan probabilitas kemunculan pasangan pikselnya. Hasil yang didapatkan untuk masing-masing indeks baris dan kolom akan dijumlahkan seluruhnya sebagai nilai akhir ekstraksi fitur. Perhitungan *dissimilarity* tersebut dapat dilihat pada persamaan 3.6.

e. *Entropy*

$$entropy = \sum_{i,j} p(i,j) \cdot \log_2(p(i,j)) \quad (3.7)$$

Keterangan:

$p(i,j)$ = probabilitas kemunculan pasangan piksel
 \log_2 = logaritma basis 2

Entropy menggambarkan tingkat ketidakpastian atau keragaman dalam distribusi intensitas piksel pada citra. Jika nilai *entropy* tinggi, maka distribusi intensitas pikselnya akan kompleks dan acak. Perhitungan ekstraksi fitur entropy dilakukan dengan cara mengalikan penjumlahan dari frekuensi kemunculan pasangan piksel i dan j dengan logaritma basis 2 dari frekuensi kemunculan pasangan piksel tersebut. Hasil yang didapatkan untuk masing-masing indeks baris dan kolom akan dijumlahkan seluruhnya sebagai nilai akhir ekstraksi fitur. Perhitungan entropy tersebut dapat dilihat pada persamaan 3.7

f. *Homogeneity* atau *Inverse Difference Moment (IDM)*

$$homogeneity = \sum_{i,j} \frac{p(i,j)}{1+(1-j)^2} \quad (3.8)$$

Keterangan:

$p(i,j)$ = probabilitas kemunculan pasangan piksel
 $(1-j)$ = 1 dikurangi nilai indeks kolom ke- j

Homogeneity menggambarkan tingkat keseragaman intensitas piksel pada jarak dan arah tertentu. Nilai ini sangat sensitif terhadap nilai disekitar diagonal utama. Nilai *homogeneity* yang mendekati 1 menandakan bahwa intensitas piksel yang berdekatan cenderung mempunyai perbedaan yang kecil. Oleh karena itu, citra dapat memiliki tekstur yang halus. Perhitungan ekstraksi fitur homogeneity didapat dari pembagian antara probabilitas kemunculan pasangan piksel dengan

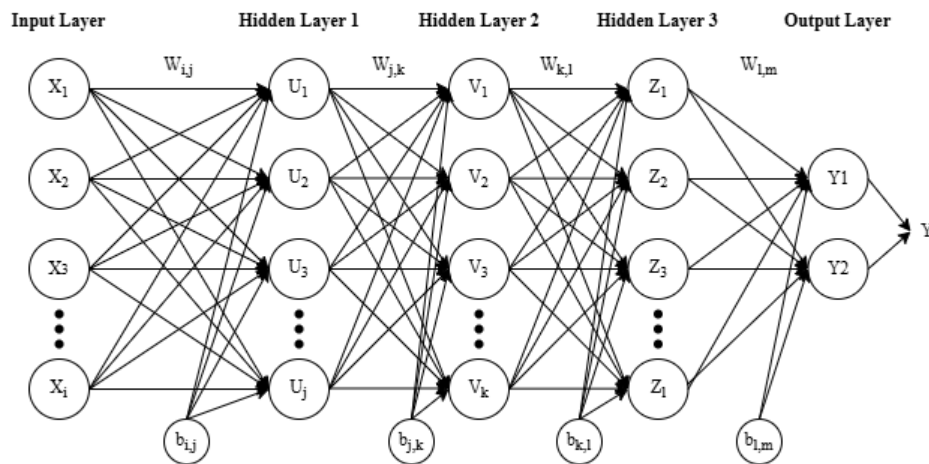
hasil pengurangan 1 dan indeks kolom yang dikuadratkan kemudian ditambah 1. Hasil yang didapatkan untuk masing-masing indeks baris dan kolom akan dijumlahkan seluruhnya sebagai nilai akhir ekstraksi fitur. Perhitungan homogeneity tersebut dapat dilihat pada persamaan 3.8. Semua proses perhitungan ekstraksi fitur tersebut berlaku untuk setiap baris dan kolom pada matriks glcm yang sudah ternormalisasi sebelumnya. Keluaran yang didapatkan yaitu nilai ekstraksi fitur *asm*, *contrast*, *correlation*, *dissimilarity*, *entropy*, dan *homogeneity* yang akan menjadi input pada proses pembelajaran model *Multi Layer Perceptron*.

3.2.3 Multi Layer Perceptron (MLP)

Dalam penelitian ini menerapkan metode *Multi Layer Perceptron* (MLP) yang banyak digunakan untuk memecahkan masalah yang membutuhkan pembelajaran terawasi serta penelitian dalam ilmu syaraf komputasi dan pemrosesan terdistribusi paralel, termasuk pengenalan pola, pengenalan suara, pengenalan gambar dan terjemahan mesin. *Multi Layer Perceptron* memiliki kelebihan yang terletak pada arsitekturnya yang sederhana serta kemudahan penggunaan algoritma pembelajaran mampu memberikan hasil yang baik. Pada *Multi Layer Perceptron*, setiap unit yang dimiliki akan saling terhubung dengan penyesuaian bobot yang diambil dari proses pembelajaran pada setiap koneksi masing-masing unit baik dalam satu lapisan maupun lebih (Suyanto, 2022).

MLP mempunyai prinsip dasar dalam penggunaan lapisan tunggal yang terorganisir menjadi beberapa lapisan. Dimulai dengan lapisan input terdiri dari nilai yang menyatakan karakter dari objek yang diperoleh pada proses penggalian

ekstraksi fitur menggunakan GLCM. Selanjutnya keluaran dari lapisan input akan diteruskan dengan lapisan tersembunyi sebanyak dua buah lapisan yang pada akhirnya akan menghasilkan keluaran.



Gambar 3.7 Arsitektur MLP

Berdasarkan arsitektur MLP yang ditunjukkan pada Gambar 3.7 maka terdapat 5 hidden layer yang terdiri dari input layer, hidden layer 1, hidden layer 2, hidden layer 3, dan output layer. *Input layer* terdiri dari neuron - neuron yang berfungsi menerima data masukan untuk mendukung proses pembelajaran dan pengenalan pada MLP dengan algoritma backpropagation. Neuron-neuron tersebut berisi data yang telah diperoleh sebelumnya, yaitu hasil ekstraksi fitur dari citra berupa data numerik nilai *asm*, *contrast*, *correlation*, *dissimilarity*, *entropy*, dan *homogeneity*. Kemudian *hidden layer* yang digunakan berupa *fully connected layer* yaitu pada tiap-tiap neuron saling terhubung dengan neuron yang sebelumnya. Unit-unit di lapisan tersembunyi menerima bobot masukan dan mentransferkannya ke lapisan keluaran melalui fungsi transfer. Output layer, pada lapisan ini berisikan neuron - neuron yang akan bereaksi terhadap data input yang diberikan. Informasi didistribusikan ke dalam jaringan melalui manipulasi data

masukan, dengan hasil dihitung pada setiap unit pemrosesan. Setiap unit pada lapisan keluaran menghitung bobot dan *error*. Proses penyebaran balik (*backpropagation*) dilakukan dari lapisan keluaran ke lapisan tersembunyi untuk mengevaluasi kesalahan dan nilai bobot yang telah dihitung, kemudian kembali ke lapisan tersembunyi untuk memperbarui bobot. Setiap neuron pada lapisan masukan memiliki bobot yang dilambangkan dengan w , di mana nilai ini akan dioperasikan dengan model penjumlahan berbobot (*weighted sum model*), yaitu dengan mengalikan bobot bersama nilai masukan. Hasil penjumlahan antara nilai masukan yang dikalikan dengan bobot tersebut kemudian ditambahkan dengan nilai bias. Nilai akhir dari operasi ini akan diterapkan dengan fungsi aktivasi sigmoid.

Dalam jaringan feedforward, sinyal masukan diteruskan dari lapisan masukan melalui lapisan tersembunyi hingga mencapai lapisan keluaran tanpa adanya koneksi balik atau mekanisme umpan balik. Pada jaringan feedforward, setiap neuron di suatu lapisan terhubung dengan semua neuron pada lapisan berikutnya, tetapi tidak memiliki koneksi balik menuju lapisan sebelumnya. Aliran informasi hanya bergerak searah, dimulai dari lapisan masukan menuju lapisan keluaran melalui lapisan-lapisan tersembunyi.

Setiap neuron pada tiap *layer* menerima sinyal masukan dari seluruh neuron di *layer* sebelumnya dan menghasilkan sinyal keluaran yang diteruskan ke seluruh neuron pada *layer* berikutnya. Setiap koneksi antar neuron mempunyai bobot yang akan menentukan tingkat kontribusi masing-masing neuron terhadap hasil keluaran. Tujuan utama dari jaringan feedforward adalah mempelajari fungsi yang

mampu memetakan masukan ke keluaran yang diharapkan. Hal ini dicapai dengan menyesuaikan bobot koneksi antar neuron selama proses pelatihan jaringan, menggunakan algoritma backpropagation.

Dalam penelitian ini, feedforward berperan untuk menghitung keluaran dari setiap lapisan pada model. Fungsi ini menerima data masukan X sebagai parameter dan meneruskannya melalui setiap lapisan dengan penerapan fungsi aktivasi di setiap lapisannya. Keluaran dari lapisan saat ini akan menjadi masukan untuk lapisan berikutnya. Pada setiap iterasi, fungsi ini menampilkan nilai X sesudah melalui lapisan tersebut. Proses ini memungkinkan untuk memahami transformasi data masukan menjadi keluaran Y pada setiap lapisan.

Fungsi aktivasi adalah fungsi yang digunakan dalam jaringan saraf untuk mengaktifkan atau menonaktifkan neuron. Fungsi aktivasi sigmoid memastikan bahwa keluaran lapisan ini selalu bernilai antara 0 dan 1, sehingga dapat digunakan sebagai lapisan keluaran model klasifikasi biner, seperti ketika memprediksi apakah seorang pasien telah mengalami stroke atau dalam keadaan normal. Fungsi aktivasi tersebut dapat berupa fungsi linier, ambang batas (*threshold*), atau *sigmoid*. Fungsi *sigmoid* umumnya digunakan karena menggabungkan perilaku linier, lengkung, dan konstan. Fungsi aktivasi sigmoid dapat diketahui pada persamaan 3.9.

$$\text{sigmoid}(x) = \frac{1}{(1+e^{-x})} \quad (3.9)$$

Keterangan:

$\text{sigmoid}(x)$ = fungsi aktivasi sigmoid untuk input x
 e = bilangan Euler (2,71828)
 e^{-x} = eksponensial dari $-x$

Algoritma yang digunakan untuk melatih model MLP adalah algoritma *backpropagation*, yang menghitung gradient loss function berdasarkan parameter model seperti bobot dan bias. Hal ini memungkinkan untuk penyesuaian bobot dan bias model agar lebih sesuai dengan data yang digunakan. Struktur *backpropagation* adalah konsep desain yang memungkinkan pemahaman dan melakukan analisis pola data sebelumnya dengan lebih akurat untuk menghasilkan keluaran yang lebih akurat dengan meminimalisir kesalahan.

Proses pelatihan merupakan langkah penting dalam mengembangkan kemampuan model *Multi Layer Perceptron* untuk mengenali pola dalam data. Mulanya bobot dan bias dalam jaringan diinisialisasi dengan nilai acak. Data pelatihan yang terdiri dari pasangan masukan dan keluaran yang diharapkan kemudian dilewatkan melalui jaringan. Keluaran yang dihasilkan dibandingkan dengan keluaran yang diharapkan dan fungsi *error* digunakan untuk mengukur kesalahan prediksi. Melalui iterasi yang berulang, jaringan dapat memperbarui parameternya dan meningkatkan kemampuannya untuk menangkap adanya pola yang kompleks. Selain itu, pemilihan fungsi aktivasi dan arsitektur jaringan merupakan faktor penting yang berperan dalam menentukan kemampuan MLP untuk memodelkan hubungan data. Hasil akhirnya adalah model MLP yang dapat digunakan untuk memprediksi dan mengklasifikasikan.

Langkah-langkah dalam membuat model *Multi Layer Perceptron* menggunakan *backpropagation* adalah sebagai berikut (Arhami dan Nasir, 2020):

1. Langkah pertama sebagai permulaan algoritma backpropagation yaitu dengan menginisialisasi bobot. Ini melibatkan asumsi nilai bobot secara acak.

2. Tahap *forward propagation*

Setiap neuron input (X_i , $i=1,2,3,\dots,n$) menerima sinyal x_i dan meneruskan sinyal tersebut ke semua neuron pada layer berikutnya (hidden layer). Setiap neuron pada hidden layer (Z_j , $j=1,2,3,\dots,n$) dijumlahkan bobot dengan neuron inputnya, ditunjukkan dengan persamaan 3.10.

$$Z_{in_j} = b_{0j} + \sum_{i=1}^n w_j x_i \quad (3.10)$$

Keterangan:

Z_{in_j} = input pada neuron ke-j
 b_{ij} = bias untuk neuron ke-j
 w_{ij} = bobot sinapsis input ke-i dengan neuron ke-j
 x_i = nilai input dari neuron ke-i

Nilai output pada neuron tersebut dihitung dengan penerapan fungsi aktivasi ditunjukkan dengan persamaan 3.11 dan selanjutnya nilai tersebut diteruskan pada keseluruhan neuron di *layer* berikutnya (*output layer*).

$$Z_j = f(Z_{in_j}) \quad (3.11)$$

Keterangan:

Z_j = output neuron ke-j pada lapisan tersebut setelah penerapan fungsi aktivasi
 Z_{in_j} = input pada neuron ke-j sebelum fungsi aktivasi diterapkan
 $f(Z_{in_j})$ = fungsi sigmoid yang diterapkan pada Z_{in_j}

Setiap neuron pada lapisan keluaran (Y) kemudian memproses sama dengan masukan pada lapisan sebelumnya, menambahkan bobot pada neuron masukan seperti terlihat pada persamaan 3.12. Kemudian, terapkan fungsi aktivasi untuk menghitung keluarannya, seperti yang ditunjukkan pada persamaan 3.13.

$$Y_{in} = b_{ij} + \sum_{j=1}^n w_{jk} z_j \quad (3.12)$$

Keterangan:

Y_{in} = input neuron pada lapisan keluaran
 b_{ij} = bias neuron ke-j.
 w_{jk} = bobot penghubung neuron ke-j dengan neuron ke-k.
 z_j = output atau aktivasi dari neuron ke-j pada lapisan sebelumnya

$$Y = f(Y_{in_k}) = \frac{1}{1 + e^{-Y_{in_k}}} \quad (3.13)$$

Keterangan:

Y = aktivasi pada neuron ke-k pada lapisan tersebut
 Y_{in_k} = nilai total input ke neuron ke-k sebelum fungsi aktivasi diterapkan
 $e^{-Y_{in_k}}$ = nilai eksponensial dari negatif Y_{in_k}
 $f(Y_{in_k})$ = fungsi sigmoid yang diterapkan pada Z_{in_j}

3. Tahap *backpropagation*

Setiap neuron pada *output layer* (Y) menerima target yang cocok dengan pola masukan pelatihan. Kesalahannya dapat dihitung dengan persamaan 3.14.

$$\delta_k = (t_k - y_k) \cdot f'(y_{in}) \quad (3.14)$$

Keterangan:

δ_k = kesalahan pada *output layer*.
 t_k = target keluaran yang diinginkan pada neuron.
 y_{in} = *input* dari *output layer* pada neuron.
 $f'(y_{in})$ = turunan dari fungsi aktivasi pada *output layer* yang mengambil *input* dari hasil perhitungan bobot dan *input* pada *output layer*.

Untuk menghitung perubahan bobot untuk update bobot dari *output layer* ke *hidden layer* sebelumnya, ditunjukkan dengan persamaan 3.15.

$$\Delta w_{jk} = \alpha \cdot \delta_k \cdot z_j \quad (3.15)$$

Keterangan:

Δw_{jk} = nilai koreksi bobot pada neuron ke-k dan neuron ke-j.
 α = *learning rate*.
 δ_k = *error* pada *output layer* untuk neuron ke-k.
 z_j = keluaran dari *layer* sebelumnya pada neuron ke-j.

Untuk meneruskan δ_k ke neuron pada lapisan paling kanan dapat dilakukan dengan cara menghitung perubahan bias yang ditunjukkan pada persamaan 3.16.

$$\Delta w_{0k} = \alpha \cdot \delta_k \quad (3.16)$$

Keterangan:

Δw_{0k} = nilai koreksi bobot dari output layer pada neuron ke-k.
 α = *learning rate*.
 δ_k = *error* pada *output layer*.

Untuk setiap neuron pada hidden layer ($Z_j, j=1,2,3,\dots,n$) dijumlahkan nilai delta input yang ditunjukkan dengan persamaan 3.10 kemudian hasilnya diberi perlakuan sama yaitu dikalikan dengan turunan aktivasinya $f(Z_{in_j})$ seperti pada persamaan 3.17.

$$\delta_{in_j} = \sum_{k=1}^m \delta_k \Delta w_{jk} \quad (3.17)$$

Keterangan:

δ_{in_j} = neuron ke-j pada *hidden layer*
 δ_k = kesalahan pada *output layer* untuk neuron ke-k
 Δw_{jk} = nilai koreksi bobot pada neuron ke-j dan neuron ke-k

Untuk nilai bobot dan bias pada output layer maupun hidden layer berlaku perhitungan dengan persamaan 3.18.

$$w_{jk(\text{baru})} = w_{jk(\text{lama})} + \Delta w_{jk} \quad (3.18)$$

Keterangan:

$w_{jk(\text{baru})}$ = nilai bobot baru
 $w_{jk(\text{lama})}$ = nilai bobot lama
 Δw_{jk} = nilai koreksi bobot pada neuron ke-j dan neuron ke-k

Dari hasil prediksi yang didapat akan dipilih probabilitas tertinggi menggunakan persamaan 3.19.

$$Y = \operatorname{argmax}(P(y|x))$$

$$\operatorname{argmax}(P(y|x)) = 1, \text{ jika } P(y = 1|x) > P(y = 0|x) \quad (3.19)$$

$$\operatorname{argmax}(P(y|x)) = 0, \text{ jika } P(y = 0|x) \geq P(y = 1|x)$$

Keterangan:

Y	= hasil akhir prediksi
$\operatorname{argmax}(P(y x))$	= fungsi untuk penentuan nilai maksimum hasil
$P(y = 1 x)$	= probabilitas kelas stroke
$P(y = 0 x)$	= probabilitas kelas normal

BAB IV

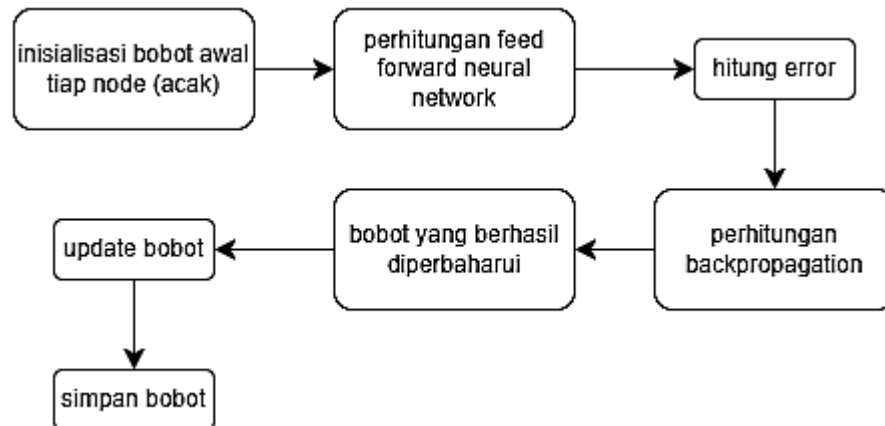
UJI COBA DAN PEMBAHASAN

Bab ini menjelaskan hasil pengujian serta pembahasan terkait Deteksi Stroke Berbasis Citra *Computed Tomography Scan* Menggunakan *Gray Level Co-occurrence Matrix* dan *Multi Layer Perceptron*.

4.1 Skenario Uji Coba

Dalam penelitian ini, dataset yang tersedia dibagi menjadi dua bagian, yaitu data training dan data testing. Proporsi dataset yang digunakan adalah 90:10, 80:20, dan 70:30 guna memperoleh sistem terbaik dengan masing-masing diberlakukan skenario yang berbeda menggunakan tiga skenario jarak (1, 2, dan kombinasi 1 dan 2) dengan melibatkan 4 sudut GLCM (0°, 45°, 90°, dan 135°). Sistem juga akan diuji dengan menerapkan 4 skenario yang melibatkan *Multi Layer Perceptron* terhadap banyaknya hidden layer (1, 2, dan 3) serta banyaknya neuron (32, 64, 128, 256) pada hidden layer.

Selanjutnya penentuan nilai awal variabel yang penting untuk penghitungan, yaitu laju pembelajaran (*learning rate*), dan jumlah *epoch* maksimum. Selain itu, inisialisasi pada bobot awal dan bias awal ditentukan secara acak. Representasi proses pelatihan metode *Multi Layer Perceptron* ditunjukkan pada Gambar 4.1.

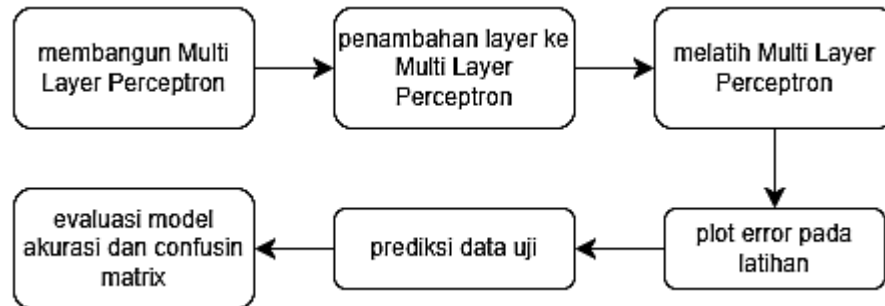


Gambar 4.1 Alur proses training MLP algoritma backpropagation

Setelah kesalahan dihitung, perubahan bobot dilakukan dengan mengalikan delta (selisih antara keluaran yang diharapkan dan keluaran yang didapatkan kemudian dikali dengan turunan fungsi aktivasi) dengan masukan dari lapisan sebelumnya. Kalikan perubahan bobot dengan laju pembelajaran untuk membatasi ukuran langkah di setiap iterasi dan menghindari *overfitting*. Dalam penelitian ini modifikasi perhitungan kesalahan bobot dilakukan untuk meminimalkan kesalahan yang didapat. Melakukan iterasi melalui lapisan-lapisan dalam jaringan dan memperbarui bobot untuk setiap layer merupakan cara meminimalisir kesalahan tadi. Setelah pelatihan selesai dilakukan, jaringan dapat digunakan untuk mengenali pola dengan menggunakan fase propagasi maju saja untuk menentukan keluaran jaringan yang berlaku persamaan 3.13 dan 3.14.

Tujuan dari proses pengujian algoritma jaringan saraf adalah untuk mengevaluasi kemampuan model yang telah dilatih dalam mengaplikasikan pengetahuannya pada data yang belum pernah dilihat sebelumnya. Setelah proses pembuatan model, proses pengujian diawali dengan menambahkan layer sesuai skenario pengujian yang akan dilakukan termasuk jumlah hidden layer beserta

jumlah neuron yang dibutuhkan kemudian model dilatih dan dilakukan plot error hingga mendapatkan nilai error yang diharapkan kemudian akan menghasilkan suatu prediksi dari data uji tersebut. Ilustrasi dari alur proses pengujian algoritma backpropagation MLP dapat ditemukan pada Gambar 4.2.



Gambar 4.2 Alur proses testing MLP algoritma backpropagation

Setelah diperoleh hasil prediksi data uji, perlu adanya evaluasi terhadap kinerja model klasifikasi yang dihitung dengan *confusion matrix*. Dalam *confusion matrix*, hasil prediksi model akan dibandingkan dengan nilai sebenarnya pada data uji atau validasi. Pada langkah ini, jumlah data yang diklasifikasikan dengan benar atau salah akan ditampilkan dalam setiap kategori termasuk kategori negatif dan positif. *Confusion matrix* memiliki empat kategori sebagai berikut :

1. TP (*true positive*) diartikan sebagai jumlah data stroke yang diklasifikasikan benar sebagai positif oleh model klasifikasi.
2. FP (*false positive*) diartikan sebagai jumlah data stroke yang diklasifikasikan salah sebagai positif oleh model klasifikasi.
3. TN (*true negative*) diartikan sebagai jumlah data stroke yang diklasifikasikan benar sebagai negative oleh model klasifikasi.

4. FN (*false negative*) diartikan sebagai jumlah data stroke yang diklasifikasikan salah sebagai negative oleh model klasifikasi.

Dari empat kategori di atas, dapat dihitung akurasi *confusion matrix* dengan rumus sebagai berikut merujuk pada penelitian oleh (Raden Roro Ayuni Purbo Okta Briliani & Irma Palupi, 2022) :

1. Akurasi

Akurasi adalah proporsi jumlah prediksi benar dengan kondisi sebenarnya.

Perhitungan akurasi dinyatakan pada persamaan 4.1.

$$Akurasi = \frac{TP+TN}{TP+FP+TN+FN} \times 100\% \quad (4.1)$$

2. Presisi

Presisi adalah perbandingan data relevan yang dipilih terhadap semua data yang terpilih dengan kata lain presisi menghitung adanya kecocokan antara data yang diminta dengan jawaban dari permintaan data tersebut. Perhitungan presisi dinyatakan pada persamaan 4.2.

$$Presisi = \frac{TP}{TP+FP} \times 100\% \quad (4.2)$$

3. Recall

Recal adalah perbandingan dari data yang berkaitan yang dipilih terhadap jumlah keseluruhan data yang tersedia. Perhitungan recall dinyatakan pada persamaan 4.3.

$$Recall = \frac{TP}{TP+FN} \times 100\% \quad (3.12)$$

Pada langkah pertama pengujian, persiapan data uji dilakukan dengan mempersiapkan dataset citra yang terdapat citra stroke maupun normal. Data

tersebut terbagi menjadi dua folder dengan jumlah data citra stroke sebanyak 3219 dan jumlah data citra normal sebanyak 4105 yang keduanya berbasis citra CT scan. Sejumlah citra tersebut diproses dengan melibatkan perhitungan nilai matriks menggunakan beberapa sudut, jarak, dan ekstraksi fitur GLCM. Ekstraksi fitur tersebut termasuk langkah penting guna menganalisis citra sebagai bentuk interpretasi tekstur citra bentuk numerik yang terbagi menjadi 6 ekstraksi fitur. Pada pengujian ini data akan terbagi menjadi data latih dan data uji dengan variasi rasio yaitu 70:30, 80:20, dan 90:10.

Pembagian data latih dan data uji dengan variasi rasio tersebut dapat dilakukan dengan memberikan inisialisasi variabel sesuai rasio yang dibutuhkan pada proporsi data yang akan digunakan untuk melatih dan menguji model sistem. Dalam penelitian ini total dari seluruh data citra yang digunakan sebanyak 7324 citra. Pada data citra tersebut terdapat sebanyak 2508 citra stroke pada folder training dan 711 citra stroke pada folder testing sehingga total citra stroke berjumlah 3219 data, kemudian sebanyak 3322 citra normal pada folder training dan 783 citra normal pada folder testing sehingga total citra normal berjumlah 4105. Terdapat tiga skenario pembagian data yang akan diterapkan pada penelitian ini, yaitu 70:30, 80:20, dan 90:10. Pada rasio 70:30, sebanyak 5126 data (70%) digunakan untuk pelatihan dan sebanyak 2198 data (30%) digunakan untuk pengujian. Untuk rasio 80:20, sebanyak 5859 data (80%) digunakan sebagai data latih dan sebanyak 1465 data (20%) digunakan sebagai data uji. Pada rasio 90:10, sebanyak 6591 data (90%) digunakan untuk pelatihan dan sebanyak 733 data (10%) digunakan untuk pengujian. Penggunaan skenario dengan pembagian

data ini diimplementasikan untuk menjaga proporsi yang seimbang ketika sistem digunakan yang berpengaruh terhadap keefektifan performa model yang dirancang dan dikembangkan.

Selanjutnya, perlu dilakukan inisialisasi terhadap nilai random state guna mengacak urutan pembagian dataset tersebut. Nilai random state pada penelitian ini dapat diinisialisasi dengan nilai sebesar 0 ataupun 42. Adanya nilai random state ini akan memberikan kekonsistenan terhadap pembagian data tiap kali menjalankan program. Konsistensi terhadap pembagian data juga akan mempengaruhi konsistensi terhadap hasil evaluasi model sehingga dalam pembagian data tersebut tidak terjadi karena adanya faktor kebetulan. Tabel 4.1 menunjukkan detail pembagian data latih dan data uji untuk ketiga variasi rasio pembagian yang digunakan dalam pengujian.

Tabel 4.1 Rasio pembagian data penelitian

Rasio	Jumlah Data	
	Training	Testing
70 : 30	5126	2198
80 : 20	5859	1465
90 : 10	6591	733

Pada tahap selanjutnya, data latih akan digunakan untuk melatih model deteksi, dan data uji akan digunakan untuk mengevaluasi performa model. Berbagai matriks klasifikasi akan dihitung untuk menilai performa model dan menentukan rasio pembagian data terbaik dan model klasifikasi yang paling optimal.

4.2 Hasil Uji Coba Jarak

Setelah melalui proses pengambilan nilai ekstraksi fitur, data akan digunakan untuk menentukan variasi jarak terbaik dari jarak 1 dan 2. Pada uji coba ini perbandingan tiga rasio data training dan testing tetap diberlakukan yaitu dengan perbandingan 70:30, 80:20, dan 90:10.

Percobaan pertama dimulai dengan penerapan jarak 1 dengan perbandingan tiga rasio data serta klasifikasi dengan *Multi Layer Perceptron*. Hasil dari *confusion matrix* percobaan tersebut ditampilkan pada Tabel 4.2.

Tabel 4.2 Hasil confusion matrix dengan jarak 1

Rasio	TP	TN	FP	FN
70:30	457	1023	199	519
80:20	321	689	113	342
90:10	159	340	54	180

Berdasarkan Tabel 4.2 diperoleh nilai *precision*, *recall*, dan *f1-score* yang tertera pada Tabel 4.3.

Tabel 4.3 Nilai precision, recall, dan f1-score dengan jarak 1

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	67,33%	69,66%	46,82%	56,00%
80:20	68,94%	73,96%	48,42%	58,52%
90:10	68,08%	74,65%	46,90%	57,61%

Pada percobaan selanjutnya menerapkan jarak 2 dengan perbandingan menggunakan tiga rasio serta penerapan *Multi Layer Perceptron* untuk klasifikasi, maka diperoleh hasil *confusion matrix* yang dapat dilihat pada Tabel 4.4.

Tabel 4.4 Hasil confusion matrix dengan jarak 2

Rasio	TP	TN	FP	FN
70:30	434	1046	176	542
80:20	294	695	107	369
90:10	155	342	52	184

Berdasarkan Tabel 4.4 diperoleh nilai *precision*, *recall*, dan *f1-score* yang tertera pada Tabel 4.5.

Tabel 4.5 Nilai *precision*, *recall*, dan *f1-score* dengan jarak 2

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	67,33%	71,15%	44,47%	54,73%
80:20	67,51%	73,32%	44,34%	55,26%
90:10	67,80%	74,88%	45,72%	56,78%

Percobaan terakhir yaitu menggunakan kombinasi jarak 1 dan 2 dan perbandingan tiga rasio serta penerapan *Multi Layer Perceptron* untuk klasifikasi, maka hasil *confusion matrix* yang diperoleh dapat dilihat pada Tabel 4.6.

Tabel 4.6 Hasil *confusion matrix* dengan kombinasi jarak 1 dan jarak 2

Rasio	TP	TN	FP	FN
70:30	517	1036	186	459
80:20	330	704	98	333
90:10	186	331	63	153

Berdasarkan Tabel 4.6 maka untuk hasil dari *precision*, *recall*, dan *f1-score* dapat dilihat pada Tabel 4.7.

Tabel 4.7 Nilai *precision*, *recall*, dan *f1-score* dengan jarak 1 dan 2

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	70,66%	73,54%	52,97%	61,58%
80:20	70,58%	77,10%	49,77%	60,49%
90:10	70,53%	74,70%	54,87%	63,27%

Hasil pengujian yang dilakukan terhadap variasi pemilihan jarak untuk ekstraksi fitur GLCM yang berbeda dengan tiga rasio pembagian data training dan testing dapat dilihat pada Tabel 4.8.

Tabel 4.8 Hasil confusion matrix uji coba jarak

Jarak	Rasio	Akurasi	Presisi	Recall	F1-score
1	70:30	67,33%	69,66%	46,82%	56,00%
	80:20	68,94%	73,96%	48,42%	58,52%
	90:10	68,08%	74,65%	46,90%	57,61%
2	70:30	67,33%	71,15%	44,47%	54,73%
	80:20	67,51%	73,32%	44,34%	55,26%
	90:10	67,80%	74,88%	45,72%	56,78%
1 dan 2	70:30	70,66%	73,54%	52,97%	61,58%
	80:20	70,58%	77,10%	49,77%	60,49%
	90:10	70,53%	74,70%	54,87%	63,27%

Hasil evaluasi pada model dengan variasi jarak didapatkan bahwa penerapan dengan kombinasi jarak 1 dan 2 memberikan hasil yang baik diantara penerapan jarak lainnya dengan akurasi terbaiknya sebesar 70,66% untuk rasio data training dan testing 70:30. Penggunaan jarak 1 menangkap hubungan lokal dengan jarak lebih pendek sedangkan pemilihan jarak 2 menangkap hubungan yang lebih global sehingga area cakupan lebih besar. Kombinasi dari kedua jarak tersebut akan memberikan opsi yang lebih kaya dan beragam terhadap pengenalan pola tekstur sehingga dapat membantu model MLP menangkap pola yang lebih kompleks dalam data.

4.3 Hasil Uji Coba Jumlah Hidden Layer

Data yang telah melalui uji coba sebelumnya akan dilanjutkan pada uji coba ini dengan skenario berbeda yang dipengaruhi oleh jumlah hidden layer dari 4 skenario jumlah hidden layer yaitu satu hidden layer, dua hidden layer, dan tiga hidden layer. Uji coba dengan skenario tersebut melibatkan *Multi Layer Perceptron* (MLP) dan GLCM sekaligus untuk mengevaluasi kinerja modelnya. Pada tiap skenario tersebut juga diberlakukan berbagai rasio pembagian data.

Hasil akurasi yang baik akan menentukan penggunaan jumlah hidden layer pada pengujian selanjutnya.

Pengujian pertama dilakukan dengan menerapkan tiga layer yang terdiri dari input layer, satu hidden layer, dan output layer. Hasil *confusion matrix* dari pengujian ini ditampilkan pada Tabel 4.9. *Confusion matrix* tersebut menginterpretasikan jumlah prediksi benar maupun yang salah. Dengan adanya hal tersebut maka dapat dilakukan evaluasi terhadap performa model untuk mengklasifikasikan data dengan jumlah hidden layer tertentu.

Tabel 4.9 Hasil confusion matrix dengan satu hidden layer

Rasio	TP	TN	FP	FN
70:30	517	1036	186	459
80:20	330	704	98	333
90:10	186	331	63	153

Berdasarkan Tabel 4.9, dapat dibuat Tabel 4.10 yang memberikan nilai *precision* yaitu proporsi untuk memprediksi data positif yang benar, *recall* yaitu proporsi untuk kejadian yang sebenarnya dan terdeteksi oleh model, serta *f1-score* yaitu untuk mengetahui keseimbangan antara *precision* dan *recall*.

Tabel 4.10 Nilai precision, recall, dan f1-score dengan satu hidden layer

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	70,66%	73,54%	52,97%	61,58%
80:20	70,58%	77,10%	49,77%	60,49%
90:10	70,53%	74,70%	54,87%	63,27%

Pengujian selanjutnya yaitu menggunakan empat layer yang terdiri dari input layer, dua hidden layer, dan output layer. Berikut hasil *confusion matrix* disajikan pada Tabel 4.11.

Tabel 4.11 Hasil confusion matrix dengan dua hidden layer

Rasio	TP	TN	FP	FN
70:30	547	1044	178	429
80:20	428	625	177	235
90:10	200	336	58	139

Berdasarkan Tabel 4.11 maka didapatkan hasil nilai untuk *precision*, *recall*, dan *f1-score* yang tersaji pada Tabel 4.12.

Tabel 4.12 Nilai precision, recall, dan f1-score dengan dua hidden layer

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	72,38%	75,45%	56,05%	64,32%
80:20	71,88%	70,74%	64,56%	67,51%
90:10	73,12%	77,52%	59,00%	67,00%

Pengujian terakhir dilakukan dengan menerapkan lima layer yang terdiri dari input layer, tiga hidden layer, dan output layer. Kemudian didapatkan hasil *confusion matrix* pada Tabel 4.13.

Tabel 4.13 Hasil confusion matrix dengan tiga hidden layer

Rasio	TP	TN	FP	FN
70:30	650	1048	174	326
80:20	475	655	147	188
90:10	244	341	53	95

Berdasarkan Tabel 4.13 maka menghasilkan nilai *precision*, *recall*, serta *f1-score* yang disajikan pada Tabel 4.14 berikut.

Tabel 4.14 Nilai precision, recall, dan f1-score dengan tiga hidden layer

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	77,25%	78,88%	66,60%	72,22%
80:20	77,13%	76,37%	71,64%	73,93%
90:10	79,81%	82,15%	71,98%	76,73%

Hasil pengujian yang dilakukan terhadap variasi jumlah hidden layer yang berbeda dengan tiga rasio pembagian data training dan testing dapat dilihat pada Tabel 4.15.

Tabel 4.15 Hasil confusion matrix uji coba jumlah hidden layer

Jumlah Hidden Layer	Rasio	Akurasi	Presisi	Recall	F1-score
1 hidden layer	70:30	70,66%	73,54%	52,97%	61,58%
	80:20	70,58%	77,10%	49,77%	60,49%
	90:10	70,53%	74,70%	54,87%	63,27%
2 hidden layer	70:30	72,38%	75,45%	56,05%	64,32%
	80:20	71,88%	70,74%	64,56%	67,51%
	90:10	73,12%	77,52%	59,00%	67,00%
3 hidden layer	70:30	77,25%	78,88%	66,60%	72,22%
	80:20	77,13%	76,37%	71,64%	73,93%
	90:10	79,81%	82,15%	71,98%	76,73%

Dari hasil yang didapatkan, dapat diketahui bahwa dengan jumlah tiga hidden layer, nilai akurasi, precision, dan recall yang didapatkan cukup baik yaitu 79,81% khususnya untuk pembagian data 90:10. Hal tersebut menunjukkan bahwa arsitektur *Multi Layer Perceptron* dengan penerapan tiga hidden layer bekerja dengan lebih baik dalam mengklasifikasikan data dibandingkan dengan penerapan satu maupun dua hidden layer, sehingga untuk keperluan pengujian berikutnya akan digunakan data dengan jumlah hidden layer sebanyak tiga.

4.4 Hasil Uji Coba Jumlah Neuron

Selanjutnya data yang telah melalui uji coba sebelumnya akan dilanjutkan pada uji coba ini dengan skenario berbeda yang dipengaruhi oleh jumlah neuron terbaik dari 4 skenario jumlah neuron pada hidden layer yang ada yaitu 32, 64, 128, dan 256. Uji coba dengan skenario tersebut melibatkan *Multi Layer Perceptron* (MLP) dan

GLCM sekaligus untuk mengevaluasi kinerja modelnya. Pada tiap skenario tersebut juga diberlakukan berbagai rasio pembagian data. Hasil akurasi yang baik akan menentukan penggunaan jumlah neuron pada pengujian selanjutnya.

Pengujian pertama dilakukan dengan menerapkan 32 neuron pada tiga hidden layer. Hasil *confusion matrix* dari pengujian ini ditampilkan pada Tabel 4.16. *Confusion matrix* tersebut menginterpretasikan jumlah prediksi benar maupun yang salah. Dengan adanya hal tersebut maka dapat dilakukan evaluasi terhadap performa model untuk mengklasifikasikan data dengan jumlah neuron tertentu.

Tabel 4.16 Hasil confusion matrix dengan 32 neuron

Rasio	TP	TN	FP	FN
70:30	650	1048	174	326
80:20	475	655	147	188
90:10	244	341	53	95

Berdasarkan Tabel 4.16, dapat dibuat Tabel 4.17 yang memberikan nilai *precision* yaitu proporsi untuk memprediksi data positif yang benar, *recall* yaitu proporsi untuk kejadian yang sebenarnya dan terdeteksi oleh model, serta *f1-score* yaitu untuk mengetahui keseimbangan antara *precision* dan *recall*.

Tabel 4.17 Nilai precision, recall, dan f1-score dengan 32 neuron

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	77,25%	78,88%	66,60%	72,22%
80:20	77,13%	76,37%	71,64%	73,93%
90:10	79,81%	82,15%	71,98%	76,73%

Pengujian selanjutnya yaitu menggunakan jumlah neuron sebanyak 64 pada masing-masing hidden layer dan hasil *confusion matrix* disajikan pada Tabel 4.18.

Tabel 4.18 Hasil confusion matrix dengan 64 neuron

Rasio	TP	TN	FP	FN
70:30	777	1063	159	199
80:20	565	682	120	98
90:10	261	336	58	78

Berdasarkan Tabel 4.18 maka didapatkan hasil nilai untuk *precision*, *recall*, dan *f1-score* yang tersaji pada Tabel 4.19.

Tabel 4.19 Nilai precision, recall, dan f1-score dengan 64 neuron

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	83,71%	83,01%	79,61%	81,28%
80:20	85,12%	82,48%	85,22%	83,83%
90:10	81,45%	81,82%	76,99%	79,33%

Selanjutnya dilakukan pengujian dengan jumlah 128 neuron pada tiga hidden layer yang digunakan dan didapatkan hasil *confusion matrix* pada Tabel 4.20.

Tabel 4.20 Hasil confusion matrix dengan 128 neuron

Rasio	TP	TN	FP	FN
70:30	855	1015	207	121
80:20	584	697	105	79
90:10	296	345	49	43

Berdasarkan Tabel 4.20 maka menghasilkan nilai *precision*, *recall*, serta *f1-score* yang disajikan pada Tabel 4.21 berikut.

Tabel 4.21 Nilai precision, recall, dan f1-score dengan 128 neuron

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	85,08%	80,51%	87,60%	83,91%
80:20	87,44%	84,76%	88,08%	86,39%
90:10	87,45%	85,80%	87,32%	86,55%

Pengujian terakhir dengan jumlah 256 neuron pada tiga hidden layer yang digunakan dan didapatkan hasil *confusion matrix* pada Tabel 4.22.

Tabel 4.22 Hasil confusion matrix dengan 256 neuron

Rasio	TP	TN	FP	FN
70:30	811	1097	149	141
80:20	600	732	70	63
90:10	309	359	35	30

Berdasarkan Tabel 4.22 maka menghasilkan nilai *precision*, *recall*, serta *f1-score* yang disajikan pada Tabel 4.23 berikut.

Tabel 4.23 Nilai precision, recall, dan f1-score dengan 256 neuron

Rasio	Akurasi	Presisi	Recall	F1-score
70:30	86,81%	84,48%	85,19%	84,83%
80:20	90,92%	89,55%	90,50%	90,02%
90:10	91,13%	89,83%	91,15%	90,48%

Hasil pengujian yang dilakukan terhadap variasi jumlah neuron yang berbeda dengan tiga rasio pembagian data training dan testing dapat dilihat pada Tabel 4.24.

Tabel 4. 24 Hasil confusion matrix uji coba jumlah neuron

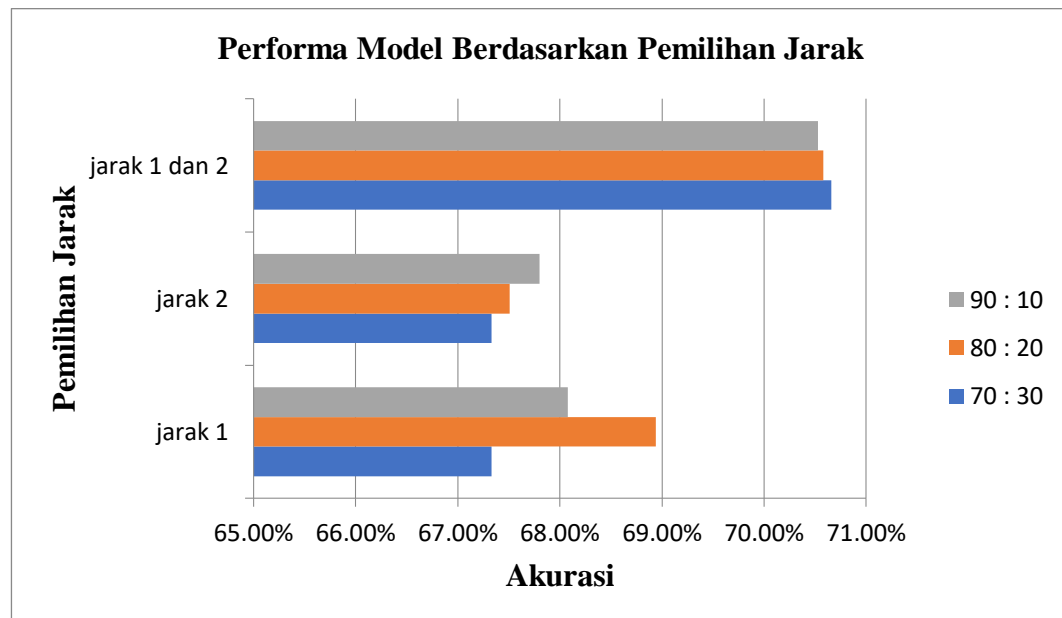
Jumlah Neuron	Rasio	Accuracy	Precision	Recall	F1-score
32	70:30	77,25%	78,88%	66,60%	72,22%
	80:20	77,13%	76,37%	71,64%	73,93%
	90:10	79,81%	82,15%	71,98%	76,73%
64	70:30	83,71%	83,01%	79,61%	81,28%
	80:20	85,12%	82,48%	85,22%	83,83%
	90:10	81,45%	81,82%	76,99%	79,33%
128	70:30	85,08%	80,51%	87,60%	83,91%
	80:20	87,44%	84,76%	88,08%	86,39%
	90:10	87,45%	85,80%	87,32%	86,55%
256	70:30	86,81%	84,48%	85,19%	84,83%
	80:20	90,92%	89,55%	90,50%	90,02%
	90:10	91,13%	89,83%	91,15%	90,48%

Dari hasil yang didapatkan, dapat diketahui bahwa pada jumlah neuron sebanyak 256, nilai akurasi, *precision*, dan *recall* yang didapatkan cukup baik untuk

pembagian data 90:10. Hal tersebut menunjukkan bahwa arsitektur *Multi Layer Perceptron* dengan penerapan 256 neuron pada tiga *hidden layer* bekerja dengan lebih baik dalam mengklasifikasikan data dibandingkan dengan penerapan jumlah neuron lainnya dengan perolehan akurasi mencapai 91,13% pada rasio data training dan testing yaitu 90:10.

4.5 Pembahasan

Hasil eksperimen yang disajikan menunjukkan variasi signifikan dalam performa model *Multi Layer Perceptron* untuk deteksi stroke berdasarkan citra CT scan otak, tergantung pada pemilihan jumlah neuron pada *hidden layer* arsitekturnya serta jarak dalam ekstraksi fitur GLCM.

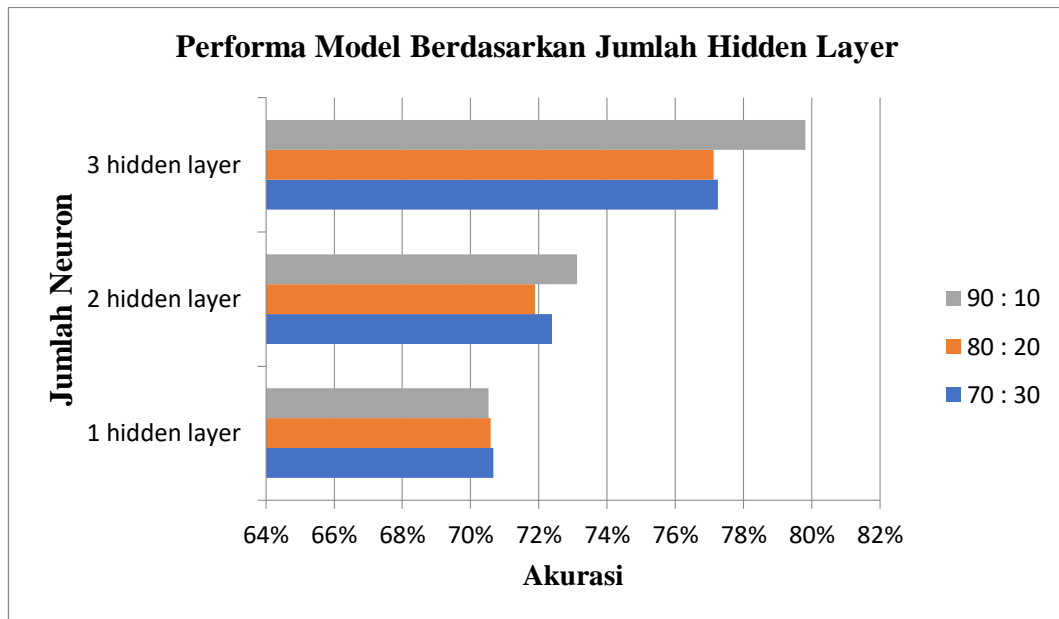


Gambar 4.3 Hasil uji coba dengan variasi jarak

Dari hasil uji coba dengan penerapan jarak yang ditunjukkan pada Gambar 4.11, dapat dilihat bahwa dengan meningkatkan jarak dari 1 menjadi kombinasi 1 dan 2 maka terjadi peningkatan performa dengan rasio pembagian data 70:30, model

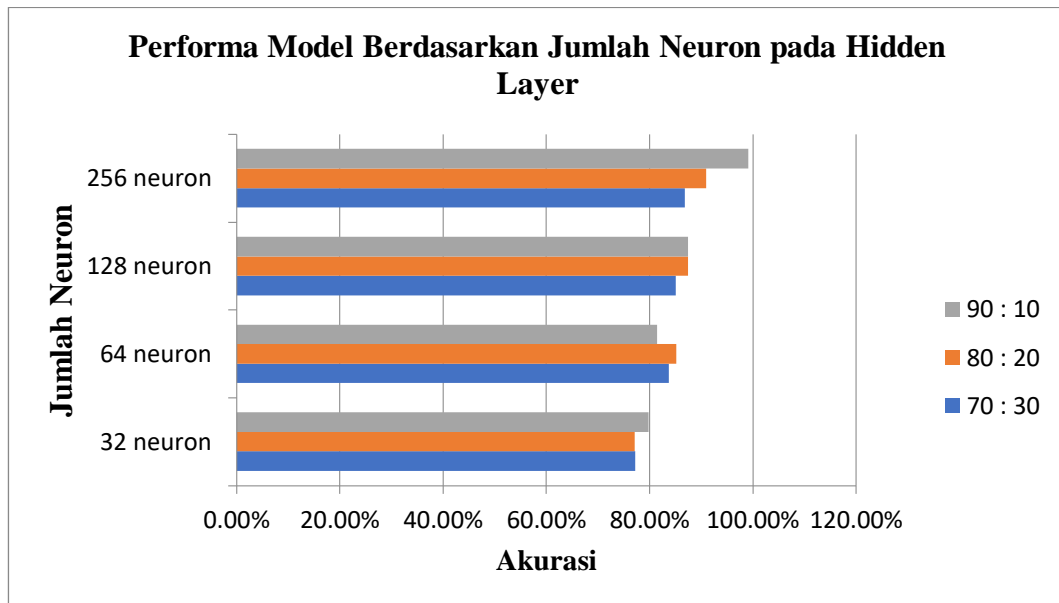
mencapai performa puncak dengan akurasi sebesar 70,66%, presisi 73,54%, *recall* 52,97%, dan *f1-score* 61,58%.

Peningkatan ini dapat disebabkan oleh faktor tertentu. GLCM dengan jarak yang berbeda menangkap karakteristik tekstur yang unik. Jarak 1 menangkap hubungan lokal (tekstur yang lebih halus dan detail). Jarak 2 menangkap hubungan yang lebih global (tekstur yang lebih kasar atau pola lebih besar). Kombinasi jarak 1 dan 2 menghasilkan fitur yang lebih kaya dan beragam, sehingga membantu model MLP memahami pola yang lebih kompleks dalam data. Model MLP bekerja baik jika diberikan fitur yang representative seperti kombinasi jarak 1 dan 2 yang dapat memperkaya fitur input. Fitur dari jarak 1 dan 2 saling melengkapi tanpa terlalu banyak redundansi. Kombinasi fitur tersebut memberikan representasi data yang lebih kaya, tanpa noise yang berlebihan. Tekstur yang berbeda dalam citra medis dapat lebih baik direpresentasikan dengan jarak kombinasi, terutama jika pola tekstur terjadi pada skala yang berbeda. Selain itu, rasio 90:10 memberikan lebih banyak data untuk pelatihan, yang mendukung model dalam mempelajari pola fitur GLCM yang lebih kompleks.



Gambar 4.4 Hasil uji coba dengan variasi jumlah hidden layer

Pengujian dengan melibatkan variasi jumlah *hidden layer* seperti yang ditunjukkan pada Gambar 4.12 memberikan perubahan dan peningkatan performa secara umum dengan peningkatan dari 1 *hidden layer*, 2 *hidden layer*, dan 3 *hidden layer* maka performa model mengalami peningkatan ketika diterapkan sebanyak 3 *hidden layer* pada arsitektur modelnya. Jumlah *hidden layer* memberikan bantuan dalam meningkatkan akurasi terutama pada rasio pembagian data 90:10. Pada rasio pembagian data tersebut model mampu mencapai performa dengan akurasi sebesar 79,81%, presisi 82,15%, *recall* 71,98%, dan *f1-score* 76,73%.



Gambar 4.5 Hasil uji coba dengan variasi jumlah neuron

Pengujian dengan melibatkan beberapa variasi jumlah neuron yang terdapat pada 3 hidden layer seperti ditunjukkan pada Gambar 4.13 mampu memberikan perubahan dan peningkatan performa secara umum dengan peningkatan dari sejumlah 32 neuron, 64 neuron, 128 neuron, hingga 256 neuron maka performa model mengalami peningkatan ketika diterapkan sebanyak 256 neuron kemudian jumlah *hidden layer* juga memberikan bantuan dalam meningkatkan akurasi dengan rasio pembagian data 90:10. Pada rasio pembagian data tersebut model mampu mencapai akurasi sebesar 91,13%, presisi 89,83%, *recall* 91,15%, dan *f1-score* 90,48%.

Akurasi yang cukup baik yang didapatkan dengan pemilihan 256 neuron dipengaruhi oleh beberapa faktor terkait kapasitas model, kompleksitas data hingga bagaimana cara MLP bekerja. Jumlah neuron yang lebih banyak (256) memberikan kapasitas model yang lebih tinggi untuk mempelajari dan memahami pola yang lebih kompleks dalam data. Dengan 256 neuron, model MLP dapat

menghindari underfitting dan menangkap hubungan non-linear yang mungkin ada di dalam data, yang meningkatkan akurasi model. Jika model terlalu kecil (misalnya 32 atau 64 neuron), model mungkin tidak mampu menangkap kompleksitas dalam data, sehingga menghasilkan performa yang lebih buruk.

Secara keseluruhan, hasil uji coba menunjukkan bahwa pemilihan jarak GLCM dengan kombinasi jarak 1 dan 2 dalam GLCM menghasilkan fitur tekstur yang lebih kaya dan lebih beragam, yang memungkinkan model untuk menangkap pola lokal dan global dalam data. Kombinasi ini membantu MLP dalam memahami hubungan tekstur yang kompleks dan meningkatkan representasi data. Kemudian berdasarkan pemilihan jumlah hidden layer dan jumlah neuron dimana dengan jumlah 256 neuron pada masing-masing 3 hidden layer memberikan kapasitas model yang lebih besar, memungkinkan model untuk mempelajari pola data yang lebih kompleks dan non-linear. Jumlah neuron yang lebih banyak mengurangi risiko underfitting, di mana model terlalu sederhana dan gagal menangkap pola penting dalam data. Model dengan 256 neuron mampu memanfaatkan lebih banyak parameter untuk mengoptimalkan bobot dan menggeneralisasi data dengan lebih baik, menghasilkan akurasi yang lebih tinggi. Akurasi yang cukup baik juga hampir terjadi di seluruh pengujian dengan rasio 90:10 dikarenakan 90% data latih memberikan model lebih banyak informasi untuk belajar, yang mendukung generalisasi model terhadap data yang lebih beragam. Data uji 10% menghasilkan akurasi yang lebih stabil, meskipun ukuran data uji kecil, karena kesalahan estimasi lebih rendah.

Penelitian ini mendeteksi adanya kelainan otak yang dapat menyebabkan munculnya penyakit stroke melalui citra CT scan. Dengan mengembangkan metode atau algoritma yang canggih, penelitian ini memberikan dukungan yang lebih baik kepada pihak terkait yang berkaitan langsung dengan bidang medis dalam diagnosis awal penyakit stroke. Selain itu, penelitian ini juga untuk mengurangi tingkat kesalahan diagnosis yang mungkin terjadi dan meningkatkan tingkat deteksi penyakit stroke, sehingga dapat meningkatkan kesempatan penyembuhan pasien. Penelitian ini diharapkan bisa membawa banyak manfaat terutama bagi daerah-daerah dengan akses terbatas terhadap layanan medis yang canggih. Secara keseluruhan, penelitian ini berkontribusi pada pengembangan teknologi medis, meningkatkan kualitas layanan kesehatan, meningkatkan kesadaran masyarakat, dan membantu dalam pencegahan serta pengendalian penyakit stroke yang masih menjadi penyakit mematikan penyebab kematian secara global. Hal tersebut termasuk dalam langkah berikhtiar untuk meraih atau mendapatkan hal yang baik, tidak putus asa terhadap ketentuan yang diberikan oleh Allah Subhanahu Wa ta'ala.

Sebagaimana firman Allah Subhanahu Wa ta'ala dalam Q.S An-Najm ayat 39 yang berbunyi:

وَأَنْ يَّسْأَلَ الْإِنْسَانَ إِلَّا مَا سَعَىٰ

“dan bahwasanya seorang manusia tiada memperoleh selain apa yang telah diusahakannya,” (Q.S An-Najm (53):39).

وَأَنْ سَعَيْهِ سَوْفَ يُرَىٰ

“dan bahwasanya usaha itu kelak akan diperlihatkan (kepadanya).” (Q.S An-Najm (53):40).

Menurut Tafsir al-Muyassar bahwasanya usaha seseorang akan diperlihatkan di akhirat. Yang baik akan dibedakan dari yang buruk untuk memuliakan orang yang berbuat baik dan merendahkan orang yang berbuat buruk. Tafsir lain juga menjelaskan bahwa amalnya pasti akan diperlihatkan secara kasat mata pada hari Kiamat (Tafsir al-Mukhtashar dan Zubdatut).

Kesungguhan dalam berihitiar pasti akan membuahkan hasil, karena setiap manusia akan memperoleh apa yang telah diusahakannya. Sebagaimana sabda Nabi Muhammad Saw. dalam sebuah hadits berikut.

وَمَنْ يَسْتَعْفِفْ يُعِفَّهُ اللَّهُ وَمَنْ يَسْتَغْنِ يُغْنِهِ اللَّهُ

“Siapa yang berusaha memelihara dirinya, Allah Swt. akan memeliharanya dan siapa yang berusaha mencukupkan dirinya maka Allah Swt. akan mencukupkannya”-(H.R. al-Bukhari dari Musa bin Ismail).

Dalam penelitian ini, mengingatkan pada prinsip bahwa teknologi seperti GLCM untuk analisis tekstur dan *Multi Layer Perceptron* untuk klasifikasi adalah alat yang membantu dalam deteksi, namun keberhasilan pengobatan akhirnya tunduk pada kehendak Ilahi. Hal ini memberikan perspektif bahwa sains dan keimanan dapat berjalan bersama dalam upaya kemanusiaan untuk melawan penyakit, dengan mengakui peran akhir dari kekuasaan Allah dalam penyembuhan. Sebagaimana juga yang disebutkan pada salah satu hadits yaitu:

لِكُلِّ دَاءٍ دَوَاءٌ، فَإِذَا أُصِيبَ دَوَاءُ الدَّاءِ بَرَأَ بِإِذْنِ اللَّهِ

“Setiap penyakit ada obatnya. Apabila obat itu tepat untuk suatu penyakit, penyakit itu akan sembuh dengan seizin Allah.”-(HR. Bukhari).

Hadits tersebut menjelaskan bahwa setiap penyakit yang diciptakan memiliki obatnya, yang merupakan manifestasi dari kebijaksanaan dan rahmat Allah. Hadist pertama dari Bukhari mengungkapkan bahwa untuk setiap penyakit, ada obat yang sesuai, mendorong manusia terutama sebagai umat Islam untuk berikhtiar dalam mencari solusi dan pengobatan atas penyakit dengan keyakinan bahwa kesembuhan adalah dengan izin Allah.

Dengan adanya hasil implementasi dan pengujian pada penelitian ini diharapkan bisa memberikan kebermanfaatan bagi banyak pihak khususnya bagi para pakar medis dalam meningkatkan efektivitas deteksi penyakit stroke berbasis citra CT scan otak untuk mendapatkan penanganan yang cepat dan akurat, serta bagi peneliti di bidang kedokteran dan ilmu terkait yang membutuhkan informasi terbaru terkait pengembangan metode deteksi penyakit stroke.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil uji coba yang telah dilakukan, didapatkan hasil bahwa penerapan *Gray Level Co-occurrence Matrix* (GLCM) sebagai metode ekstraksi fitur dan *Multi Layer Perceptron* (MLP) sebagai model klasifikasi menunjukkan performa yang efektif dalam mendeteksi penyakit stroke melalui citra CT scan otak. Pengujian parameter jarak pada GLCM menunjukkan bahwa kombinasi jarak 1 dan 2 memberikan akurasi terbaik sebesar 70,66%, dengan presisi 74%, *recall* 53%, dan *f1-score* 62%. Sementara itu, analisis jumlah hidden layer pada MLP menunjukkan bahwa model dengan tiga hidden layer memberikan akurasi terbaik sebesar 79,81%, presisi 82%, *recall* 72%, dan *f1-score* 77% serta penerapan jumlah neuron pada hidden layer mencapai performa terbaik dengan akurasi sebesar 91,13%, presisi 89,83%, *recall* 91,15%, dan *f1-score* 90,48% dengan jumlah 256 neuron. Hal ini menegaskan bahwa variasi jarak pada GLCM serta optimasi jumlah hidden layer dan neuron pada MLP merupakan faktor penting dalam meningkatkan kualitas deteksi stroke.

5.2 Saran

Dalam pengembangan sistem deteksi stroke berbasis citra CT scan otak memerlukan beberapa penyempurnaan untuk meningkatkan akurasi dan kinerja deteksi. Berdasarkan hasil penelitian ini, diharapkan pengembangan lebih lanjut dapat menghasilkan sistem yang lebih optimal. Berikut saran yang dapat penulis berikan.

1. Pemilihan dan pengolahan dataset, khususnya pada tahap *preprocessing*, perlu diperhatikan kembali. Ukuran *resizing* yang lebih besar disarankan untuk mempertahankan detail citra, sehingga model dapat menangkap fitur-fitur halus yang penting, terutama dalam mengenali tekstur pada citra medis.
2. Penggunaan algoritma deteksi selain *Multi Layer Perceptron*, seperti *Random Forest*, *Decision Tree*, *Convolutional Neural Network*, atau *Support Vector Machine*, perlu dipertimbangkan untuk mendapatkan performa yang lebih optimal.
3. Pengembangan antarmuka berbasis web atau platform lain direkomendasikan untuk memudahkan pengguna dalam mengunggah citra CT scan dan mendapatkan hasil deteksi secara cepat, sehingga dapat mendukung diagnosis awal secara lebih efisien.

DAFTAR PUSTAKA

- Aini, D. H. N., Kurniasari, D., Nuryaman, A., & Usman, M. (2023). Implementation of Artificial Neural Network With Backpropagation Algorithm for Rating Classification on Sales of Blackmores in Tokopedia. *Jurnal Teknik Informatika (Jutif)*, 4(2), 365–372. <https://doi.org/10.52436/1.jutif.2023.4.2.539>
- Amraeni, Y. (2021). Issu Kesehatan Masyarakat dalam SDG's. Pekalongan: Nasya Expanding Management.
- Cecep Abdul Cholik. (2021). Teknologi Informasi, ICT,. *Jurnal Fakultas Teknik*, 2(2), 39–46.
- Chakraborty, S., Aich, S., & Kim, H.-C. (2020). 3D Textural, Morphological and Statistical Analysis of Voxel of Interests in 3T MRI Scans for the Detection of Parkinson's Disease Using Artificial Neural Networks. *Healthcare*, 8(1), 34. <https://doi.org/10.3390/healthcare8010034>
- Euis Saraswati, Yuyun Umaidah, & Apriade Voutama. (2021). Penerapan Algoritma Artificial Neural Network untuk Klasifikasi Opini Publik Terhadap Covid-19. *Generation Journal*, 5(2), 109–118. <https://doi.org/10.29407/gj.v5i2.16125>
- Ewees, A. A., Elaziz, M. A., Alameer, Z., Ye, H., & Jianhua, Z. (2020). Improving multilayer perceptron neural network using chaotic grasshopper optimization algorithm to forecast iron ore price volatility. *Resources Policy*, 65(November 2019), 101555. <https://doi.org/10.1016/j.resourpol.2019.101555>
- Geneva, R. (2023). Gambaran Karakteristik Individu dengan Kejadian Stroke pada Pasien Poliklinik Penyakit Saraf. *Jurnal Kedokteran STM (Sains dan Teknologi Medik)*, VI(II), 159–167. <https://jurnal.fk.uisu.ac.id/index.php/stm>
- Hokkinen, L., Mäkelä, T., Savolainen, S., & Kangasniemi, M. (2021). Evaluation of a CTA-based convolutional neural network for infarct volume prediction in anterior cerebral circulation ischaemic stroke. *European Radiology Experimental*, 5(1), 1–11. <https://doi.org/10.1186/s41747-021-00225-1>
- Hussain, A., & Khunteta, A. (2020). Semantic Segmentation of Brain Tumor from MRI Images and SVM Classification using GLCM Features. *Proceedings of the 2nd International Conference on Inventive Research in Computing Applications, ICIRCA 2020*, 38–43. <https://doi.org/10.1109/ICIRCA48905.2020.9183385>
- Kasdi, A. (2014). Maqashid Syari'ah Dan Hak Asasi Manusia (Implemetasi Ham Dalam Pemikiran Islam). *Jurnal Penelitian*, 8(2), 247–268. <https://journal.iainkudus.ac.id/index.php/jurnalPenelitian/article/view/836>

- Lingga, L. (2013). *All About Stroke Hidup Sebelum dan Pasca Stroke*. Jakarta: Elex Media Komputindo.
- Listia, R., Harjoko, A., Mada, U. G., & Utara, S. (2014). *Klasifikasi Massa pada Citra Mammogram Berdasarkan Gray Level Cooccurrence Matrix (GLCM)*. 8(1).
- Mataija, M., & Nemeč, D.S. (2021). *Brain Tumor Detection Based on MRI Images Using Multilayer Perceptron: Proceedings of The International Scientific Student Conference, University of Rijeka, 2021*.
- Nithya, R., & Santhi, B. (2011). Comparative study on feature extraction method for breast cancer classification. *Journal of Theoretical and Applied Information Technology*, 33(2), 220–226.
- Peng, C. C., Wang, S. H., Liu, S. J., Yang, Y. K., & Liao, B. H. (2020). Artificial Neural Network Application to the Stroke Prediction. *2nd IEEE Eurasia Conference on Biomedical Engineering, Healthcare and Sustainability 2020, ECBIOS 2020*, 130–133. <https://doi.org/10.1109/ECBIOS50299.2020.9203638>
- Pinzon, R. T. (2016). *Awas Stroke*. Yogyakarta: Betha Grafika Yogyakarta
- Rachman, R. (2021). Implementasi Case Based Reasoning Mendiagnosa Penyakit Stroke Menggunakan Algoritma Probabilistic Symmetric. *Jurnal Informatika*, 8(1), 10–16. <https://doi.org/10.31294/ji.v8i1.8563>
- Raden Roro Ayuni Purbo Okta Briliani, & Irma Palupi. (2022). Klasifikasi Penyakit Kulit menggunakan Image Processing dan Artificial Neural Network (ANN). *E-Proceeding of Engineering*, 9(3), 1902.
- Ramchoun, H., Amine, M., Idrissi, J., Ghanou, Y., & Ettaouil, M. (2016). Multilayer Perceptron: Architecture Optimization and Training. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4(1), 26. <https://doi.org/10.9781/ijimai.2016.415>
- Rezka, A. N., Hidayat, B., & Syawqie, A. (2017). Pengenalan Individu Berdasarkan Pola Rugee Palatina Menggunakan Histogram of Oriented Gradients Dan Multi Layer Perceptron. *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri*, 1–7.
- Sanubary, I. (2018). Brain Tumor Detection Using Backpropagation Neural Networks. *Indonesian Journal of Physics and Nuclear Applications*, 3(3), 83–88. <https://doi.org/10.24246/ijpna.v3i3.83-88>
- Sharma, R., Kim, M., & Gupta, A. (2022). Motor imagery classification in brain-machine interface with machine learning algorithms: Classical approach to multi-layer perceptron model. *Biomedical Signal Processing and Control*, 71(PA), 103101. <https://doi.org/10.1016/j.bspc.2021.103101>
- Situmorang, G. T., Widodo, A. W., & Rahman, M. A. (2019). Penerapan Metode

Gray Level Co-occurrence Matrix (GLCM) untuk ekstraksi ciri pada telapak tangan. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 3(5), 4710–4716.

Subudhi, A., Dash, M., & Sabut, S. (2019). ScienceDirect Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier. *Integrative Medicine Research*, 1–13. <https://doi.org/10.1016/j.bbe.2019.04.004>

Suyanto. (2022). *Machine Learning Tingkat Dasar dan Lanjut* (2nd ed.). Informatika Bandung.

Tazin, T., Alam, M. N., Dola, N. N., Bari, M. S., Bourouis, S., & Monirujjaman Khan, M. (2021). Stroke Disease Detection and Prediction Using Robust Learning Approaches. *Journal of Healthcare Engineering*, 2021. <https://doi.org/10.1155/2021/7633381>

Usha, R., & Perumal, K. (2019). SVM classification of brain images from MRI scans using morphological transformation and GLCM texture features. *International Journal of Computational Systems Engineering*, 5(1), 18. <https://doi.org/10.1504/ijcsyse.2019.098415>

Zuriati Z, & Qomariyah N. (2023). Klasifikasi Penyakit Stroke Menggunakan Algoritma K-Nearest Neighbor (KNN) INFORMASI ARTIKEL ABSTRAK Classification of Stroke Using the K-Nearest Neighbor (KNN) Algorithm. *ROUTERS: Jurnal Sistem Dan Teknologi Informasi*, 1(1), 1–8

LAMPIRAN

Lampiran I Contoh dataset ekstraksi fitur GLCM pada sudut 0 jarak 1

contrast_0_1	correlation_0_1	dissimilarity_0_1	homogeneity_0_1	ASM_0_1	entropy_0_1
1.241879921	0.831629362	0.424950787	0.865421339	0.447105739	2.628788942
1.310654528	0.860029666	0.435654528	0.8640573	0.463515604	2.561158407
1.330708661	0.843731485	0.396899606	0.884813739	0.529183705	2.26407709
0.813115157	0.916701038	0.308932087	0.895516454	0.527205463	2.2304561
0.771899606	0.913895844	0.302165354	0.896999569	0.542868759	2.175638189
0.723056102	0.80633419	0.156496063	0.96609003	0.8589501	0.700407305
0.943528543	0.853960171	0.342273622	0.882554427	0.46153694	2.410727168
0.933932087	0.85589184	0.341904528	0.882570685	0.459932942	2.41466583
0.927042323	0.855800333	0.341043307	0.882656147	0.4596464	2.416620818
0.931840551	0.854730301	0.345472441	0.880817339	0.460656736	2.422372605
0.927288386	0.855586025	0.34707185	0.879829792	0.460041749	2.430247708
0.918184055	0.856075185	0.346456693	0.879452642	0.459289068	2.428217301
0.916461614	0.856525794	0.344857283	0.880606721	0.457835585	2.43530246
1.388410433	0.835535081	0.403051181	0.885895245	0.535098482	2.229559998
0.944143701	0.851295669	0.354945866	0.877125955	0.457280079	2.442242094
0.941683071	0.851443401	0.351870079	0.878072053	0.455826507	2.439421689
0.935654528	0.852708999	0.350639764	0.878817345	0.454817158	2.449327887
0.744340551	0.823121537	0.162770669	0.963898238	0.845362605	0.751858658
0.914124016	0.855981245	0.345595472	0.88020555	0.453121246	2.450458512
0.898129921	0.858554371	0.341904528	0.880503536	0.452394461	2.442871656
0.900098425	0.857546032	0.342150591	0.880536783	0.452040429	2.440118441
0.901943898	0.857726	0.343134843	0.880183288	0.450388471	2.448463425
0.910063976	0.857117918	0.345964567	0.879765127	0.451164146	2.458049452
0.924704724	0.854074422	0.352977362	0.877656821	0.450365357	2.468276349
1.345103346	0.840706104	0.404035433	0.8828757	0.533712215	2.244982986
0.931102362	0.852824486	0.358021654	0.875652799	0.449934423	2.472220661
0.944389764	0.851167303	0.362450787	0.874320471	0.449267536	2.476070148
0.933932087	0.853666197	0.356422244	0.876271106	0.448575578	2.476165551
0.93257874	0.854175008	0.357406496	0.875678504	0.448879749	2.475041116
0.742125984	0.842263247	0.16117126	0.964897137	0.836741615	0.771285608
0.917937992	0.856298346	0.352485236	0.87703456	0.448187169	2.4653879
0.902312992	0.858025447	0.343626969	0.880699214	0.447964655	2.449924376
0.88976378	0.860458145	0.340551181	0.881186141	0.446628124	2.454875341
0.89554626	0.858177774	0.343011811	0.880686911	0.448476141	2.447350028
0.889271654	0.859135851	0.340551181	0.880809064	0.448219774	2.435765898
1.384227362	0.839802487	0.407603346	0.882914587	0.533700503	2.241183411
0.901697835	0.858722543	0.34461122	0.879431477	0.447452624	2.448481412
0.887795276	0.860283651	0.346825787	0.87812229	0.446457546	2.453809112
0.893823819	0.86033988	0.347563976	0.8785488	0.445958097	2.465186982
0.925442913	0.858549819	0.361589567	0.873596342	0.444505393	2.488582413
...
0.657726378	0.892383404	0.241879921	0.915016507	0.567331425	1.893880523
0.662401575	0.889988513	0.239788386	0.916957768	0.57268962	1.871154017
0.619709646	0.898545744	0.229822835	0.91934121	0.57839409	1.857704504
0.62167815	0.898059042	0.225393701	0.921261014	0.586766841	1.818957393

Lampiran II Contoh dataset ekstraksi fitur GLCM pada sudut 45 jarak 1

contrast_45_1	correlation_45_1	dissimilarity_45_1	homogeneity_45_1	ASM_45_1	entropy_45_1
1.486514973	0.79955682	0.481616963	0.854491983	0.439159381	2.702278945
1.73005146	0.816267607	0.519437039	0.850891695	0.449035798	2.668281861
1.71498543	0.799757176	0.468038936	0.874215182	0.518408769	2.343363749
1.227230454	0.875000258	0.391468783	0.882547998	0.515128115	2.338447555
1.196602393	0.867306711	0.388244776	0.882823973	0.529322916	2.29077897
0.926157852	0.753766746	0.189720379	0.961450832	0.850719937	0.731006617
1.231322463	0.81046241	0.388864778	0.877405826	0.453296278	2.46835378
1.220782442	0.812661161	0.384276769	0.879084406	0.452336089	2.465348895
1.209622419	0.812874653	0.385764772	0.878042139	0.452234371	2.467067418
1.217434435	0.811246554	0.391220782	0.875759059	0.452749451	2.475969312
1.213218426	0.812088319	0.391840784	0.87480839	0.452139627	2.481649115
1.210490421	0.811291374	0.388616777	0.876316248	0.451799034	2.473070827
1.218178436	0.810331799	0.389236778	0.876488963	0.449842653	2.486070243
1.799553599	0.788057339	0.480438961	0.873796386	0.520229567	2.318383806
1.241242482	0.805571057	0.397172794	0.874041286	0.449696267	2.49272444
1.244342489	0.80476874	0.398040796	0.873809819	0.449467298	2.487238063
1.234050468	0.806789975	0.3997768	0.872548043	0.446840746	2.505139231
0.961249922	0.773252389	0.196912394	0.9594469	0.836635683	0.784130402
1.203670407	0.811386188	0.391096782	0.875000738	0.444978375	2.502696078
1.188914378	0.813761743	0.384152768	0.877325743	0.444763634	2.493596611
1.187550375	0.813053107	0.388120776	0.875286234	0.443770871	2.493718221
1.187178374	0.813727043	0.386756774	0.875940557	0.443364667	2.496236154
1.204414409	0.811906022	0.394196788	0.873414125	0.443086022	2.511457302
1.217062434	0.808955307	0.404240808	0.869563253	0.44229584	2.521721675
1.791245582	0.789088467	0.483786968	0.87154401	0.520889781	2.330809163
1.215822432	0.808836635	0.402876806	0.870558724	0.442413549	2.52124558
1.248930498	0.804215734	0.407588815	0.869753535	0.441318757	2.526547398
1.226486453	0.808841039	0.400272801	0.872283804	0.440757603	2.527928464
1.236282473	0.807701269	0.406844814	0.869855466	0.440269982	2.533477227
0.946245892	0.800343218	0.19480439	0.960157835	0.82843972	0.803222702
1.220658441	0.809910014	0.402256805	0.87089692	0.43998008	2.525358852
1.208754418	0.810799777	0.396056792	0.873736452	0.439917928	2.510622814
1.184698369	0.815170137	0.391716783	0.873935517	0.438599856	2.512382159
1.193378387	0.812001058	0.391220782	0.874656933	0.440047275	2.504697964
1.195734391	0.811579389	0.39010478	0.87458423	0.439994005	2.492153867
1.779031558	0.795289245	0.479384959	0.872956063	0.519568879	2.326952235
1.221898444	0.809552147	0.397172794	0.873057477	0.438327521	2.512144228
1.203670407	0.81156299	0.398288797	0.871256669	0.437854186	2.51154936
1.213342427	0.811406715	0.399280799	0.871856668	0.436578342	2.528601993
1.249302499	0.810048536	0.409448819	0.869623925	0.435939453	2.550895006
...
0.90482981	0.850653561	0.276644553	0.913409151	0.564834595	1.929700261
0.848657697	0.861920686	0.266972534	0.914844527	0.57184346	1.910932477
0.860561721	0.859764233	0.261640523	0.917924383	0.578838278	1.877782595
0.861429723	0.859361515	0.261392523	0.918133117	0.587062507	1.8453126

Lampiran III Contoh dataset ekstraksi fitur GLCM pada sudut 90 jarak 1

contrast_90_1	correlation_90_1	dissimilarity_90_1	homogeneity_90_1	ASM_90_1	entropy_90_1
0.973917323	0.867958989	0.378813976	0.871536737	0.453981524	2.577609336
1.185531496	0.873392083	0.412893701	0.868352564	0.466480619	2.543737305
1.19863435	0.859219375	0.377276083	0.8867457	0.53483743	2.240031186
0.725393701	0.925687596	0.291338583	0.89894581	0.530430114	2.204812962
0.752706693	0.916036782	0.300688976	0.896046734	0.543205549	2.16914601
0.472810039	0.873360948	0.114419291	0.971811653	0.866928788	0.662145752
0.878813976	0.863976725	0.322711614	0.88788711	0.462300687	2.391481417
0.876722441	0.864719438	0.323449803	0.887508129	0.461315574	2.394660043
0.865649606	0.865349853	0.323818898	0.886630358	0.460929708	2.395878747
0.868110236	0.864665567	0.325172244	0.886110037	0.462190849	2.396520179
0.860728346	0.865951948	0.31988189	0.888591392	0.461962001	2.40029941
0.850024606	0.866759138	0.320374016	0.886977922	0.462093408	2.388831181
0.86515748	0.864557576	0.321850394	0.887334933	0.459487581	2.406570771
1.275036909	0.848940278	0.384042815	0.888754409	0.538797119	2.212052583
0.883366142	0.860868244	0.328740157	0.884895101	0.459598717	2.408201927
0.886318898	0.860177458	0.32898622	0.885018352	0.458348446	2.406977813
0.883981299	0.860843413	0.330708661	0.883835712	0.456619731	2.420094501
0.496432087	0.882032298	0.119832677	0.969972333	0.854070674	0.708984656
0.867249016	0.863366325	0.325295276	0.885804875	0.454820594	2.422480462
0.868356299	0.863243391	0.326033465	0.885689753	0.45240748	2.428295702
0.863927165	0.863270672	0.327386811	0.884862659	0.451846707	2.427443339
0.860851378	0.864207997	0.328001969	0.884368116	0.451483724	2.428527178
0.859128937	0.865114833	0.328248031	0.884168776	0.451737363	2.435655656
0.870693898	0.86259775	0.331077756	0.883582765	0.45111523	2.441432393
1.241080217	0.85300132	0.383550689	0.886724755	0.538802695	2.220713948
0.873400591	0.861945168	0.335383858	0.881621731	0.450688638	2.44403871
0.887672244	0.860105797	0.335999016	0.881811771	0.451088383	2.441079494
0.882012795	0.861801207	0.331938976	0.883556328	0.450364135	2.443235876
0.886687992	0.861350829	0.332923228	0.883585255	0.450607844	2.444073224
0.486958661	0.896498331	0.117125984	0.971323848	0.844723811	0.732217251
0.888533465	0.860901575	0.33390748	0.882896132	0.449755459	2.439108082
0.87832185	0.861800337	0.331938976	0.882919786	0.448234971	2.435137258
0.880167323	0.861963159	0.333292323	0.88290287	0.447356919	2.44008367
0.89554626	0.858177774	0.332554134	0.884732084	0.449259118	2.432824787
0.867249016	0.862624324	0.325172244	0.885982978	0.448942561	2.416846206
1.283218504	0.851468783	0.392470472	0.884527984	0.537378374	2.22486547
0.891978346	0.860245386	0.33550689	0.882254538	0.447651905	2.434071884
0.875615157	0.862200491	0.32898622	0.884579686	0.447461932	2.43134712
0.867372047	0.864472973	0.327755906	0.884586643	0.446721518	2.439671349
0.887180118	0.864398131	0.332554134	0.883452923	0.446271432	2.456383826
...
0.624876969	0.89622059	0.23191437	0.917739091	0.574501716	1.849620957
0.585260827	0.904185448	0.220472441	0.921441753	0.579705201	1.842880439
0.579970472	0.904898145	0.216412402	0.922674046	0.587705045	1.803689818
0.569512795	0.906434349	0.214320866	0.924044309	0.596010299	1.774962615

Lampiran IV Contoh dataset ekstraksi fitur GLCM pada sudut 135 jarak 1

contrast_135_1	correlation_135_1	dissimilarity_135_1	homogeneity_135_1	ASM_135_1	entropy_135_1
1.496186992	0.798252636	0.48012896	0.855684009	0.437808112	2.705085398
1.730299461	0.816241269	0.518569037	0.850244974	0.448011289	2.66660446
1.737925476	0.797078682	0.469526939	0.874887366	0.517759311	2.344703105
1.014198028	0.896698707	0.3500527	0.889771972	0.519464898	2.2969176
1.043090086	0.884329954	0.360592721	0.887556649	0.532104194	2.267042593
0.917849836	0.755975559	0.187364375	0.962025367	0.851172183	0.72947083
1.281790564	0.80269385	0.398288797	0.875664582	0.452246831	2.478612492
1.251782504	0.807903953	0.392584785	0.877311128	0.451989726	2.469311093
1.245954492	0.807254179	0.394072788	0.875806075	0.451097978	2.477274434
1.241862484	0.807459181	0.396924794	0.874566737	0.451774692	2.485160334
1.232066464	0.809169005	0.398040796	0.873492807	0.450382422	2.49401897
1.226982454	0.808720359	0.398164796	0.872459397	0.451017636	2.483708943
1.233430467	0.807957085	0.395932792	0.874175177	0.448515724	2.497590454
1.866637733	0.780156496	0.49023498	0.872754119	0.520905174	2.318801326
1.252278505	0.803842368	0.401512803	0.87257609	0.449178325	2.496608494
1.252650505	0.803465253	0.397916796	0.874427827	0.448286759	2.492044097
1.261702523	0.80246061	0.406720813	0.870393022	0.44578873	2.511602828
0.943765888	0.777376668	0.193192386	0.960200497	0.837880037	0.780838121
1.240994482	0.805537547	0.402504805	0.871355574	0.443559539	2.517998866
1.229214458	0.80744891	0.395932792	0.874211344	0.444106969	2.506651682
1.228346457	0.806630894	0.397668795	0.873288649	0.44293765	2.507169008
1.227478455	0.807403802	0.399404799	0.872462423	0.441926893	2.511279935
1.222022444	0.809156167	0.401388803	0.871288554	0.441829903	2.522151291
1.232314465	0.806561166	0.407836816	0.869122167	0.442074175	2.525970296
1.793725587	0.788796457	0.487258975	0.870304524	0.520184365	2.334087244
1.233802468	0.80600964	0.406720813	0.869961161	0.441025722	2.530542798
1.252278505	0.803690896	0.409696819	0.869276134	0.441236128	2.529410272
1.251782504	0.804898421	0.409200818	0.869059724	0.440070853	2.535123718
1.246822494	0.806061812	0.407340815	0.87042225	0.439554269	2.538717217
0.942401885	0.801154299	0.193812388	0.960289806	0.828879579	0.801677857
1.242606485	0.806492102	0.405480811	0.870867913	0.439535187	2.528775086
1.225742451	0.808140725	0.400768802	0.871972918	0.438839628	2.517977632
1.222766446	0.809230974	0.400520801	0.872154637	0.437438277	2.525175153
1.236902474	0.805144488	0.399280799	0.87374339	0.439229038	2.51577205
1.228594457	0.806401388	0.399280799	0.872561033	0.439112263	2.503658585
1.879471759	0.783731726	0.500341001	0.868759466	0.519124299	2.33740678
1.243598487	0.80616993	0.403124806	0.871684986	0.438027097	2.51743154
1.218302437	0.809272317	0.399032798	0.872443528	0.437546023	2.517763302
1.232686465	0.808400016	0.404612809	0.870778234	0.436590447	2.534653655
1.260586521	0.808332845	0.415276831	0.867547437	0.435294852	2.559090484
...
0.892801786	0.852638843	0.276024552	0.913036781	0.565956348	1.926024074
0.855973712	0.860730347	0.272428545	0.912779037	0.57115812	1.922587199
0.851385703	0.861259543	0.263996528	0.916387002	0.579017576	1.882902824
0.858329717	0.859867628	0.263996528	0.917335899	0.587620383	1.850314761

Lampiran V Contoh dataset ekstraksi fitur GLCM pada sudut 0 jarak 2

contrast_0_2	correlation_0_2	dissimilarity_0_2	homogeneity_0_2	ASM_0_2	entropy_0_2
2.068824405	0.721049609	0.593377976	0.835825745	0.424843857	2.805433176
2.384920635	0.746731287	0.64000496	0.831186298	0.432580317	2.769782119
2.265997024	0.735473467	0.565848214	0.859559462	0.505091876	2.423609105
1.603794643	0.836652689	0.46875	0.868863157	0.506663435	2.414916294
1.523189484	0.83109878	0.458581349	0.870051714	0.523692588	2.354570333
1.133060516	0.69877615	0.227058532	0.95541959	0.841104872	0.757938616
1.756696429	0.729603294	0.482762897	0.863334913	0.444540316	2.546291804
1.751488095	0.731231766	0.484002976	0.862999796	0.443663511	2.549747478
1.743427579	0.730308093	0.486235119	0.861190092	0.442945786	2.554144222
1.74516369	0.729438081	0.491071429	0.858822796	0.443645751	2.562108144
1.732886905	0.731610063	0.486111111	0.861017441	0.442223424	2.571170691
1.733382937	0.72978704	0.486359127	0.860872913	0.442616463	2.564877529
1.730654762	0.730551779	0.487475198	0.86060039	0.441191481	2.572677474
2.39422123	0.718078502	0.587177579	0.856930494	0.50736182	2.397235051
1.757564484	0.724705985	0.494915675	0.857773674	0.441132288	2.57301233
1.754340278	0.724764464	0.493303571	0.858480224	0.439974117	2.569820869
1.749751984	0.726060546	0.494791667	0.857399731	0.437883811	2.587885313
1.174479167	0.722970222	0.235367063	0.9531281	0.826981025	0.811677629
1.730902778	0.728781028	0.494419643	0.856446198	0.436681242	2.586751429
1.711185516	0.731961595	0.486359127	0.859257192	0.435109572	2.584491428
1.720238095	0.729207588	0.488219246	0.858958614	0.435548615	2.578346912
1.736235119	0.727589317	0.491691468	0.858146879	0.434186077	2.587934868
1.735987103	0.728901414	0.492435516	0.857763561	0.434712065	2.593707545
1.736607143	0.727412763	0.498883929	0.855434353	0.434146361	2.599199985
2.35999504	0.722177729	0.587177579	0.855922929	0.507184829	2.412197822
1.742311508	0.726068386	0.501364087	0.854595158	0.432352041	2.612033143
1.765500992	0.723248945	0.5078125	0.852702192	0.432642094	2.609623636
1.749255952	0.727374076	0.498883929	0.855908092	0.432702299	2.606595438
1.738343254	0.729618979	0.498139881	0.856219589	0.432633761	2.607398134
1.184151786	0.750159787	0.23735119	0.95309385	0.818393733	0.831442139
1.727306548	0.731022164	0.49640377	0.855877164	0.431781012	2.601427595
1.701760913	0.733642936	0.491195437	0.857200654	0.431182481	2.590828825
1.694072421	0.735711381	0.488715278	0.857588179	0.430570984	2.592702106
1.706349206	0.731201289	0.48921131	0.858658042	0.432085425	2.585688616
1.693824405	0.733102714	0.484747024	0.859711741	0.431685915	2.575402598
2.425347222	0.720974993	0.595238095	0.85492843	0.506055477	2.410981465
1.715277778	0.732663951	0.490451389	0.858314215	0.431022536	2.58809886
1.693948413	0.734819944	0.490203373	0.857873987	0.430491243	2.591727396
1.708209325	0.734499116	0.494295635	0.856777846	0.42976039	2.605522327
1.725570437	0.737644749	0.500496032	0.855179546	0.429542505	2.621001064
...
1.324032738	0.781472817	0.349826389	0.902182613	0.557274543	1.999360163
1.25297619	0.796146731	0.338045635	0.904033875	0.564354358	1.984997265
1.274305556	0.792351414	0.339781746	0.904411509	0.571038976	1.954378865
1.288194444	0.789697613	0.337797619	0.905789768	0.578876014	1.918937394

Lampiran VI Contoh dataset ekstraksi fitur GLCM pada sudut 45 jarak 2

contrast_45_2	correlation_45_2	dissimilarity_45_2	homogeneity_45_2	ASM_45_2	entropy_45_2
1.486514973	0.79955682	0.481616963	0.854491983	0.439159381	2.702278945
1.73005146	0.816267607	0.519437039	0.850891695	0.449035798	2.668281861
1.71498543	0.799757176	0.468038936	0.874215182	0.518408769	2.343363749
1.227230454	0.875000258	0.391468783	0.882547998	0.515128115	2.338447555
1.196602393	0.867306711	0.388244776	0.882823973	0.529322916	2.29077897
0.926157852	0.753766746	0.189720379	0.961450832	0.850719937	0.731006617
1.231322463	0.81046241	0.388864778	0.877405826	0.453296278	2.46835378
1.220782442	0.812661161	0.384276769	0.879084406	0.452336089	2.465348895
1.209622419	0.812874653	0.385764772	0.878042139	0.452234371	2.467067418
1.217434435	0.811246554	0.391220782	0.875759059	0.452749451	2.475969312
1.213218426	0.812088319	0.391840784	0.87480839	0.452139627	2.481649115
1.210490421	0.811291374	0.388616777	0.876316248	0.451799034	2.473070827
1.218178436	0.810331799	0.389236778	0.876488963	0.449842653	2.486070243
1.799553599	0.788057339	0.480438961	0.873796386	0.520229567	2.318383806
1.241242482	0.805571057	0.397172794	0.874041286	0.449696267	2.49272444
1.244342489	0.80476874	0.398040796	0.873809819	0.449467298	2.487238063
1.234050468	0.806789975	0.3997768	0.872548043	0.446840746	2.505139231
0.961249922	0.773252389	0.196912394	0.9594469	0.836635683	0.784130402
1.203670407	0.811386188	0.391096782	0.875000738	0.444978375	2.502696078
1.188914378	0.813761743	0.384152768	0.877325743	0.444763634	2.493596611
1.187550375	0.813053107	0.388120776	0.875286234	0.443770871	2.493718221
1.187178374	0.813727043	0.386756774	0.875940557	0.443364667	2.496236154
1.204414409	0.811906022	0.394196788	0.873414125	0.443086022	2.511457302
1.217062434	0.808955307	0.404240808	0.869563253	0.44229584	2.521721675
1.791245582	0.789088467	0.483786968	0.87154401	0.520889781	2.330809163
1.215822432	0.808836635	0.402876806	0.870558724	0.442413549	2.52124558
1.248930498	0.804215734	0.407588815	0.869753535	0.441318757	2.526547398
1.226486453	0.808841039	0.400272801	0.872283804	0.440757603	2.527928464
1.236282473	0.807701269	0.406844814	0.869855466	0.440269982	2.533477227
0.946245892	0.800343218	0.19480439	0.960157835	0.82843972	0.803222702
1.220658441	0.809910014	0.402256805	0.87089692	0.43998008	2.525358852
1.208754418	0.810799777	0.396056792	0.873736452	0.439917928	2.510622814
1.184698369	0.815170137	0.391716783	0.873935517	0.438599856	2.512382159
1.193378387	0.812001058	0.391220782	0.874656933	0.440047275	2.504697964
1.195734391	0.811579389	0.39010478	0.87458423	0.439994005	2.492153867
1.779031558	0.795289245	0.479384959	0.872956063	0.519568879	2.326952235
1.221898444	0.809552147	0.397172794	0.873057477	0.438327521	2.512144228
1.203670407	0.81156299	0.398288797	0.871256669	0.437854186	2.51154936
1.213342427	0.811406715	0.399280799	0.871856668	0.436578342	2.528601993
1.249302499	0.810048536	0.409448819	0.869623925	0.435939453	2.550895006
...
0.90482981	0.850653561	0.276644553	0.913409151	0.564834595	1.929700261
0.848657697	0.861920686	0.266972534	0.914844527	0.57184346	1.910932477
0.860561721	0.859764233	0.261640523	0.917924383	0.578838278	1.877782595
0.861429723	0.859361515	0.261392523	0.918133117	0.587062507	1.8453126

Lampiran VII Contoh dataset ekstraksi fitur GLCM pada sudut 90 jarak 2

contrast_90_2	correlation_90_2	dissimilarity_90_2	homogeneity_90_2	ASM_90_2	entropy_90_2
1.778397817	0.760209341	0.544766865	0.842884528	0.431207064	2.769858069
2.240203373	0.762099662	0.623015873	0.832470592	0.437048814	2.758441733
2.11266121	0.752600617	0.54780506	0.860938829	0.509793663	2.410398533
1.477306548	0.849535567	0.445064484	0.873150732	0.511097995	2.389479795
1.558407738	0.82719355	0.458085317	0.872543019	0.521123041	2.361677868
0.832465278	0.778689318	0.173363095	0.963935466	0.852204638	0.720548768
1.559399802	0.759971864	0.444320437	0.869941879	0.448319029	2.510425104
1.532738095	0.764799252	0.440724206	0.870133574	0.447278191	2.509250437
1.5234375	0.764338496	0.441468254	0.868890321	0.446128255	2.513990107
1.536334325	0.761813997	0.4453125	0.867674084	0.447941935	2.516878062
1.527529762	0.763415826	0.445808532	0.86717163	0.446032577	2.528915309
1.519469246	0.763133539	0.439980159	0.869633557	0.445416048	2.52248977
1.550099206	0.758662743	0.44828869	0.867182259	0.443157076	2.539197525
2.212239583	0.738646207	0.563182044	0.860076826	0.512367721	2.381957563
1.572668651	0.753666923	0.453869048	0.865817581	0.442904666	2.543355954
1.585069444	0.751321084	0.457465278	0.864523913	0.44229141	2.538684771
1.564856151	0.755007656	0.454985119	0.864891139	0.440994728	2.550193145
0.856522817	0.797968042	0.178075397	0.962480218	0.839814317	0.767348311
1.53608631	0.75930725	0.45014881	0.865429909	0.438877712	2.551793929
1.510540675	0.763390404	0.44171627	0.86786541	0.437384219	2.549035259
1.515997024	0.761358331	0.446304563	0.866444338	0.437067854	2.547854402
1.5234375	0.760976699	0.450272817	0.864862646	0.436440835	2.550970365
1.51500496	0.763410856	0.450644841	0.864018433	0.435732656	2.564095806
1.540550595	0.758186858	0.454241071	0.864229173	0.435584236	2.570115931
2.125062004	0.74893606	0.556981647	0.857856862	0.512258753	2.391154248
1.543402778	0.757341433	0.458705357	0.862432977	0.434553135	2.575319341
1.585689484	0.751435293	0.465525794	0.861254385	0.434567915	2.577121809
1.568700397	0.755514112	0.460689484	0.86225693	0.433974201	2.577468344
1.565848214	0.756448769	0.462053571	0.862042573	0.434317495	2.578634277
0.854910714	0.819625257	0.178199405	0.962749566	0.830133411	0.791107894
1.5546875	0.757902568	0.459697421	0.862387286	0.433192991	2.576580488
1.549975198	0.757400208	0.458953373	0.862005223	0.432449658	2.567329198
1.525545635	0.762002885	0.450396825	0.864751851	0.431423632	2.567331984
1.547743056	0.756186285	0.453993056	0.865131626	0.432818431	2.564541678
1.52641369	0.759481757	0.449032738	0.866001822	0.433228617	2.546663499
2.205357143	0.745404584	0.564360119	0.858385741	0.511854218	2.387893866
1.559647817	0.756919788	0.459325397	0.862482506	0.432279237	2.562418253
1.535218254	0.759668441	0.454861111	0.863671284	0.431242024	2.566255451
1.5390625	0.760789004	0.458705357	0.861712402	0.430710919	2.578210223
1.583953373	0.759176168	0.468501984	0.859780093	0.429430257	2.601214479
...
1.24702381	0.794182884	0.329985119	0.906918019	0.559237649	1.977510989
1.192088294	0.806052902	0.324156746	0.907002832	0.565221804	1.970369278
1.186259921	0.806698485	0.319940476	0.90827598	0.573113868	1.932135831
1.179315476	0.807472497	0.319320437	0.90876308	0.581395967	1.900282454

Lampiran VIII Contoh dataset ekstraksi fitur GLCM pada sudut 135 jarak 2

<u>contrast_135_2</u>	<u>correlation_135_2</u>	<u>dissimilarity_135_2</u>	<u>homogeneity_135_2</u>	<u>ASM_135_2</u>	<u>entropy_135_2</u>
1.496186992	0.798252636	0.48012896	0.855684009	0.437808112	2.705085398
1.730299461	0.816241269	0.518569037	0.850244974	0.448011289	2.66660446
1.737925476	0.797078682	0.469526939	0.874887366	0.517759311	2.344703105
1.014198028	0.896698707	0.3500527	0.889771972	0.519464898	2.2969176
1.043090086	0.884329954	0.360592721	0.887556649	0.532104194	2.267042593
0.917849836	0.755975559	0.187364375	0.962025367	0.851172183	0.72947083
1.281790564	0.80269385	0.398288797	0.875664582	0.452246831	2.478612492
1.251782504	0.807903953	0.392584785	0.877311128	0.451989726	2.469311093
1.245954492	0.807254179	0.394072788	0.875806075	0.451097978	2.477274434
1.241862484	0.807459181	0.396924794	0.874566737	0.451774692	2.485160334
1.232066464	0.809169005	0.398040796	0.873492807	0.450382422	2.49401897
1.226982454	0.808720359	0.398164796	0.872459397	0.451017636	2.483708943
1.233430467	0.807957085	0.395932792	0.874175177	0.448515724	2.497590454
1.866637733	0.780156496	0.49023498	0.872754119	0.520905174	2.318801326
1.252278505	0.803842368	0.401512803	0.87257609	0.449178325	2.496608494
1.252650505	0.803465253	0.397916796	0.874427827	0.448286759	2.492044097
1.261702523	0.80246061	0.406720813	0.870393022	0.44578873	2.511602828
0.943765888	0.777376668	0.193192386	0.960200497	0.837880037	0.780838121
1.240994482	0.805537547	0.402504805	0.871355574	0.443559539	2.517998866
1.229214458	0.80744891	0.395932792	0.874211344	0.444106969	2.506651682
1.228346457	0.806630894	0.397668795	0.873288649	0.44293765	2.507169008
1.227478455	0.807403802	0.399404799	0.872462423	0.441926893	2.511279935
1.222022444	0.809156167	0.401388803	0.871288554	0.441829903	2.522151291
1.232314465	0.806561166	0.407836816	0.869122167	0.442074175	2.525970296
1.793725587	0.788796457	0.487258975	0.870304524	0.520184365	2.334087244
1.233802468	0.80600964	0.406720813	0.869961161	0.441025722	2.530542798
1.252278505	0.803690896	0.409696819	0.869276134	0.441236128	2.529410272
1.251782504	0.804898421	0.409200818	0.869059724	0.440070853	2.535123718
1.246822494	0.806061812	0.407340815	0.87042225	0.439554269	2.538717217
0.942401885	0.801154299	0.193812388	0.960289806	0.828879579	0.801677857
1.242606485	0.806492102	0.405480811	0.870867913	0.439535187	2.528775086
1.225742451	0.808140725	0.400768802	0.871972918	0.438839628	2.517977632
1.222766446	0.809230974	0.400520801	0.872154637	0.437438277	2.525175153
1.236902474	0.805144488	0.399280799	0.87374339	0.439229038	2.51577205
1.228594457	0.806401388	0.399280799	0.872561033	0.439112263	2.503658585
1.879471759	0.783731726	0.500341001	0.868759466	0.519124299	2.33740678
1.243598487	0.80616993	0.403124806	0.871684986	0.438027097	2.51743154
1.218302437	0.809272317	0.399032798	0.872443528	0.437546023	2.517763302
1.232686465	0.808400016	0.404612809	0.870778234	0.436590447	2.534653655
1.260586521	0.808332845	0.415276831	0.867547437	0.435294852	2.559090484
...
0.892801786	0.852638843	0.276024552	0.913036781	0.565956348	1.926024074
0.855973712	0.860730347	0.272428545	0.912779037	0.57115812	1.922587199
0.851385703	0.861259543	0.263996528	0.916387002	0.579017576	1.882902824
0.858329717	0.859867628	0.263996528	0.917335899	0.587620383	1.850314761

Lampiran IX Hasil pengujian skenario uji coba

jarak	sudut	data	hidden & neuron	akurasi	presisi		recall		f1score	
					no	yes	no	yes	no	yes
1	0, 45, 90, 135	70 : 30	1 & 32	67.33%	66%	70%	84%	47%	74%	56%
1	0, 45, 90, 135	80 : 20	1 & 32	68.94%	67%	74%	86%	48%	75%	59%
1	0, 45, 90, 135	90 : 10	1 & 32	68.08%	65%	75%	86%	47%	74%	58%
2	0, 45, 90, 135	70 : 30	1 & 32	67.33%	66%	71%	86%	44%	74%	55%
2	0, 45, 90, 135	80 : 20	1 & 32	67.51%	65%	73%	87%	44%	74%	55%
2	0, 45, 90, 135	90 : 10	1 & 32	67.80%	65%	75%	87%	46%	74%	57%
1, 2	0, 45, 90, 135	70 : 30	1 & 32	70.66%	69%	74%	85%	53%	76%	62%
1, 2	0, 45, 90, 135	80 : 20	1 & 32	70.58%	68%	77%	88%	50%	77%	60%
1, 2	0, 45, 90, 135	90 : 10	1 & 32	70.53%	68%	75%	84%	55%	75%	63%
1, 2	0, 45, 90, 135	70 : 30	2 & 32	72.38%	71%	75%	85%	56%	77%	64%
1, 2	0, 45, 90, 135	80 : 20	2 & 32	71.88%	73%	71%	78%	65%	75%	68%
1, 2	0, 45, 90, 135	90 : 10	2 & 32	73.12%	71%	78%	85%	59%	77%	67%
1, 2	0, 45, 90, 135	70 : 30	3 & 32	77.25%	76%	79%	86%	67%	81%	72%
1, 2	0, 45, 90, 135	80 : 20	3 & 32	77.13%	78%	76%	82%	72%	80%	74%
1, 2	0, 45, 90, 135	90 : 10	3 & 32	79.81%	78%	82%	87%	72%	82%	77%
1, 2	0, 45, 90, 135	70 : 30	3 & 64	83.71%	84%	83%	87%	80%	86%	81%
1, 2	0, 45, 90, 135	80 : 20	3 & 64	85.12%	87%	82%	85%	85%	86%	84%
1, 2	0, 45, 90, 135	90 : 10	3 & 64	81.45%	81%	82%	85%	77%	83%	79%
1, 2	0, 45, 90, 135	70 : 30	3 & 128	85.08%	89%	81%	83%	88%	86%	84%
1, 2	0, 45, 90, 135	80 : 20	3 & 128	87.44%	90%	85%	87%	88%	88%	86%
1, 2	0, 45, 90, 135	90 : 10	3 & 128	87.45%	89%	86%	88%	87%	88%	87%
1, 2	0, 45, 90, 135	70 : 30	3 & 256	86.81%	89%	84%	88%	85%	88%	85%
1, 2	0, 45, 90, 135	80 : 20	3 & 256	90.92%	92%	90%	91%	90%	92%	90%
1, 2	0, 45, 90, 135	90 : 10	3 & 256	91.13%	92%	90%	91%	91%	92%	90%