

**PENERAPAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ) PADA
DETEKSI PENYAKIT STROKE BERDASARKAN CITRA
*COMPUTED TOMOGRAPHY SCAN***

SKRIPSI

**Oleh:
ENDAH KUMALA HADI
NIM.19650034**



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

**PENERAPAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ)
PADA DETEKSI PENYAKIT STROKE BERDASARKAN CITRA
*COMPUTED TOMOGRAPHY SCAN***

SKRIPSI

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

Oleh:
ENDAH KUMALA HADI
NIM. 19650034

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2023**

HALAMAN PERSETUJUAN

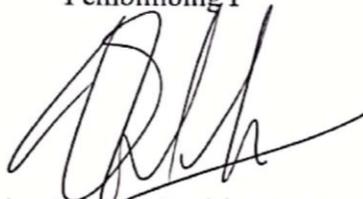
PENERAPAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ) PADA DETEKSI PENYAKIT STROKE BERDASARKAN CITRA *COMPUTED TOMOGRAPHY SCAN*

SKRIPSI

Oleh:
ENDAH KUMALA HADI
NIM. 19650034

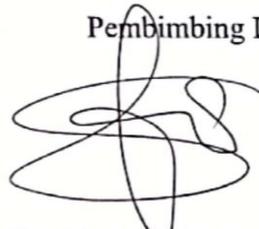
Telah Diperiksa dan Disetujui untuk Diuji
Tanggal: 4 Desember 2023

Pembimbing I



Fajar Rohman Hariri, M.Kom
NIP. 19890515 201801 1 001

Pembimbing II



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

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



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

HALAMAN PENGESAHAN

PENERAPAN METODE *LEARNING VECTOR QUANTIZATION* (LVQ) PADA DETEKSI PENYAKIT STROKE BERDASARKAN CITRA *COMOPUTED TOMOGRAPHY SCAN*

SKRIPSI

Oleh:
ENDAH KUMALA HADI
NIM. 19650034

Telah Dipertahankan Di Depan Dewan Penguji Skripsi
Dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 22 Desember 2023

Susunan Dewan Penguji

Ketua Penguji : Dr. Cahyo Crysdiان
NIP. 19740424 200901 1 008

Anggota Penguji I : Agung Teguh Wibowo Almais, M.T
NIDT. 19860301 20180201 1 235

Anggota Penguji II : Fajar Rohman Hariri, M.Kom
NIP. 19890515 201801 1 001

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

()
()
()
()

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




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

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Endah Kumala Hadi
NIM : 19650034
Fakultas/Prodi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Penerapan Metode *Learning Vector Quantization (LVQ)*
Pada Deteksi Penyakit Stroke Berdasarkan Citra *Computed Tomography SCAN*

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

Apabila dikemudian hari terbukti atau dapat dibuktikan skripsi ini merupakan hasil jiplakan, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Malang, 2 Desember 2023
Yang membuat pernyataan,



Endah Kumala Hadi
NIM. 19650034

MOTTO

“Allah tidak membebani seseorang, kecuali menurut kesanggupannya. Baginya ada sesuatu (pahala) dari (kebajikan) yang diusahakannya dan terhadapnya ada (pula) sesuatu (siksa) atas (kejahatan) yang diperbuatnya.”

HALAMAN PERSEMBAHAN

Puji Syukur atas kehadiran Allah *Subhanahu wa ta'ala*

Shalawat serta salam kepada Rasulullah *Shallallahu 'alaihi wasallam*

Penulis bersyukur telah diberikan rahmat dan petunjuk-Nya sehingga mampu menyelesaikan skripsi ini. Penulis mempersembahkan skripsi ini untuk orang tua, om, teman, sahabat yang memberikan banyak support, kasih, sayang, perhatian, arahan sehingga penulis mampu melewati dan menyelesaikan skripsi ini. Terima kasih banyak yang tak terhingga untuk segala waktu dan support yang telah diberikan.

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh

Segala puji syukur dan alhamdulillah penulis ucapkan atas segala rahmat-Nya yang melimpah, sehingga penulis berhasil menyelesaikan skripsi yang berjudul “Penerapan Metode Learning Vector Quantization (LVQ) Pada Deteksi Penyakit Stroke Berdasarkan Citra CT SCAN” sebagai syarat diraihnya gelas Sarjana pada Program Studi Teknik Informatika jenjang Strata-1 UIN Maulana Malik Ibrahim Malang. Sholawat serta salam selalu tercurahkan kepada Nabi Muhammad SAW, yang telah membawa umatnya dari zaman Jahiliyah ke jaman Islamiyah.

Penulis menyampaikan banyak terima kasih kepada seluruh pihak yang telah membantu baik dalam bentuk kritik maupun saran sehingga penyusunan skripsi ini dapat terselesaikan. Atas segala bantuan yang telah diberikan penulis menyampaikan ucapan terima kasih seiring doa ditujukan kepada yang terhormat:

1. Ibu, ayah, dan om yang senantiasa memberikan dukungan, doa, serta arahan dalam proses penyelesaian skripsi ini.
2. Prof. Dr. H. M. Zainuddin, MA, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta jajarannya.
3. Prof. Dr. Sri Harini, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang beserta jajarannya.
4. Dr. Fachrul Kurniawan ST., M.MT, IPM selaku ketua jurusan Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.

5. Fajar Rohman Hariri, M.Kom selaku dosen pembimbing I yang telah memberikan bimbingan dalam penyusunan tugas akhir hingga selesai.
6. Dr. M. Amin Hariyadi, M.T selaku dosen pembimbing II sekaligus dosen wali yang telah memberikan bimbingan dalam penyusunan tugas akhir hingga selesai.
7. Dr. Cahyo Crysdiان dan Agung Teguh Wibowo Almais, M.T selaku dosen penguji I dan II yang telah memberikan arahan dan saran dalam penyelesaian skripsi ini.
8. Seluruh Dosen dan Jajaran Staf Jurusan Teknik Informatika yang telah memberikan ilmu yang sangat bermanfaat.
9. Teman-teman YG SWAG, Healing Tai Kucing, dan Anak Kos Bapak Bardji serta teman-teman Angkatan 19 Teknik Informatika “ALIEN” yang telah memberikan banyak bantuan, support, semangat, serta motivasi dalam menyelesaikan penyusunan skripsi ini

Penulis menyadari masih terdapat banyak kekurangan dan berharap skripsi ini dapat memberikan manfaat kepada pembaca khususnya penulis sendiri. *Amin Ya Rabbal Alamin.*

Wassalamu'alaikum Warahmatullahi Wabarakatuh

Malang, Desember 2023

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN	iii
PERNYATAAN KEASLIAN TULISAN	iv
MOTTO	vi
DAFTAR ISI	x
DAFTAR GAMBAR	xii
DAFTAR TABEL	xiii
ABSTRAK	xiv
ABSTRACT	xv
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Pernyataan Masalah	6
1.3 Tujuan Penelitian	6
1.4 Batasan Masalah	6
1.5 Manfaat Penelitian	6
BAB II STUDI PUSTAKA	8
2.1 Deteksi Stroke	8
2.2 <i>Learning Vector Quantization (LVQ)</i>	13
BAB III METODOLOGI PENELITIAN	17
3.1 Pengumpulan Data	17
3.2 Desain Sistem.....	18
3.3 <i>Pre-processing</i>	19
3.3.1 <i>Grayscale</i>	20
3.3.2 <i>Filtering</i>	21
3.3.3 <i>Auto cropping</i>	22
3.3.4 <i>Removing Skull</i>	24
3.3.5 Peningkatan kontras	25
3.4 <i>Morphological Operations</i>	27
3.5 <i>Learning Vector Quantization</i>	29
3.5.1 <i>Arsitektur Learning Vector Quantization</i>	29
3.5.2 Proses training.....	32
BAB IV UJI COBA DAN PEMBAHASAN	35
4.1 Skenario Uji Coba.....	35
4.1.1 Pembagian Data	35
4.1.2 Menghitung Kinerja Sistem	37
4.2 Hasil Training	40
4.2.1 Hasil Training Rasio Data 60:40	40
4.2.2 Hasil Training Rasio Data 70:30	45
4.2.3 Hasil Training Rasio Data 80:20	50
4.3 Proses Testing	56
4.3.1 Hasil Testing Rasio Data 60:40	57
4.3.2 Hasil Testing Rasio Data 70:30	61
4.3.3 Hasil Testing Rasio Data 80:20	65

4.4 Pembahasan.....	71
BAB V KESIMPULAN DAN SARAN.....	75
5.1 Kesimpulan	75
5.2 Saran	75
DAFTAR PUSTAKA	
LAMPIRAN	

DAFTAR GAMBAR

Gambar 3. 1 Citra Otak Normal (a) dan Citra Otak Stroke (b).....	17
Gambar 3. 2 Desain Sistem.....	18
Gambar 3. 3 Alur <i>Preprocessing</i>	20
Gambar 3. 4 Citra asli (a) Citra <i>grayscale</i> (b)	21
Gambar 3. 5 Citra hasil <i>median filter</i>	22
Gambar 3. 6 Citra hasil <i>auto cropping</i>	23
Gambar 3. 7 Citra hasil <i>removing skull</i>	25
Gambar 3. 8 Citra hasil peningkatan kontras	26
Gambar 3. 9 Citra hasil <i>region of interest</i>	28
Gambar 3. 10 Arsitektur jaringan <i>Learning Vector Quantization</i>	30
Gambar 3. 11 <i>Flowchart</i> proses <i>training</i> algoritma LVQ	34
Gambar 4. 1 Contoh dataset yang dikumpulkan	36
Gambar 4. 2 Grafik konvergensi dengan <i>learning rate</i> 0.1	40
Gambar 4. 3 Grafik konvergensi dengan <i>learning rate</i> 0.2	41
Gambar 4. 4 Grafik konvergensi dengan <i>learning rate</i> 0.01	43
Gambar 4. 5 Grafik konvergensi dengan <i>learning rate</i> 0.03	44
Gambar 4. 6 Grafik konvergensi dengan <i>learning rate</i> 0.1	45
Gambar 4. 7 Grafik konvergensi <i>learning rate</i> 0.2.....	47
Gambar 4. 8 Grafik konvergensi dengan <i>learning rate</i> 0.01	48
Gambar 4. 9 Grafik konvergensi dengan <i>learning rate</i> 0.03	49
Gambar 4. 10 Grafik konvergensi dengan <i>learning rate</i> 0.1	51
Gambar 4. 11 Grafik konvergensi dengan <i>learning rate</i> 0.2	52
Gambar 4. 12 Grafik konvergensi dengan <i>learning rate</i> 0.01	53
Gambar 4. 13 Grafik konvergensi dengan <i>learning rate</i> 0.03	55
Gambar 4. 14 <i>Flowchart</i> proses <i>testing</i> algoritma LVQ.....	56
Gambar 4. 15 <i>Confusion matrix</i> 60:40 <i>learning rate</i> 0.1.....	57
Gambar 4. 16 <i>Confusion matrix</i> 60:40 <i>learning rate</i> 0.2.....	58
Gambar 4. 17 <i>Confusion matrix</i> 60:40 <i>learning rate</i> 0.01.....	59
Gambar 4. 18 <i>Confusion Matrix</i> 60:40 <i>learning rate</i> 0.03	60
Gambar 4. 19 <i>Confusion matrix</i> 70:30 <i>learning rate</i> 0.1.....	62
Gambar 4. 20 <i>Confusion matrix</i> 70:30 <i>learning rate</i> 0.2.....	63
Gambar 4. 21 <i>Confusion matrix</i> 70:30 <i>learning rate</i> 0.01.....	64
Gambar 4. 22 <i>Confusion matrix</i> 70:30 <i>learning rate</i> 0.03.....	65
Gambar 4. 23 <i>Confusion matrix</i> 80:20 <i>learning rate</i> 0.1.....	66
Gambar 4. 24 <i>Confusion matrix</i> 80:20 <i>learning rate</i> 0.2.....	67
Gambar 4. 25 <i>Confusion matrix</i> 80:20 <i>learning rate</i> 0.01.....	68
Gambar 4. 26 <i>Confusion matrix</i> 80:20 <i>learning rate</i> 0.03.....	69
Gambar 4. 27 Perbandingan hasil pengujian parameter	70

DAFTAR TABEL

Tabel 2. 1 Penelitian Terkait Deteksi dan Prediksi Penyakit Stroke	12
Tabel 2. 2 Penelitian terkait Learning Vector Quantization (LVQ)	16
Tabel 4. 1 Jumlah data penelitian.....	37
Tabel 4. 2 Rasio Pembagian Data	37
Tabel 4. 3 <i>Confusion Matrix</i>	39
Tabel 4. 4 Nilai <i>precision, recall, f1-score</i> pada <i>learning rate</i> 0.1.....	58
Tabel 4. 5 Nilai <i>Precision, Recall, dan F1-Score learning rate</i> 0.2	59
Tabel 4. 6 Nilai <i>Precision, Recall, dan F1-Score learning rate</i> 0.01	60
Tabel 4. 7 Nilai <i>Precision, Recall, dan F1-Score learning rate</i> 0.03	61
Tabel 4. 8 Nilai <i>precision, recall, dan f1-score learning rate</i> 0.1	62
Tabel 4. 9 Nilai <i>precision, recall, dan f1-score learning rate</i> 0.2	63
Tabel 4. 10 Nilai <i>precision, recall, dan f1-score learning rate</i> 0.01	64
Tabel 4. 11 Nilai <i>Precision, recall, dan f1-score learning rate</i> 0.03	65
Tabel 4. 12 Nilai <i>precision, recall dan f1-score learning rate</i> 0.1.	66
Tabel 4. 13 Nilai <i>precision, recall, dan f1-score learning rate</i> 0.2	67
Tabel 4. 14 Nilai <i>Precision, recall, dan f1-score learning rate</i> 0.01	68
Tabel 4. 15 Nilai <i>Precision, recall, dan f1-score learning rate</i> 0.03	69
Tabel 4. 16 Hasil Pengujian Parameter	70
Tabel 4. 17 Perbandingan penelitian.....	72

ABSTRAK

Hadi, Endah Kumala. 2023. **Penerapan Metode *Learning Vector Quantization* (LVQ) Pada Deteksi Penyakit Stroke Berdasarkan Citra CT SCAN**. Skripsi. Jurusan Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Fajar Rohman Hariri, M.Kom (II) Dr. M. Amin Hariyadi, M.T.

Kata kunci: Deteksi Stroke, *Learning Vector Quantization*, Stroke,

Deteksi stroke merupakan salah satu proses untuk menentukan seseorang menderita stroke atau tidak dengan menentukan jenis stroke yang sedang dialami dan area mana yang terdampak. Stroke sendiri merupakan suatu kondisi medis dimana pasokan darah ke otak mengalami penyumbatan atau pecah pembuluh darah sehingga menyebabkan area otak yang terdampak dapat mengalami mati dalam waktu yang singkat. Pada penelitian ini, deteksi stroke diimplementasikan terhadap citra ct-scan yang diambil dari Radiopaedia, sebuah platform bersifat *peer-reviewed* yang menyediakan informasi radiologi secara gratis, terbuka, dan kolaboratif. Metode yang digunakan pada penelitian ini adalah *Learning Vector Quantization* (LVQ) yang menghasilkan sebuah keluaran berupa hasil deteksi gambar stroke dan normal. Uji coba yang dilakukan terhadap deteksi penyakit stroke menggunakan total 12 skenario pengujian mampu menghasilkan hasil akurasi paling optimal sebesar 76.5%.

ABSTRACT

Hadi, Endah Kumala. 2023. **Application of Learning Vector Quantization (LVQ) Method In Stroke Disease Based on CT SCAN Images**. Undergraduate Thesis. Department of Informatics Engineering Faculty of Science and Technology Maulana Malik Ibrahim State Islamic University Malang. Supervisor: (I) Fajar Rohman Hariri, M.Kom (II) Dr. M. Amin Hariyadi, M.T.

Stroke detection is one of the processes to determine whether someone is suffering from a stroke or not by determining the type of stroke being experienced and which area is affected. Stroke itself is a medical condition where the blood supply to the brain is blocked or ruptured, causing the affected area of the brain to die in a short time. In this study, stroke detection was implemented on CT-scan images taken from Radiopaedia, a peer-reviewed platform that provides free, open-access, and collaborative radiology information. The method used in this study is Learning Vector Quantization (LVQ), which produces an output in the form of stroke and normal image detection results. The test conducted on stroke detection using a total of 12 testing scenarios was able to produce the most optimal accuracy results of 76.5%.

Key words: *Stroke Detection, Learning Vector Quantization, Stroke*

مستخلص البحث

هادي، إنداه كومالا. ٢٠٢٣. تطبيق طريقة التعلم الكمي المتجه (LVQ) في التنبؤ بالسكتة الدماغية بناء على صورة الأشعة المقطعية. البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: فجر رحمن حريري، الماجستير. المشرف الثاني: د. أمين هاردي، الماجستير.

الكلمات الرئيسية: سكتة دماغية، تنبؤ بسكتة دماغية، تعلم كمي متجه.

التنبؤ بالسكتة الدماغية هو إحدى العمليات لتحديد ما إذا كان شخص ما قد أصيب بسكتة دماغية أم لا عن طريق تحديد نوعها التي يتم تجربتها والمناطق المتأثرة. السكتة الدماغية نفسها هي حالة طبية يتم فيها حظر تدفق الدم إلى الدماغ أو تمزق الأوعية الدموية، مما يتسبب في موت منطقة الدماغ المصابة في وقت قصير. في هذا البحث، تم تنفيذ التنبؤ بالسكتة الدماغية على صور الأشعة المقطعية المأخوذة من موقع Radiopaedia، منصة تمت مراجعتها من قبل الأقران توفر معلومات أشعة مجانية ومفتوحة وتعاونية. الطريقة المستخدمة في هذا البحث هي التعلم الكمي المتجه (LVQ) الذي ينتج مخرجات في شكل السكتة الدماغية ونتائج الكشف عن الصور العادية. تمكنت التجارب التي أجريت على التنبؤ بالسكتة الدماغية باستخدام ١٢ سيناريو الاختبار من إنتاج أفضل نتائج الدقة بنسبة ٧٦.٥%.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi komputer seiring bertambahnya waktu semakin mengalami perkembangan dan mampu menjangkau segala aspek kehidupan. Dunia kedokteran menjadi salah satu aspek yang terus berkembang dalam memanfaatkan teknologi. Hal ini dapat dilihat dari alat-alat yang digunakan. Semakin canggih alat yang digunakan maka akan semakin akurat dalam membantu seorang dokter dalam melakukan diagnosa penyakit yang dialami oleh pasien.

Menurut World Health Organization (WHO), stroke dapat didefinisikan sebagai suatu keadaan manusia yang dapat menyebabkan kematian yang terjadi karena otak fokal atau global mengalami gangguan yang telah berlangsung selama 24 jam atau lebih (Z & N, 2023). Di Indonesia, penyakit stroke menjadi salah satu penyumbang terbesar angka kematian dan menduduki urutan ketiga di dunia dalam menyumbang angka kecacatan (Eshmawi et al., 2022). Angka kematian yang melonjak terjadi pada masyarakat yang tergolong dalam usia muda, hal ini dipicu oleh lambannya penanganan yang ada. Penyakit stroke terjadi akibat dari pecahnya pembuluh darah atau aliran darah di otak mengalami penyumbatan (Arifianto et al., 2014).

Stroke dapat berakibat fatal apabila tidak segera mendapatkan pertolongan. Penyakit ini dapat memicu rusaknya otak, kecacatan, bahkan dapat menyebabkan kematian. Gejala stroke yang muncul seringkali tidak disadari oleh banyak masyarakat sehingga penanganan yang dilakukan pun terlambat. Diagnosa yang

tepat diperlukan untuk mengenali gejala stroke yang muncul agar penderita stroke dapat segera ditangani. Dalam proses mengenali penyakit stroke tersebut terdapat masalah yang muncul. Masalah tersebut dapat diatasi dengan melakukan deteksi pada penyakit stroke menggunakan CT Scan untuk dapat membedakan antara otak yang normal dengan otak yang sudah terkena stroke. Sehingga dibutuhkan deteksi agar mampu membantu dokter dalam menentukan jenis stroke yang diderita oleh pasien. Selain itu, dengan adanya deteksi pada penyakit stroke ini dapat membantu mengetahui hasil prediksi yang lebih baik agar dapat dilakukan pengobatan dengan tepat.

Salah satu metode yang dapat digunakan untuk membantu dalam pengenalan citra adalah jaringan syaraf tiruan. Menurut Setyowati & Mariani (2021), jaringan syaraf tiruan merupakan sistem pemrosesan dengan kemampuan seperti manusia yang mampu menyelesaikan permasalahan kompleks melalui perubahan bobot. Manipulasi dari perubahan bobot ini dapat dilakukan dengan menggunakan beberapa algoritma salah satunya adalah algoritma Learning Vector Quantization (LVQ).

Metode Learning Vector Quantization (LVQ) merupakan salah satu metode pemrosesan citra yang dapat bekerja dengan akurasi yang baik dan memiliki waktu yang lebih cepat. Berdasarkan penelitian sebelumnya yang telah dilakukan oleh Dewi et al. (2019) tentang klasifikasi terhadap sel serviks. Penelitian ini menggunakan metode deteksi dini yaitu tes pap smear. Dari tes pap smear ini memiliki angka 5% - 50% rentang angka salah diagnosa karena pemeriksaan manual. Klasifikasi otomatis dapat dilakukan untuk meminimalisir kesalahan yang

terjadi dengan menggunakan algoritma jaringan syaraf tiruan yaitu Learning Vector Quantization (LVQ). Kemudian setelah pemeriksaan otomatis yang dilakukan menggunakan pendekatan yang diusulkan, algoritma ini mampu mengidentifikasi citra digital dengan learning rate sebesar 0,1 dengan pengurangan learning rate 0,5. Akurasi tertinggi yang mampu dicapai sebesar 93.33% dengan 45 data uji yang mampu dibaca. Algoritma Learning Vector Quantization (LVQ) selain mampu menghasilkan tingkat akurasi yang baik, algoritma ini mampu melakukan pemrosesan dengan lebih cepat. Metode lainnya juga dapat dipadukan dengan metode ini untuk menghasilkan penyelesaian masalah yang lebih baik dalam melakukan pendeteksian suatu objek.

Klasifikasi adalah proses pengelompokan atau penentuan kategori berdasarkan karakteristik atau sifat tertentu. Dalam konteks machine learning dan pengenalan pola, klasifikasi adalah tugas untuk mengidentifikasi kategori atau kelas dari objek atau situasi berdasarkan pengamatan terhadap karakteristik atau sifatnya. Seperti yang dijelaskan dalam QS. Al-Hujurat ayat 13.

يَا أَيُّهَا النَّاسُ إِنَّا خَلَقْنَاكُمْ مِنْ ذَكَرٍ وَأُنْثَىٰ وَجَعَلْنَاكُمْ شُعُوبًا وَقَبَائِلَ لِتَعَارَفُوا إِنَّ أَكْرَمَكُمْ عِنْدَ اللَّهِ أَتَقْوَمُ إِنَّ اللَّهَ عَلِيمٌ خَبِيرٌ

“Wahai manusia, sesungguhnya Kami telah menciptakan kamu dari seorang laki-laki dan perempuan. Kemudian, Kami menjadikan kamu berbangsa-bangsa dan bersuku-suku agar kamu saling mengenal. Sesungguhnya yang paling mulia di antara kamu di sisi Allah adalah orang yang paling bertakwa. Sesungguhnya Allah Maha Mengetahui lagi Mahateliti.” (QS. Al-Hujurat: 13).

Ayat tersebut menjelaskan bahwa tentang pentingnya saling mengenal dan memahami antara satu sama lain. Menurut tafsir Jalalain, dapat disimpulkan bahwa Allah SWT menciptakan manusia dari sepasang laki-laki dan perempuan, Adam

dan Hawa. Allah juga menciptakan manusia dengan berbagai suku dan bangsa dengan berbagai tingkat nasab dan keturunan. Bukan untuk membanggakan keturunan atau keturunan kita, tetapi untuk saling mengenal dan mempererat persaudaraan kita. Ketakwaan adalah satu-satunya cara untuk menilai kehormatan dan kebanggaan yang sebenarnya. Orang yang paling bertakwa di sisi Allah adalah yang paling mulia. Allah SWT adalah Yang Maha Mengetahui dan Mengetahui apa yang ada di dalam hati setiap orang.

Stroke merupakan penyakit yang menurut mayoritas dokter dapat disembuhkan namun menurut kebanyakan masyarakat tidak bisa disembuhkan. Dalam perspektif Islam, al-Qur'an memiliki salah satu nama yaitu As-Syifa' yang artinya adalah penawar (obat). Bagi mereka yang beriman, al-Qur'an dianggap mampu memberikan obat dari setiap penyakit dengan berdoa namun juga perlu untuk melakukan ikhtiar. Namun melakukan pencegahan jauh lebih mudah daripada menyembuhkan. Sehingga setiap orang perlu untuk menerapkan pola hidup sehat agar terhindar dari penyakit. Hal ini dijelaskan dalam QS. Al-Isra' ayat 82.

وَنُنَزِّلُ مِنَ الْقُرْآنِ مَا هُوَ شِفَاءٌ وَرَحْمَةٌ لِّلْمُؤْمِنِينَ ۖ وَلَا يَرْبُدُّ الظَّالِمِينَ إِلَّا خَسَارًا

“Dan Kami turunkan dari Al-Qur'an (sesuatu) yang menjadi penawar dan rahmat bagi orang yang beriman, sedangkan bagi orang yang zalim (Al-Qur'an itu) hanya akan menambah kerugian.” (QS. Al-Isra': 82)

Ayat tersebut menjelaskan bahwa Allah SWT. telah menurunkan al-Qur'an yang dapat menjadi penawar bagi penyakit hati manusia seperti rasa ragu, gundah, ataupun sifat munafik yang dapat merusak diri manusia. Dalam tafsir Jalalain menjelaskan bahwa pada ayat tersebut menyatakan keutamaan Al-Qur'an

sebagai penawar dan rahmat bagi orang-orang yang beriman. Al-Qur'an dapat menjadi penyembuh dan penawar penyakit baik fisik maupun mental, dan juga memberikan rahmat dan keberkahan bagi mereka yang mengamalkannya. Namun, bagi orang-orang yang bertindak zalim dan tidak mengimani Al-Qur'an, ayat ini juga mengingatkan bahwa Al-Qur'an tidak akan memberikan manfaat bagi mereka dan justru akan memberikan kerugian. Oleh karena itu, setiap orang perlu untuk menghargai dan mengamalkan ajaran Al-Qur'an sebagai pedoman hidup yang benar karena ketika menerapkan prinsip-prinsip ajaran Al-Qur'an dalam kehidupan sehari-hari dapat membantu mencegah dan mengatasi penyakit, termasuk stroke.

Pada penelitian ini, deteksi pada penyakit stroke perlu dilakukan evaluasi. Evaluasi performa dilakukan dengan menguji seberapa besar nilai akurasi yang mampu dihasilkan oleh metode deteksi yang digunakan. Semakin besar nilai akurasi yang dihasilkan, maka semakin baik pula sistem yang dibuat dapat digunakan.

Penelitian ini menggunakan data citra berupa citra CT Scan (Computed Tomography Scan) otak. Data citra CT Scan yang ada kemudian akan dilakukan preprocessing yang dapat digunakan sebagai acuan sebelum melakukan deteksi penyakit stroke. Deteksi ini akan membagi data citra yang ada menjadi dua kelas, yaitu normal dan stroke. Adapun hasil penelitian yang diharapkan yaitu penelitian yang dilakukan dapat mencapai hasil akurasi yang baik sehingga penelitian yang dilakukan dapat bermanfaat dalam mengenali objek berupa gambar CT Scan otak untuk mendeteksi penyakit stroke sehingga resiko fatal dapat diminimalisir.

1.2 Pernyataan Masalah

Seberapa besar nilai akurasi dari metode Learning Vector Quantization (LVQ) dalam mendeteksi penyakit stroke berdasarkan citra CT Scan ?

1.3 Tujuan Penelitian

Mengukur nilai akurasi dari metode Learning Vector Quantization (LVQ) dalam melakukan deteksi pada penyakit stroke menggunakan citra CT Scan otak.

1.4 Batasan Masalah

Terdapat batasan-batasan masalah yang diterapkan pada penelitian yang dikerjakan agar tetap terarah dan fokus pada tujuan yang ada antara lain :

- a. Data yang digunakan pada penelitian ini berupa dataset citra CT Scan otak.
- b. Citra yang digunakan pada penelitian ini berupa citra CT Scan (*Computed Tomography Scan*) dengan format (*.jpg dan *.png).
- c. Data CT Scan yang digunakan berasal dari Radiopaedia dengan alamat "<https://radiopaedia.org/>" yang diambil pada tanggal 16 Agustus.

1.5 Manfaat Penelitian

Beberapa manfaat yang didapatkan dari penelitian ini antara lain :

- a. Bagi pihak Rumah Sakit khususnya dokter dan bidang Radiologi, penelitian ini dapat digunakan untuk membantu mempercepat proses diagnosis penyakit stroke dan meningkatkan keakurasian agar pasien dapat menerima penanganan tepat waktu dan tepat.
- b. Bagi mahasiswa Teknik Informatika dapat digunakan untuk mengerti konsep *Image Processing* dan Jaringan Syaraf Tiruan *Learning Vector*

Quantization dalam penerapan terhadap citra medis yang merupakan salah satu bidang *Artificial Intelligent*.

BAB II

STUDI PUSTAKA

2.1 Deteksi Stroke

Stroke merupakan salah satu jenis penyakit yang berbahaya. Gejala penyakit ini adalah kelumpuhan otot dan reflek kehilangan kemampuan melihat seperti penglihatan terasa kabur, kelelahan, kepala terasa ringan, kesemutan atau sensasi sentuhan berkurang, kesulitan berbicara (Sari et al., 2019). Penyakit stroke tergolong kedalam penyakit yang sering sekali tidak disadari oleh masyarakat, sehingga ketika gejala tersebut muncul kondisi penderita sudah dalam tahap parah. Penyebab dari penyakit stroke adalah adanya pecah pembuluh darah atau pembuluh darah mengalami penyumbatan (Rachman, 2021). Adanya teknologi deteksi objek dapat membantu dalam penanganan atau pengenalan suatu objek (Li et al., 2020). Objek tersebut berupa penyakit stroke dengan bantuan pengenalan terhadap citra medis yaitu CT Scan. Dengan adanya deteksi dini stroke dapat meningkatkan akurasi yang dihasilkan dan penanganan medis yang lebih cepat dan tepat. Deteksi sendiri dapat dijabarkan sebagai proses pemeriksaan terhadap objek menggunakan cara dan teknik tertentu (Apriani et al., 2020).

Tazin et al. (2021) melakukan deteksi dan prediksi terhadap penyakit stroke dengan menggunakan empat metode yang berbeda, yaitu Logistic Regression (LR), Decision Tree (DT), Random Forest Classifier (RF), dan Voting Classifier. Metode RF memiliki tingkat akurasi yang lebih baik dibandingkan metode lainnya, yaitu 96%. Adapun data yang digunakan diperoleh dari open-access Stroke Prediction Dataset. Penelitian ini juga memberikan hasil akurasi yang

lebih baik dibandingkan metode yang digunakan pada penelitian sebelumnya dengan menggunakan dataset yang sama.

Hokkinen et al. (2021) melakukan penelitian untuk memprediksi volume infark akhir dari CTA dan membandingkan hasilnya dengan perangkat lunak berbasis CTP. Metode yang digunakan pada penelitian ini adalah CNN. Sedangkan data pasien stroke yang digunakan berasal dari Rumah Sakit Universitas Helsinki dari bulan Januari 2018 hingga Juni 2019. Total pasien yang dilakukan analisa sebanyak 83 pasien. Metode CNN menghasilkan tingkat korelasi yang baik sebanyak $r = 0.89$, sensitivitas 1.00 dan spesifitas sebanyak 0.94.

Rachman (2021) melakukan diagnosa terhadap pasien yang memiliki gejala penyakit stroke berdasarkan pengujian yang dilakukan pada program sistem pakar yang dibuat oleh penulis. Sistem pakar ini menerapkan metode Case Based Reasoning dan algoritma yang digunakan adalah Probabilistic Symmetric yang mampu membantu dalam mengambil keputusan berdasarkan pernyataan yang bersifat pasti. Pengujian pada sistem akar diagnosa penyakit stroke ini berdasarkan pada tampilan pertanyaan terkait gejala yang dirasakan oleh pasien, kemudian gejala-gejala yang dipilih akan mengantarkan pasien pada hasil diagnosa penyakit stroke. Golongan penyakit stroke pada sistem pakar ini ada 4 yaitu, Stroke Iskemik, Stroke Hemoragik, Stroke Ringan, Stroke Brain Stem. Pada sistem pakar ini, penentuan jenis penyakit stroke menggunakan algoritma K-Nearest Neighbor (KNN) untuk menentukan similarity dari gejala penyakit stroke yang ada, kemudian hasil dari perhitungan KNN sebelumnya digunakan untuk perhitungan disimilaritas

pada algoritma probabilistic symmetric yang didapatkan hasil yaitu 0.1667. Setelah itu dilakukan perhitungan similaritas dan didapatkan hasil akhir 66.66%.

Z & N (2023) melakukan klasifikasi penyakit stroke dan dalam proses pengujiannya menggunakan metode K-Nearest Neighbor (KNN). Data yang digunakan dalam penelitian ini berjumlah 5110 record yang dibagi kedalam 3 split data training dan testing yaitu: 90%:10%, 80%:20%, dan 70%:30%. Nilai K yang digunakan adalah K=3 dan K=5 untuk mengetahui akurasi tertinggi dari proses pengujian. Sebelum tahap pengujian, terlebih dahulu dilakukan preprocessing pada data yang digunakan. Pada tahap preprocessing ini diketahui bahwa masih terdapat missing value berjumlah 201 data pada 11 atribut dataset yang ada. Kemudian dilakukan proses pengujian dataset pada K=3 dan K=5. Pada K=3, diperoleh hasil akurasi tertinggi pada perbandingan 70%:30% sebesar 89.17%. Selanjutnya pada K=5, hasil akurasi terbaik pada perbandingan 90%:10% sebesar 93.54%. Sehingga akurasi terbaik dari pengujian algoritma KNN menggunakan confusion matrix terletak pada K=5 dan perbandingan data training dan data testing sebesar 90%:10%.

Peng et al. (2020) melakukan sebuah penelitian untuk memprediksi stroke yang didasarkan pada data fisiologis dasar, riwayat penyakit, dan lingkungan hidup pasien. Penelitian tersebut menggunakan Artificial Neural Network. Penyakit stroke yang didiagnosis pada penelitian ini yaitu stroke iskemik, stroke hemoragik, dan pendarahan subarachnoid yang mana menggunakan data sebanyak 43.400 data training. Dataset tersebut kemudian dibagi menjadi 3 subset, 70% dataset training, 15% dataset uji, dan 15% dataset untuk validasi. Penelitian ini menerapkan dua algoritma dari metode ANN yaitu Levenberg Marquardt dan Scaled Conjugate

Gradient. Kedua algoritma tersebut mampu meningkatkan keakuratan dari prediksi stroke yaitu sebesar 98% pada kondisi 1000-fold cross validation dengan 10 set tes yang menghasilkan deviasi error terbaik sebanyak 1%. Pada penelitian ini, hidden node berpengaruh pada performa namun tidak terlalu berpengaruh terhadap akurasi pada klasifikasi yang terjadi di validasi dan testing.

Kemudian penelitian yang dilakukan oleh Subudhi et al., (2020) untuk diagnosis stroke iskemik akut. Dataset yang digunakan yaitu 192 slice dataset MRI otak yang dibagi menjadi tiga jenis stroke. Penelitian ini menggunakan Expectation-Maximization (EM) untuk proses segmentasi bagian yang terkena dampak stroke dan kemudian dilanjutkan untuk bagian yang terpisah menggunakan Fractional-Order Darwinian Particle Swarm Optimization (FODPSO). Citra otak yang sudah disegmentasi kemudian akan dilakukan pengklasifikasian. Pada penelitian ini, dilakukan perbandingan model klasifikasi yang akan diterapkan yaitu Support Vector Machine dan Random Forest. Dimana diperoleh hasil bahwa, RF memiliki tingkat akurasi lebih tinggi dibandingkan SVM yaitu sebesar 93,4%. Sedangkan SVM hanya sebesar 92%.

Berdasarkan penelitian yang telah dikumpulkan terkait deteksi dan klasifikasi penyakit stroke diperoleh hasil seperti pada Tabel 2.1.

Tabel 2. 1 Penelitian Terkait Deteksi dan Prediksi Penyakit Stroke

No.	Peneliti	Topik	Metode	Subjek
1.	Tazin et al. (2021)	Deteksi dan prediksi penyakit stroke	Logistic Regression (LR), Decision Tree (DT), Random Forest Classifier (RF), dan Voting Classifier	Open-access Stroke Prediction Dataset berjumlah 5110 record yang dibagi menjadi split data training dan testing sebesar 80%:20%.
2.	Hokkinen et al. (2021)	Prediksi volume infark akhir dari CTA	CNN	Data pasien stroke berjumlah 83 data dari Rumah Sakit Universitas Helsinki dari bulan Januari 2018 hingga Juni 2019.
3.	Rachman (2021)	Diagnosa pasien yang memiliki gejala penyakit stroke	<i>Case Based Reasoning</i> , <i>Probabilistic Symmetric</i> , dan <i>K-Nearest Neighbor (KNN)</i>	Data pasien stroke.
4.	Z & N (2023)	Klasifikasi penyakit stroke	<i>K-Nearest Neighbor (KNN)</i>	Data berjumlah 5110 record yang dibagi kedalam 3 split data training dan testing yaitu : 90%:10%, 80%:20%, dan 70%:30%.
5.	Peng et al. (2020)	Prediksi stroke berdasarkan pada data fisiologis dasar, Riwayat penyakit, dan lingkungan hidup pasien.	<i>Artificial Neural Network (ANN)</i> menggunakan algoritma <i>Levenberg Marquardt</i> dan <i>Scaled Conjugate Gradient</i> .	Dataset sebanyak 43.400 data yang dibagi menjadi 3 subset, 70% data training, 15% dataset uji, dan 15% dataset untuk validasi.

Lanjutan Tabel 2.1

No.	Peneliti	Topik	Metode	Subjek
6.	Subudhi et al. (2020)	Diagnosis stroke iskemik akut	<i>Expectation-Maximization (EM)</i> dan <i>Fractional-Order Darwinian Particle Swarm Optimization (FODPSO)</i>	192 slice dataset MRI otak yang dibagi menjadi tiga jenis stroke.

Berdasarkan Tabel 2.1, terdapat beberapa perbedaan dengan penelitian terdahulu yaitu terdapat berbagai metode yang dapat digunakan untuk melakukan deteksi dan prediksi terhadap penyakit stroke seperti metode KNN, ANN, LR, DT, RF, dan Voting Classifier. Dari beberapa metode tersebut, terdapat metode yang memiliki tingkat akurasi yang lebih baik dibandingkan metode lainnya. Seperti penelitian Tazin et al. (2021) yang menggunakan metode Random Forest (RF) dan penelitian oleh Hokkinen et al. (2021) menggunakan metode CNN.

Penelitian terdahulu yang telah dikumpulkan menunjukkan pula bahwa jumlah dataset berpengaruh terhadap tingkat akurasi yang dihasilkan oleh metode tersebut. Seperti penelitian Tazin et al. (2021) yang memiliki akurasi lebih baik dibandingkan penelitian sebelumnya dengan menggunakan dataset yang sama.

2.2 Learning Vector Quantization (LVQ)

Algoritma Learning Vector Quantization merupakan jenis algoritma jaringan syaraf tiruan yang mengklasifikasikan data berdasarkan kelas target ketika proses pembelajaran dilakukan. Tujuan dari algoritma Learning Vector Quantization (LVQ) adalah untuk mengenali pola prototipe yang terdapat pada lapisan kompetitif terawasi (supervised learning). Pada penerapannya, algoritma

LVQ telah banyak digunakan terutama dalam mengenali pola citra, suara, dan lain-lain. Algoritma LVQ merupakan single-layer yang mana setiap lapisan inputnya terhubung dengan lapisan output dengan bantuan neuron atau bobot dalam setiap koneksinya.

Hendriyani (2020) membahas tentang pengenalan pola bangun datar geometri. Pada penelitian ini bertujuan untuk membandingkan metode Backpropagation dan Learning Vector Quantization (LVQ) dalam mengenali pola objek pada bangun datar geometri. Input dari penelitian ini adalah citra dari bentuk bangun datar geometri yang diproses dalam program Matlab. Berdasarkan output yang dihasilkan, menunjukkan bahwa metode Backpropagation bekerja lebih baik dalam pengenalan pola bangun geometri dibandingkan metode learning vector quantization. Metode backpropagation mampu menghasilkan nilai akurasi lebih banyak yaitu sebesar 91.43% dengan tingkat pengenalan citra sebanyak 32 pola dari 35 pola yang ada dibandingkan metode learning vector quantization yaitu sebesar 65.71% pada iterasi nilai epoch yang ditentukan.

Menurut Santony et al. (2020) dalam penelitiannya untuk meningkatkan pengenalan kualitas citra sidik jari. Permasalahan yang muncul dalam penelitian ini adalah kualitas gambar sidik jari yang terkadang memiliki kualitas yang rendah disebabkan oleh jenis kulit yang berbeda. Pada proses memperjelas kualitas gambar yang ada dengan cara memadukan dua metode yaitu Minutiae Extraction untuk meningkatkan kualitas gambar sidik jari dan Learning Vector Quantization untuk melakukan pengecekan keakurasian dari proses mengenali citra sidik jari. Penelitian ini menggunakan empat jenis data citra yang berbeda jumlahnya, yaitu

25, 50, 75, dan 100. Setelah dilakukan pengujian, didapatkan hasil bahwa akurasi terbaik terdapat pada 75 jenis data citra dengan nilai akurasi sebesar 83.34%. Sehingga dengan perpaduan kedua metode tersebut dianggap mampu membantu dalam pengenalan sidik jari dengan lebih baik.

Penelitian berikutnya dilakukan Nurpadillah et al. (2019) untuk mengenali pola aksara sunda. Penelitian ini menggunakan data yang berasal dari gambar pola aksara sunda yaitu aksara swara dengan menerapkan metode Learning Vector Quantization (LVQ), yang mana menggunakan 6 jenis nilai learning rate dan 10 jenis hidden layer. Epoch yang diberikan pada masing-masing variasi sama yaitu 1000, sehingga dari proses pengujian pada penelitian ini didapatkan hasil bahwa pada learning rate 0.01 dengan jumlah hidden layer sebesar 90 layer metode LVQ dapat bekerja secara optimal. Data yang digunakan pada penelitian ini sebanyak 42 data dan yang berhasil diverifikasi sebanyak 28 data dengan epoch terbaik pada 241, kemudian nilai akurasi terbaik yang didapat yaitu 66.66% dengan besar MSE sebanyak 0.408 dalam waktu 17 menit 22 detik.

Zheng & Xie (2018) melakukan penelitian untuk mengidentifikasi tanda-tanda lalu lintas pada teknologi mengemudi cerdas. Model pengenalan gambar untuk membantu mengurangi efek buruk dari informasi gambar yang tidak memadai dan tersembunyi pada penelitian ini adalah klasifikasi CNN, PCA dan LVQ. Tingkat akurasi jika menggunakan metode CNN hanya 92,57%. Sedangkan apabila model tersebut dikombinasikan dengan model PCA dan LVQ, tingkat akurasi yang dihasilkan lebih tinggi yaitu 94,62%. Adapun data yang digunakan

berasal dari GTSRB yang khusus untuk klasifikasi tanda-tanda lalu lintas yang memuat 43 jenis tanda.

Berdasarkan penelitian yang telah dikumpulkan terkait penerapan metode learning vector quantization (lvq) diperoleh hasil seperti pada Tabel 2.2 berikut.

Tabel 2. 2 Penelitian terkait Learning Vector Quantization (LVQ)

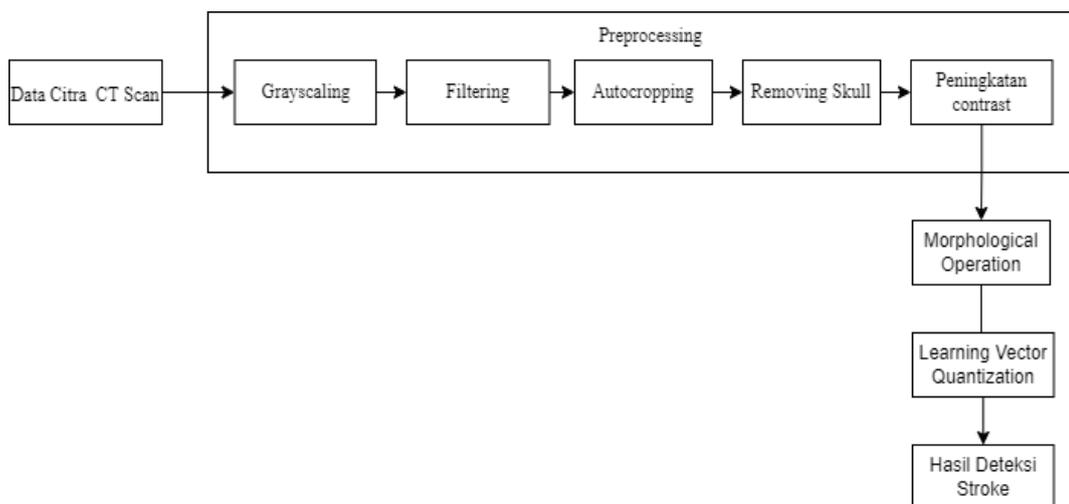
No.	Peneliti	Topik	Metode	Subyek
1.	Hendriyani (2020)	Pengenalan pola bangun datar geometri	<i>Backpropagation</i> dan <i>Learning Vector Quantization (LVQ)</i> .	Citra bangun datar geometri.
2.	Santony et al. (2020)	Peningkatan pengenalan kualitas citra sidik jari	<i>Minutiae Extraction</i> dan <i>Learning Vector Quantization (LVQ)</i> .	Citra sidik jari dengan jumlah yang berbeda, 25 citra, 50 citra, 75 citra, dan 100 citra.
3.	Nurpadillah et al. (2019)	Pengenalan karakter tulisan (pola aksara sunda)	<i>Learning Vector Quantization (LVQ)</i>	Citra pola aksara sunda sebanyak 42 data
4.	Zheng & Xie (2018)	Identifikasi tanda-tanda lalu lintas pada teknologi <i>smart-driving</i>	CNN, PCA, dan LVQ	GTSRB untuk klasifikasi tanda-tanda lalu lintas yang terdapat 43 jenis tanda

Berdasarkan Tabel 2.2 dapat disimpulkan bahwa metode Learning Vector Quantization (LVQ) dapat digunakan untuk melakukan suatu pengenalan terhadap data citra. Selain itu, pada penelitian sebelumnya metode LVQ juga dapat dikombinasikan dengan metode lainnya untuk pengenalan pola maupun citra. Subjek yang digunakan pun juga bervariasi, tergantung pada tujuan penelitian untuk memperoleh hasil akurasi yang lebih baik.

Pada kelas Normal, dataset yang digunakan sebanyak 160 citra CT Scan. Sedangkan pada kelas Hemoragik menggunakan sebanyak 160 citra CT Scan otak. Data citra yang digunakan pada penelitian ini hanya file dengan format '.jpg' dan '.png' yang kemudian sudah disamaratakan menjadi '.png'. Kemudian, dataset tersebut akan digunakan untuk melatih model deteksi menggunakan Learning Vector Quantization (LVQ).

3.2 Desain Sistem

Pada penelitian ini, akan dikembangkan sebuah sistem dengan beberapa tahapan yang telah dijelaskan dalam Gambar 3.2 berikut.



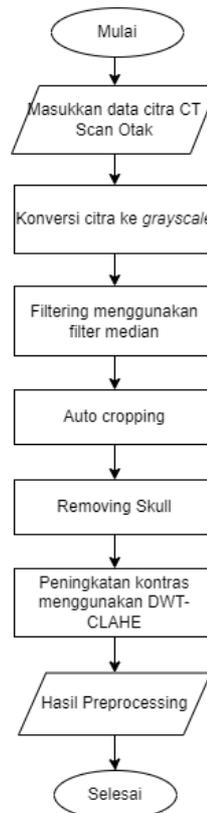
Gambar 3. 2 Desain Sistem

Sistem yang dibuat akan dikembangkan berdasarkan desain sistem pada Gambar 3.2, menggunakan bahasa pemrograman Python dan menggunakan dataset citra CT Scan otak sebagai input. Data akan diproses melalui beberapa tahapan seperti grayscaleing, filtering, autocropping, removing skull, dan juga peningkatan kontras melalui preprocessing, serta dilakukan morphological operations untuk mengetahui area yang perlu dilakukan analisis untuk kemudian dilakukan diagnosa.

Selanjutnya, akan dibuat sebuah model deteksi menggunakan metode Learning Vector Quantization (LVQ). Kemudian, model tersebut akan diuji dengan menggunakan skenario uji coba dan confusion matrix untuk evaluasi terhadap performa agar diperoleh nilai akurasi, presisi, recall, dan f1-score.

3.3 Pre-processing

Data yang sebelumnya telah dikumpulkan harus melalui tahap preprocessing sebelum dapat digunakan pada tahap berikutnya. Setiap tahap preprocessing disesuaikan dengan kebutuhan penelitian. Pada penelitian ini, preprocessing yang dilakukan meliputi beberapa proses yaitu grascaling, median filter, autocropping, removing skull, dan peningkatan kontras. Tujuan dilakukannya preprocessing terhadap data citra CT Scan otak adalah agar dapat membantu dalam analisis dan diagnosis medis sehingga memastikan bahwa citra yang digunakan mempunyai kualitas yang baik ketika pengolahan data dan siap untuk digunakan sebagai proses analisis lebih lanjut. Adapun alur preprocessing yang akan dilakukan pada penelitian ini dapat dilihat pada Gambar 3.3 berikut.



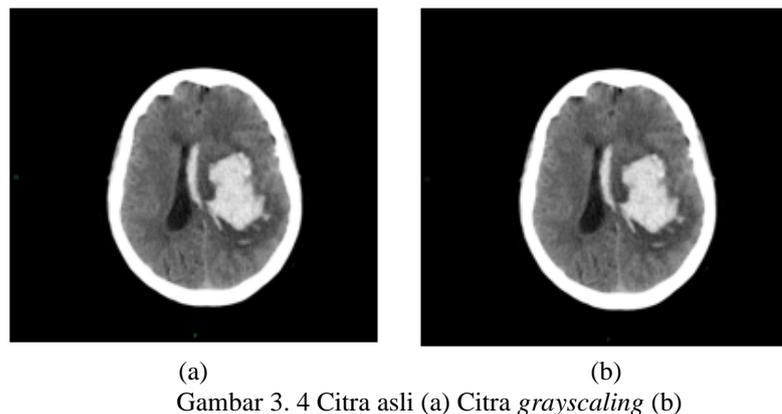
Gambar 3. 3 Alur *Preprocessing*

3.3.1 *Grayscale*

Grayscale merupakan proses untuk melakukan konversi citra berwarna menjadi tingkat keabuan atau grayscale. Citra berwarna memiliki piksel yang terdapat komponen warna berupa R (merah), G (hijau), dan B (biru). Proses grayscale menghasilkan piksel yang hanya memiliki nilai kecerahan tunggal dan memiliki nilai intensitas berskala abu-abu tanpa ada informasi warna. Grayscale dapat dihitung dengan metode rata-rata bobot yang mana nilai intensitas dihasilkan dari penggabungan komponern RGB. Masing-masing komponen tersebut dikalikan dengan bobotnya masing-masing sehingga diperoleh nilai kecerahan piksel baru. Rumus yang digunakan untuk proses grayscale sebagai berikut :

$$\text{Grayscale} = (0.299 * R) + (0.587 * G) + (0.114 * B) \quad (3.1)$$

Rumus pada persamaan 3.1 mengambil komponen R, G, B kemudian dijumlahkan dengan bobot masing-masing komponen sehingga menghasilkan skala keabuan yang merepresentasikan intensitas cahaya pada piksel tersebut. Jika jumlah total dari ketiga komponen tersebut semakin besar, maka dalam skala abu-abu akan semakin terang pula pikselnya. Hasil dari proses grayscale ini digunakan dalam tahapan preprocessing selanjutnya. Gambar 3.4 berikut menampilkan citra CT Scan otak sebelum dan sesudah dilakukan proses grayscale.



Gambar 3. 4 Citra asli (a) Citra *grayscale* (b)

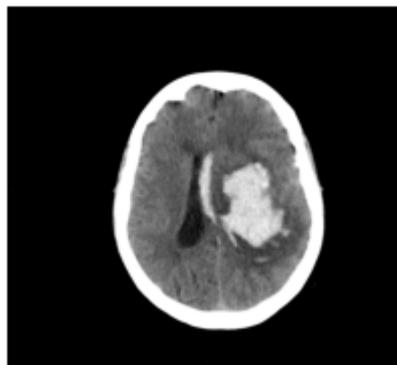
3.3.2 Filtering

Filtering citra merupakan langkah untuk memproses citra dengan menggunakan suatu filter untuk meningkatkan kualitas citra. Tujuan dari filtering citra adalah untuk menghilangkan noise atau informasi yang tidak diinginkan dari citra dan meningkatkan fitur-fitur dari citra yang akan digunakan pada tahap selanjutnya. Metode filtering citra yang digunakan pada penelitian ini adalah median filter. Median filter merupakan salah satu dari jenis smoothing filter. Jenis filter ini bertujuan untuk menghilangkan noise yang terdapat dalam citra dengan mengganti nilai piksel di lokasi tertentu dengan nilai median dari kumpulan

intensitas piksel tetangga disekitarnya. Pada proses ini menyangkut proses mengurutkan piksel-piksel dalam lingkup tersebut dengan berurutan secara ascending yaitu mengurutkan dari yang terkecil ke yang terbesar atau sebaliknya. Kemudian nilai tengah (median) dipilih jika jumlah pikselnya ganjil dan menggantinya sebagai nilai baru dari piksel yang sedang diproses (Sundararajan, 2017). Rumus yang digunakan pada median filter sebagai berikut (Shaban et al., 2018):

$$F(X,Y) = \text{median}\{g(s,t)\} \quad (3.2)$$

Rumus pada persamaan 3.2 menunjukkan bahwa Median filter dilakukan dengan menghitung nilai piksel pada citra hasil dengan mengambil nilai tengah dari kumpulan piksel tetangga. Tahapan ini dilakukan setelah citra input hasil grayscale, kemudian dilakukan smoothing untuk mengurangi noise dengan menggunakan kernel 3x3. Pada gambar 3.5 berikut ditunjukkan bagaimana citra sebelum diterapkan filter median dan juga setelah diterapkan.

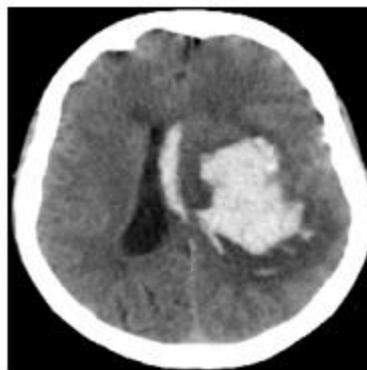


Gambar 3. 5 Citra hasil *median filter*

3.3.3 *Auto cropping*

Auto cropping merupakan langkah pemrosesan citra yang bertujuan untuk melakukan identifikasi dan memotong area tertentu dari sebuah citra sehingga

daerah-daerah yang tidak relevan akan dibuang. Tujuan dilakukannya auto cropping adalah untuk mempertahankan area yang dianggap penting saja untuk selanjutnya digunakan dalam pencarian Region of Interest (ROI). Auto cropping yang diterapkan pada penelitian ini dilakukan dengan menggunakan teknik deteksi kontur. Citra yang dihasilkan berdasarkan proses auto cropping dengan menerapkan inputan hasil dari median filter sebelumnya dapat dilihat pada Gambar 3.6 berikut.



Gambar 3. 6 Citra hasil *auto cropping*

Citra yang sebelumnya sudah di preprocessing menjadi bentuk citra grayscale dan telah diterapkan median filter akan digunakan sebagai input proses yang akan digunakan pada deteksi kontur untuk memperoleh area yang lebih relevan. Deteksi kontur yang digunakan pada penelitian ini dilakukan dengan mencari kontur eksternalnya untuk mengetahui kontur terluar dari citra dan juga dilakukan penghilangan titik-titik yang tidak signifikan agar menghemat memori. Proses selanjutnya dilakukan dengan mencari kontur terbesar dari kontur eksternal yang ditemukan pada citra. Pencarian kontur terbesar ini diperlukan pada deteksi kontur karena menunjukkan objek utama dari citra. Citra yang telah diperoleh kontur terbesar kemudian dipotong berdasarkan titik-titik ekstrem dari citra input.

Setelah itu dilakukan resize terhadap citra yang telah berhasil dipotong untuk menghilangkan area yang tidak relevan untuk menghindari kesalahan pada penentuan Region of Interest (ROI). Resize pada pemrosesan citra ini bertujuan untuk mengubah ukuran citra menjadi 256x256 agar tidak terlalu besar ketika dilakukan komputasi sehingga kualitas yang diperoleh akan lebih baik dan analisis dapat berjalan dengan optimal. Selain itu, digunakan untuk menyamakan ukuran citra yang sebelumnya berbeda-beda.

3.3.4 Removing Skull

Removing skull pada processing ini bertujuan untuk memisahkan bagian tengkorak dengan bagian otak. Tujuan diterapkan teknik ini agar analisis yang dilakukan dapat berfokus pada area otak dan menghilangkan area lain yang tidak diperlukan. Pada penelitian ini, removing skull yang dijalankan menerapkan 2 metode yaitu yang pertama menggunakan Laplacian Filter dan Thresholding.

Laplacian filter merupakan salah satu operator deteksi tepi yang digunakan untuk melakukan identifikasi terhadap perubahan tajam yang terjadi pada intensitas citra. Filter ini bekerja dengan cara menghitung laplacian citra. Laplacian disini merupakan sebuah turunan kedua dari intensitas citra terhadap koordinat spasial. Rumus yang digunakan pada penelitian ini adalah sebagai berikut (Sundararajan, 2017) :

$$\nabla^2 f(x, y) = \frac{\partial^2 f(x, y)}{\partial x^2} + \frac{\partial^2 f(x, y)}{\partial y^2} \quad (3.3)$$

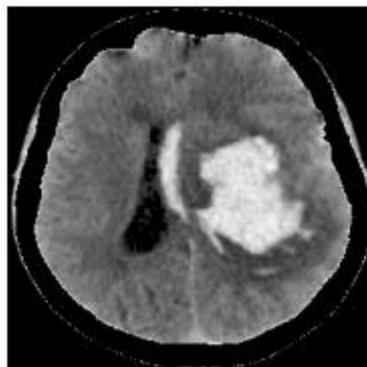
Keterangan :

$\nabla^2 f(x, y)$: Operator Laplacian yang digunakan untuk mengukur perubahan tajam pada $f(x, y)$

$\frac{\partial^2 f(x, y)}{\partial x^2}$: Turunan kedua dari fungsi $f(x, y)$ terhadap variabel x untuk mengukur perubahan dengan arah horizontal

$\frac{\partial^2 f(x, y)}{\partial y^2}$: Turunan kedua dari fungsi $f(x, y)$ terhadap variabel y untuk mengukur perubahan dengan arah vertikal

Metode kedua yang digunakan adalah thresholding. Metode ini digunakan untuk menghilangkan elemen-elemen citra yang tidak diperlukan. Metode ini diterapkan ke dalam daerah di luar tepi sehingga mempertegas tepi berdasarkan Laplacian Filter. Pada penelitian ini kernel alpha yang digunakan adalah 0.5 dengan thresholding yang ditetapkan sebesar 0.007. Sehingga citra dengan nilai dibawah 0.007 akan diubah menjadi 0. Contoh hasil proses segmentasi otak terlihat pada Gambar 3.7.

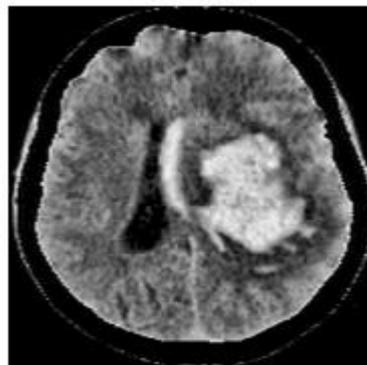


Gambar 3. 7 Citra hasil *removing skull*

3.3.5 Peningkatan kontras

Pada tahap terakhir preprocessing ini dilakukan dengan menerapkan dua metode yaitu Discrete Wavelet Transform (DWT) dan Contrast Limited Adaptive

Histogram Equalization (CLAHE). DWT merupakan salah satu metode yang digunakan untuk analisis data dalam domain frekuensi waktu. DWT dilakukan dengan membagi citra menjadi beberapa tingkatan resolusi dan frekuensi sehingga dapat dilakukan pemrosesan yang memiliki resolusi berbeda. Sedangkan CLAHE merupakan metode processing citra yang berguna untuk peningkatan citra secara adaptif. Peningkatan kontras yang dilakukan berdasarkan pada histogram citra. Dimana CLAHE merupakan metode modifikasi dari Histogram Equalization (HE). Hasil penerapan antara metode DWT dan CLAHE mampu menghasilkan kontras yang lebih baik daripada sebelumnya. Gambar 3.8 merupakan contoh penerapan DWT-CLAHE terhadap citra yang sebelumnya telah melalui beberapa processing lainnya.



Gambar 3. 8 Citra hasil peningkatan kontras

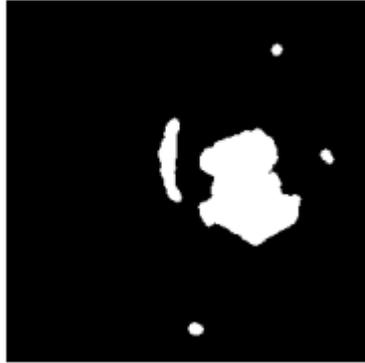
Proses ini pertama dilakukan dengan menerapkan library dwt yang berasal dari pywt. Citra yang telah melalui proses segmentasi untuk menghilangkan area tengkorak agar tidak terjadi kesalahan pemrosesan dibagi menjadi dua yaitu c_a (Approximation Coefficient) dan juga c_d (Detail Coefficient) yang disimpan kedalam wavelet filter bior5.5. Kemudian dilakukan thresholding terhadap koefisien approx. menggunakan CLAHE terhadap data yang telah diterapkan DWT.

Pada tahap penerapan CLAHE ini menggunakan clip limit sebesar 0.01. Citra hasil rekonstruksi kemudian disimpan dalam ukuran 256x256. Setelah itu, menghasilkan citra yang memiliki kontras lebih baik.

3.4 *Morphological Operations*

Region of Interest (ROI) merupakan suatu area tertentu pada sebuah citra yang dipilih yang bertujuan untuk memisahkan atau menghilangkan background sehingga objek dari citra mudah untuk dianalisis (Widiyanto, 2020). Adanya ROI pada suatu citra dapat mengurangi jumlah data yang perlu diproses. ROI dapat digunakan untuk pengolahan citra medis (Pratomo et al., 2020). Salah satunya pada citra CT Scan penderita stroke. Dengan adanya ROI, dapat membantu tenaga kesehatan dalam memeriksa bagian otak yang terdampak stroke dengan memisahkan area otak dengan area lain dari citra CT Scan. ROI dapat diperoleh dengan menerapkan metode Morphological Operations.

Morphological operations merupakan rangkaian operasi matematika untuk melakukan manipulasi dan mengubah bentuk objek dalam citra berdasarkan struktur dan karakteristik spasialnya. Operasi morfologi memiliki dua operasi dasar yaitu erosi dan dilasi. Adapun contoh hasil dari proses region of interest (ROI) berdasarkan morphological operations dapat dilihat pada Gambar 3.9 berikut.



Gambar 3. 9 Citra hasil *region of interest*

ROI yang diperoleh dari gambar tersebut dilakukan dengan menerapkan binary thresholding ke dalam citra hasil preprocessing. Binary thresholding dilakukan dengan mengubah citra menjadi citra biner yaitu 255 (putih) dan 0 (hitam). Nilai ambang batas diterapkan pada proses ini untuk mengubah citra, yaitu menggunakan nilai 110. Citra diatas nilai ambangng batas akan dirubah menjadi putih dan citra yang berada di bawah nilai ambang batas akan dirubah menjadi hitam. Piksel yang memiliki intensitas diatas 110 maka akan dijadikan bagian dari mask.

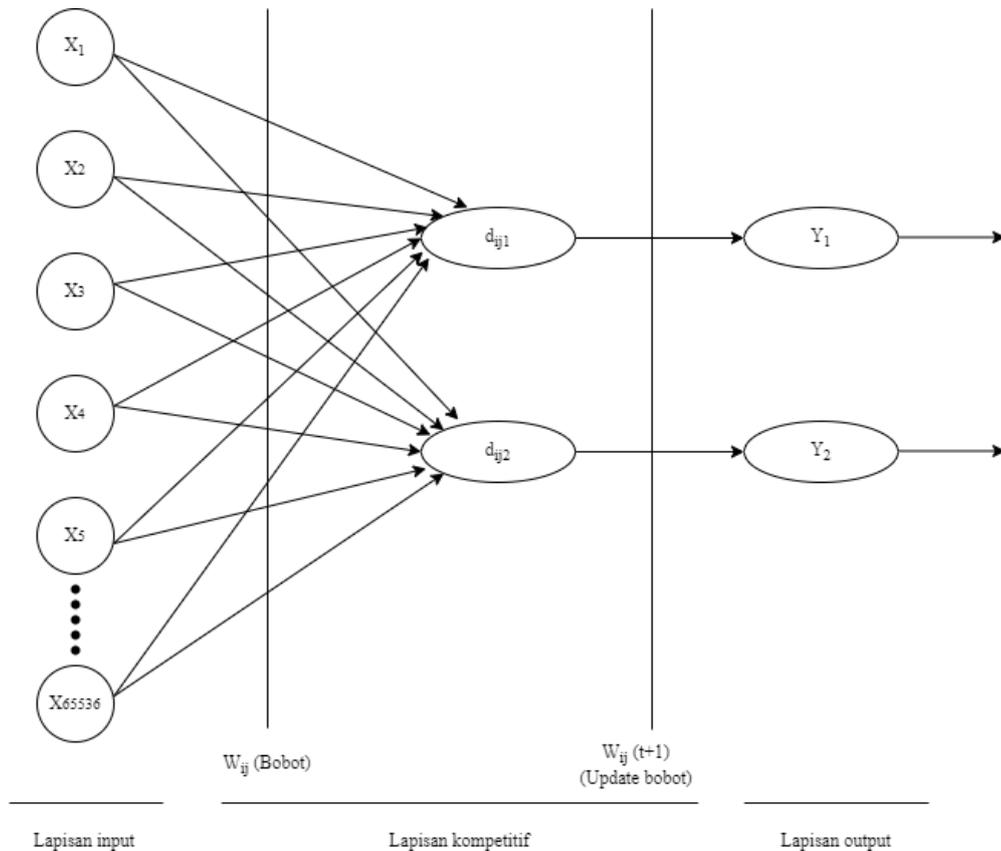
Setelah itu, dilanjutkan dengan menerapkan morphological operations yaitu erosi dan dilasi. Operasi yang dilakukan pada tahap ini juga melibatkan structuring element dalam bentuk disk berjari-jari 7 piksel. Erosi dilakukan untuk menipiskan atau mengikis tepi piksel yang memiliki label 255 (putih) pada citra biner. Kemudian dilanjutkan dengan menerapkan dilasi setelah proses erosi selesai. Dilasi dilakukan dengan mengisi lubang-lubang kecil yang kosong pada masker atau disebut juga dengan menebalkan area masker untuk menghilangkan bagian

yang masih terdapat noise. Hasil masker pada tahap dilasi ini yang nantinya akan digunakan sebagai input pada proses analisis.

3.5 *Learning Vector Quantization*

3.5.1 *Arsitektur Learning Vector Quantization*

Metode LVQ memiliki proses pembelajaran yang perlu dilakukan dengan cara menentukan maksimal epoch yang harus dilewati. Lapisan kompetitif pada LVQ akan belajar secara otomatis untuk mengklasifikasikan vektor input yang diberikan. Vektor input yang ada dikelompokkan kedalam suatu kelas yang sama sebagai hasil dari proses pada lapisan kompetitif berdasarkan node tetangganya atau jarak yang berdekatan. Metode LVQ digunakan untuk mengklasifikasikan pola ke dalam beberapa kelas dengan mencari bobot akhir melalui perhitungan jarak antara data dan bobot akhir (Rahmat et al., 2017). Sehingga LVQ dapat direpresentasikan kedalam 3 bagian, yaitu layer masukan (input), layer kompetitif, dan layer keluaran (output).



Gambar 3. 10 Arsitektur jaringan *Learning Vector Quantization*

Pada Gambar 3.10, dapat dilihat bahwa terdapat lapisan input yang berperan sebagai penerima data citra. Masukan pada layer ini berupa intensitas piksel citra hasil preprocessing yaitu hasil dari proses operasi morfologi untuk mendapatkan citra masking. Layer kedua yaitu lapisan kompetitif. Lapisan kompetitif ini berisi vektor input X_i yang akan dibandingkan dengan vektor bobot W_{ij} dari setiap neuron input dengan menggunakan jarak d_{ij} . Pada lapisan kompetitif ini, bobot yang digunakan diperoleh dari jumlah piksel dikalikan dengan jumlah neuron kompetitif yang digunakan, dimana jumlah piksel sebanyak 65536 piksel dengan 2 neuron kompetitif. Sehingga jumlah bobot yang digunakan sebanyak 131072. Kemudian layer terakhir yaitu lapisan output. Layer ini berisi hasil deteksi

untuk vektor input yang sebelumnya telah diperoleh neuron dengan jarak terdekat yang dipilih sebagai pemenang. Gambar 3.11 menunjukkan flowchart dari metode Learning Vector Quantization.

X_i adalah nilai input pada neuron ke- i di lapisan input. Nilai input ini berupa intensitas piksel. Setelah melalui inisialisasi bobot prototipe dan juga iterasi untuk mengevaluasi setiap sampel pelatihan, kemudian dilakukan perhitungan menggunakan d_{ij} atau Euclidean untuk menghitung jarak antara sampel pelatihan dengan setiap prototipe kelas. Secara umum, jarak Euclidean dapat dirumuskan seperti berikut (Hariri, 2018) :

$$d_{ij} = \sqrt{\sum_{i=0}^{n-1} (X_{ij} - W_{ij})^2} \quad (3.4)$$

Keterangan :

d_{ij} : jarak *Euclidean*.

i : indeks setiap elemen vector, sedangkan n merupakan jumlah neuron pada lapisan input.

X_{ij} : vektor input.

W_{ij} : vektor bobot yang menghubungkan neuron ke- i pada lapisan input dengan neuron ke- j di lapisan kompetitif.

Bobot pemenang akan diperbarui sesuai dengan kelas sampel setelah diketahui prototipe mana yang paling dekat dengan pemenang untuk memperbarui representasi kelompok kelas. Update bobot dilakukan dengan menggunakan rumus persamaan 3.5 dan 3.6.

Jika kelas sampel sesuai dengan target :

$$w_{ij}(t + 1) = w_{ij}(t) + \alpha(t) \cdot (x_i(t) - w_{ij}(t)) \quad (3.5)$$

Jika kelas sampel tidak sesuai dengan target :

$$w_{ij}(t + 1) = w_{ij}(t) - \alpha(t) \cdot (x_i(t) - w_{ij}(t)) \quad (3.6)$$

Keterangan :

$w_{ij}(t+1)$: bobot baru.

x : piksel masukan.

N_e : nilai *neighborhood*.

t : waktu.

α : *learning rate*.

3.5.2 Proses training

Proses training merupakan proses yang dilakukan pada metode learning vector quantization (lvq) untuk mengubah vektor bobot dari setiap neuron input agar sesuai dengan kelas yang diwakilinya. Tahap ini melibatkan perhitungan jarak antara vektor input dengan vektor bobot dari setiap neuron serta melakukan penyesuaian bobot berdasarkan pada aturan tertentu yang ada. Proses pembelajaran ini sesuai untuk proses pengelompokkan suatu pola. Tahapan yang dijalankan pada proses training ini termasuk dengan melakukan inisialisasi bobot prototipe, iterasi melalui setiap sampel pelatihan, menghitung jarak Euclidean, pemilihan bobot pemenang, dan pembaruan bobot prototipe.

Adapun algoritma yang dikerjakan pada proses pembelajaran metode LVQ secara garis besar sebagai berikut (Kusumadewi, 2004):

1. Tetapkan: bobot awal dari unit input ke-j terhadap kelas ke-i yang merepresentasikan unit input ke-j (W_{ij}), maksimum epoh (*MaxEpoch*)

yang digunakan, parameter *learning rate* (α), pengurangan *learning rate* ($\text{Dec } \alpha$), dan menggunakan minimal *learning rate* ($\text{Min } \alpha$).

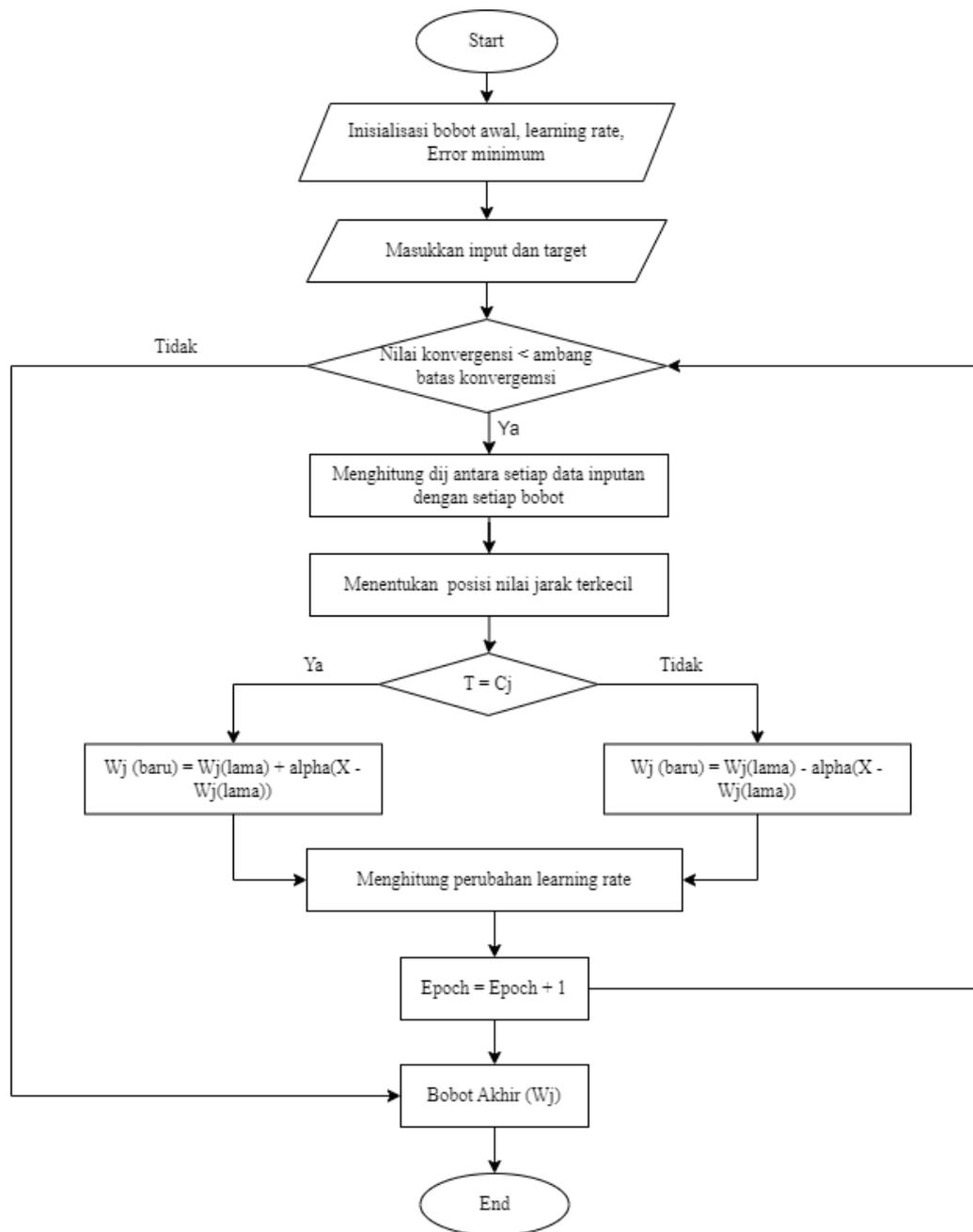
2. Masukkan: data input X_{ij} dan kelas target T_k .
3. Tetapkan kondisi awal: $\text{Epoch}=0$.
4. Kerjakan jika: ($\text{epoch} \leq \text{MaxEpoch}$) dan ($\alpha \geq \text{Min } \alpha$)
 - a. $\text{Epoch} = \text{Epoch} + 1$;
 - b. Kerjakan untuk $i = 1$ sampai n
 - i. Tentukan j sedemikian hingga $|X_i - W_j|$ minimum (C_j)
 - ii. Perbaiki W_j dengan ketentuan:
 - Jika $T = C_j$ maka:

$$W_j (\text{new}) = W_j (\text{old}) + \alpha (X_i - W_j (\text{old})) \quad (3.7)$$
 - Jika $T \neq C_j$ maka:

$$W_j (\text{new}) = W_j (\text{old}) - \alpha (X_i - W_j (\text{old})) \quad (3.8)$$
 - c. Kurangi nilai *learning rate*.

Setelah melalui proses training maka akan diperoleh bobot-bobot akhir.

Alur proses training pada metode Learning Vector Quantization (LVQ) digambarkan pada Gambar 3.11 berikut.



Gambar 3. 11 Flowchart proses training algoritma LVQ

BAB IV

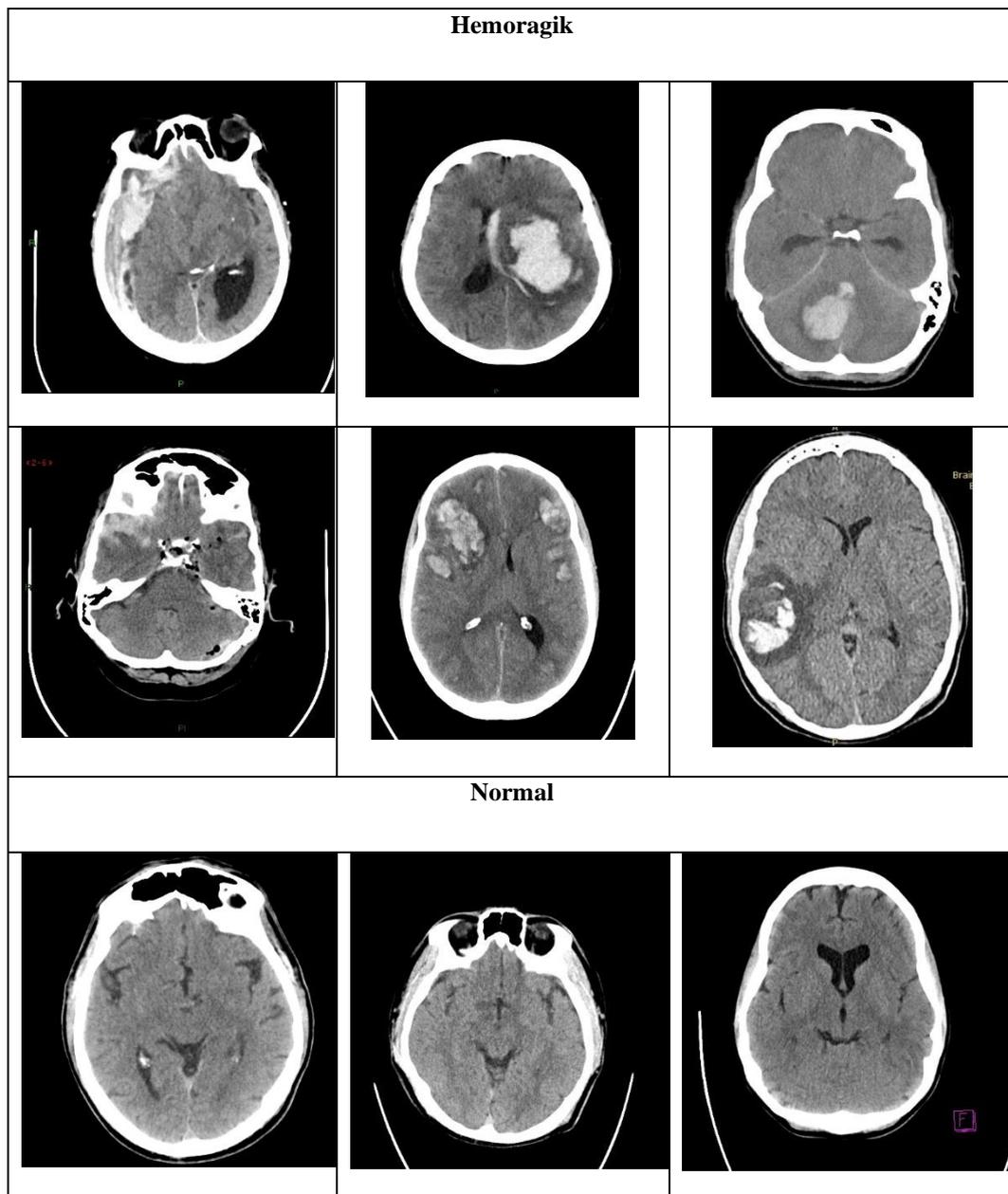
UJI COBA DAN PEMBAHASAN

4.1 Skenario Uji Coba

Pada penelitian ini dilakukan beberapa tahapan uji coba terhadap sistem dengan teratur dan dan terstruktur. Tahap pertama yang dilakukan dengan membagi dataset menjadi data pelatihan dan pengujian. Kemudian untuk mendapatkan hasil deteksi citra stroke menggunakan model LVQ dengan inputan berupa hasil dari masking. Hasil deteksi kemudian dilakukan perhitungan menggunakan confusion matrix untuk mendapatkan hasil akurasi, presisi, recall, dan f1-score. Selain itu, perhitungan juga dilakukan dengan menerapkan beberapa learning rate pada rasio split data yang digunakan untuk mengetahui akurasi tertinggi dapat dihasilkan pada learning rate beberapa.

4.1.1 Pembagian Data

Dataset yang digunakan pada penelitian ini terdiri dari dua kelas, yaitu kelas Normal dan Stroke. Pada kelas Normal terdapat 160 data citra dan untuk kelas Stroke terdapat 160 data citra. Dataset diperoleh secara online dari website <https://radiopaedia.org/>. Berikut adalah contoh dataset yang diambil dari Radiopaedia.org, dimana terdapat beberapa jenis stroke yang berbeda, seperti yang terlihat pada Gambar 4.1.



Gambar 4. 1 Contoh dataset yang dikumpulkan

Data yang telah dikumpulkan tersebut kemudian dilakukan proses preprocessing dan resize untuk memperkecil ukuran piksel yang digunakan untuk pelatihan dan pengujian agar lebih mudah ketika pengolahan data. Selanjutnya data hasil preprocessing yang telah dilakukan resize akan menunjukkan area yang

teridentifikasi terkena stroke. Area tersebut ditunjukkan dengan adanya daerah lesi yang berhasil dideteksi. Data tersebut digunakan sebagai data pelatihan yang kemudian dilakukan pemisahan data menjadi beberapa ratio data, yaitu 60:40, 70:30, dan 80:20. Data pelatihan yang digunakan lebih besar daripada data pengujian. Dataset yang digunakan dapat dilihat pada Tabel 4.2.

Tabel 4. 1 Jumlah data penelitian

Kelas	Jumlah
Normal	160 data
Stroke	160 data

Pembagian data pelatihan dan pengujian dilakukan dengan ratio 0.6, 0.7, 0.8 dengan cara melakukan inisialisasi pada variabel ratio. Pembagian dataset ini dilakukan dengan membagi antara training dan validation berdasarkan ratio. Variasi pembagian data yang berbeda pada penelitian ini dapat memberikan hasil akurasi yang optimal dengan ratio pembagian terbaik yang diperoleh. Pembagian data ratio training dan validation untuk masing-masing kelas dapat dilihat pada Tabel 4.3 sebagai berikut.

Tabel 4. 2 Rasio Pembagian Data

Rasio	Training	Validation
0.6	192	128
0.7	224	96
0.8	256	64

4.1.2 Menghitung Kinerja Sistem

Pada penelitian ini, dataset yang ada dibagi menjadi training dan testing pada masing-masing kelas. Dalam persentase, data dibagi menjadi 60% data latih dan 40% data validasi, 70% data latih dan 30% data validasi, serta 80% data latih

dan 20% data uji. Selain itu, pada skenario uji coba ini dilakukan perubahan terhadap learning rate untuk mengetahui pengaruhnya terhadap tingkat akurasi yang dihasilkan. Learning rate merupakan parameter penting yang digunakan dalam proses training model LVQ yang menentukan seberapa cepat atau lambat model belajar dari data. Berdasarkan hal tersebut, dilakukan uji coba menggunakan empat nilai learning rate yang berbeda pada penelitian ini, yaitu 0.1, 0.2, 0.01, dan 0.03. Pemilihan parameter ini berdasarkan pada penelitian yang telah dilakukan oleh Yu et al. (2020) dengan mengemukakan parameter learning rate 0.1 hingga 5.0 sehingga dapat memberikan hasil yang baik dalam menurunkan nilai loss. Keempat nilai learning rate tersebut dilakukan uji coba dengan menggunakan pengaturan yang sama pada parameter yang berbeda. Pada skenario uji coba ini juga dilakukan penghitungan menggunakan confusion matrix.

Confusion matrix digunakan untuk mengevaluasi performa dari suatu model klasifikasi. Dalam confusion matrix, hasil prediksi dari model klasifikasi akan dibandingkan dengan hasil sebenarnya (aktual) pada data uji atau validasi. Pada tahap ini akan menampilkan jumlah data yang diklasifikasikan dengan benar atau salah dalam setiap kategori, baik kategori positif maupun negative. Adapun confusion matrix terdiri dari empat kategori, yaitu True Positive (TP), False Positive (FP), True Negative (TN), dan False Negative (FN). Masing-masing memiliki pengertian sebagai berikut :

1. TP (*True Positive*) : Jumlah data pasien yang benar mengalami stroke dan diklasifikasikan sebagai stroke oleh model.

2. FP (*False Positive*) : Jumlah data pasien yang sebenarnya tidak mengalami stroke tetapi salah diidentifikasi sebagai stroke.
3. TN (*True Negative*) : Jumlah pasien yang benar tidak mengalami stroke dan diklasifikasikan sebagai non-stroke oleh model.
4. FN (*False Negative*) : Jumlah data pasien yang sebenarnya stroke tetapi tidak terdeteksi oleh model sehingga dianggap sebagai non-stroke.

Tabel 4.4 menunjukkan pembagian antara kelas prediksi dan kelas aktual. Setelah dilakukan pengujian, akan diperoleh hasil akurasi dan juga jumlah data yang berhasil dideteksi dengan baik. Dengan adanya matriks confusion matrix dapat digunakan sebagai acuan untuk mengevaluasi kinerja model deteksi yang diterapkan.

Tabel 4. 3 *Confusion Matrix*

<i>Confusion Matrix</i>		<i>Predicted Class</i>	
		<i>Negative (N)</i>	<i>Positive (P)</i>
<i>Actual Class</i>	<i>Negative (N)</i>	<i>TN (True Negative)</i>	<i>FP (False Positive)</i>
	<i>Positive (P)</i>	<i>False Negative (FN)</i>	<i>True Positive (TP)</i>

Berdasarkan kategori tersebut, kemudian dihitung akurasi, presisi, recall, dan F1-score dengan menggunakan persamaan berikut :

$$Accuracy = \frac{TP + TN}{(TP + FP + TN + FN)} \quad (4.1)$$

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

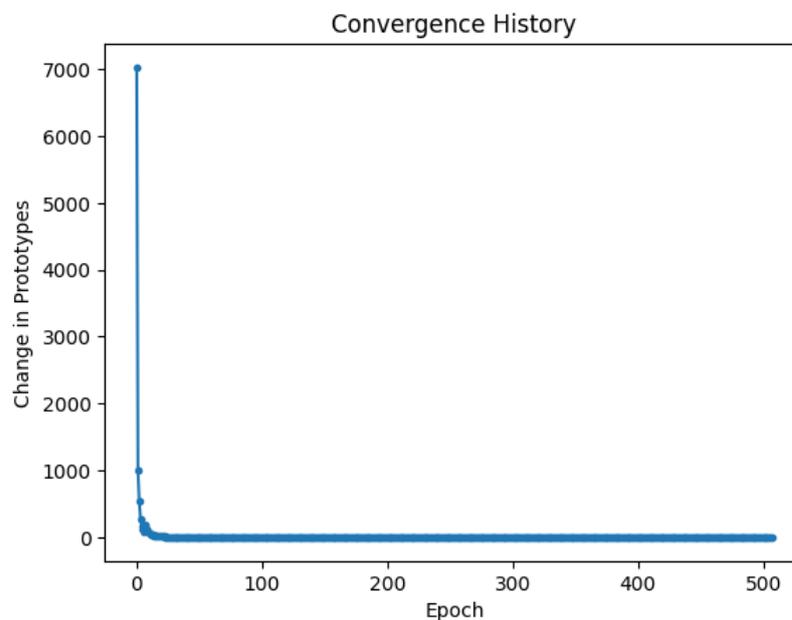
$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

$$F1 - score = 2 \times \frac{Presisi * Recall}{Presisi + Recall} \quad (4.4)$$

4.2 Hasil Training

4.2.1 Hasil Training Rasio Data 60:40

Pelatihan yang dilakukan pada penelitian ini menggunakan nilai random state sebanyak 5, dan convergence threshold atau ambang konvergensi sebesar $1e-2$ serta patience sebanyak 10. Hasil uji coba pada skenario 1 ini dilakukan dengan menerapkan ratio data 60:40 dengan learning rate sebesar 0.1.

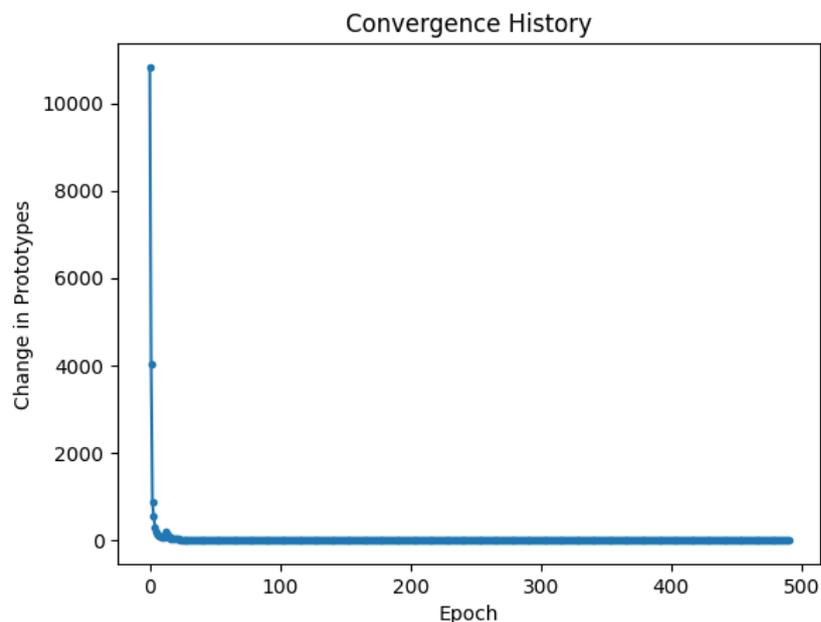


Gambar 4. 2 Grafik konvergensi dengan *learning rate* 0.1

Berdasarkan Gambar 4.1 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.1 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak pada 7027.741188037243. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model

LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai diatas 1012.5477547155662. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.1 dengan rasio data 60:40 berhenti pada epoch 506.

Selanjutnya Gambar 4.2 menunjukkan grafik konvergensi jika menggunakan learning rate 0.2.

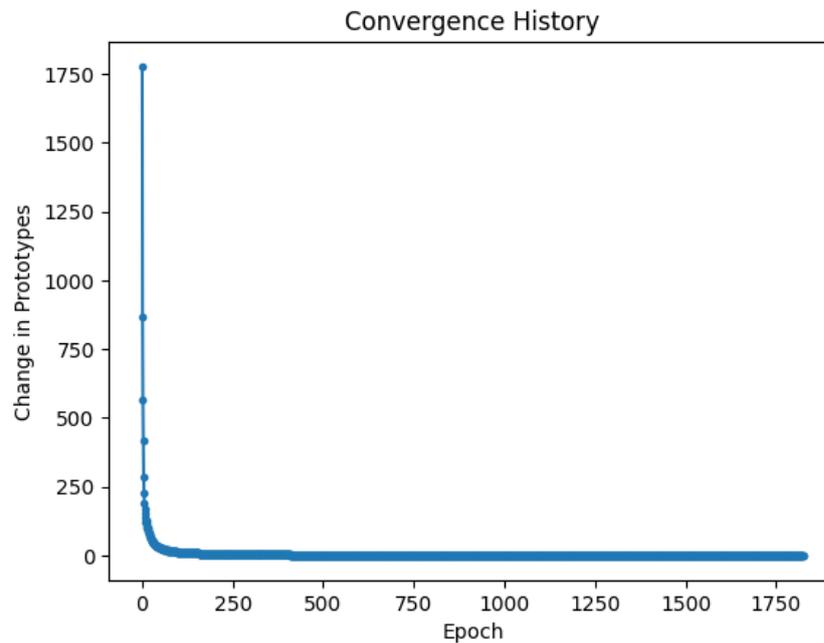


Gambar 4. 3 Grafik konvergensi dengan *learning rate* 0.2

Berdasarkan Gambar 4.2 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot

prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.2 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak diatas 10000. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai diatas 4000. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.2 dengan rasio data 60:40 berhenti pada epoch 491.

Pelatihan selanjutnya dilakukan dengan mengubah parameter learning rate menjadi 0.01 untuk rasio data sama 60:40.

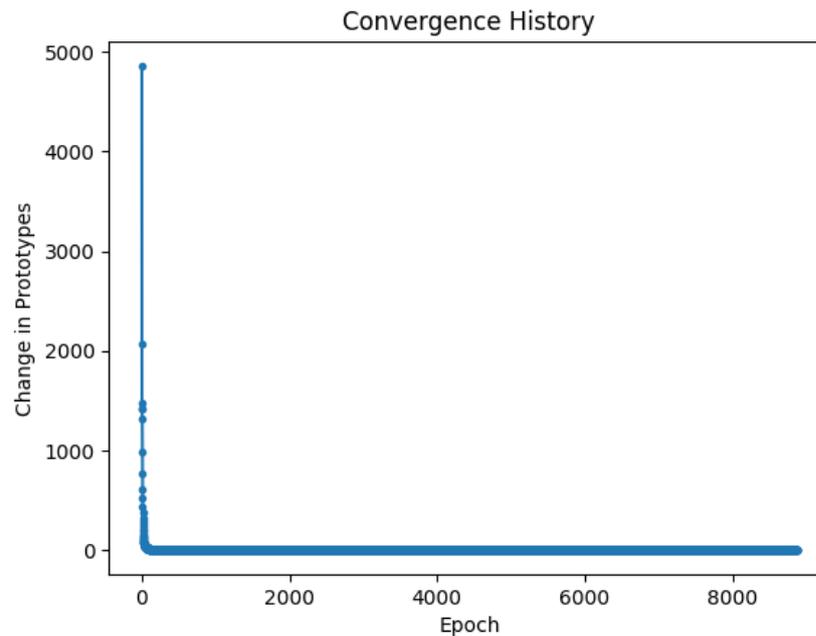


Gambar 4. 4 Grafik konvergensi dengan *learning rate* 0.01

Berdasarkan Gambar 4.3 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.3 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak diatas 1776.6723435788836. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai diatas 864.7947014587294. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut

hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.1 dengan rasio data 60:40 berhenti pada epoch 1826.

Parameter pengujian selanjutnya yang digunakan adalah learning rate 0.03 dengan ratio data 60:40 menunjukkan hasil konvergensi seperti pada Gambar 4.4.



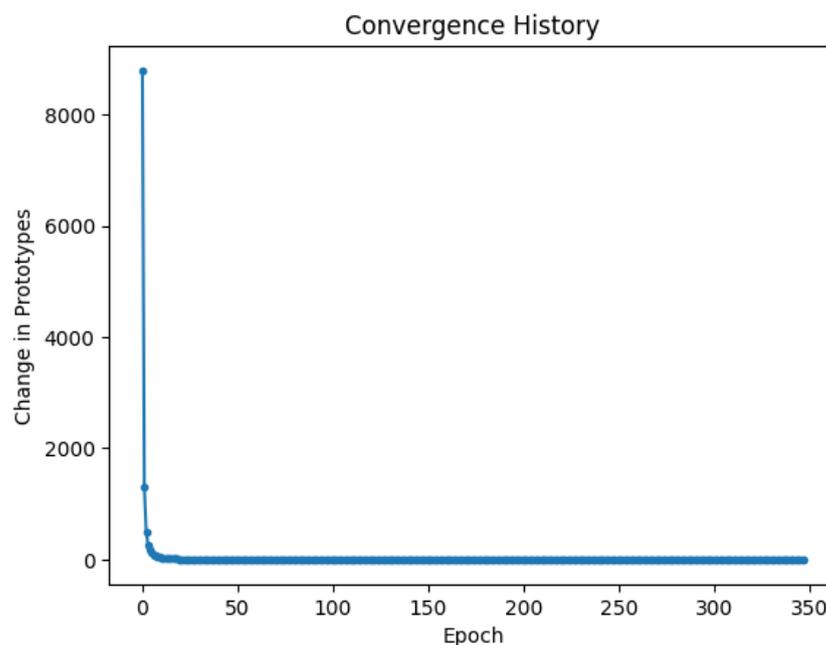
Gambar 4. 5 Grafik konvergensi dengan *learning rate* 0.03

Berdasarkan Gambar 4.4 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.4 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak 4900. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai 2200. Perubahan ini lebih kecil dibandingkan dengan

yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.03 dengan rasio data 60:40 berhenti pada epoch 8400.

4.2.2 Hasil Training Rasio Data 70:30

Hasil training menggunakan rasio data 70:30 dengan parameter learning rate 0.1, 0.2, 0.01, dan 0.03. Grafik konvergensi yang diperoleh jika menerapkan parameter learning rate 0.1 ditunjukkan pada Gambar 4.5.

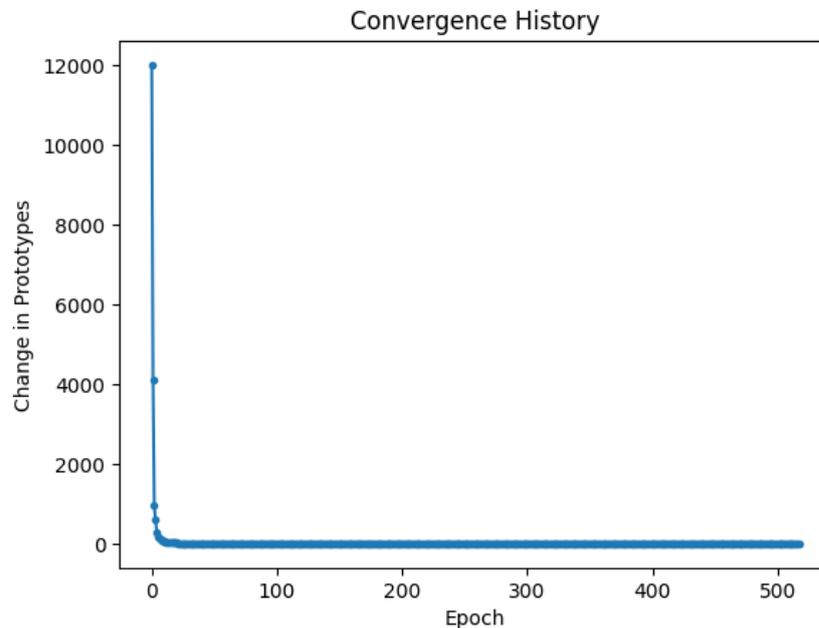


Gambar 4. 6 Grafik konvergensi dengan learning rate 0.1

Berdasarkan Gambar 4.5 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot

prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.5 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak diatas 8789.030257598984. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai diatas 1314.9744383207276. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.1 dengan rasio data 70:30 berhenti pada epoch 348.

Selanjutnya pengujian dilakukan dengan mengubah parameter learning rate menjadi 0.2 untuk ratio data 70:30. Hasil yang diperoleh ditunjukkan pada grafik konvergensi pada Gambar 4.7.

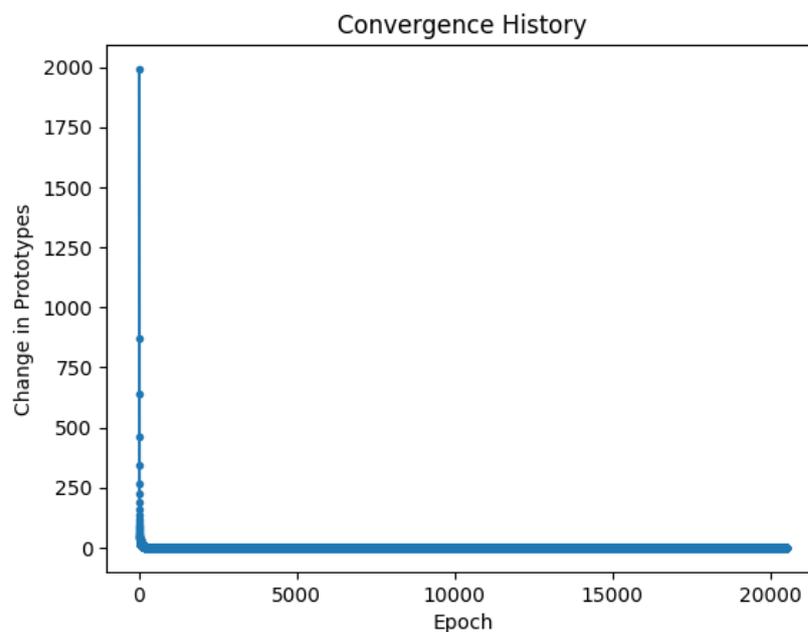


Gambar 4. 7 Grafik konvergensi learning rate 0.2

Berdasarkan Gambar 4.6 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.6 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak diatas 12010.265899166188. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai diatas 4106.678316504948. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut

hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.2 dengan rasio data 70:30 berhenti pada epoch 518.

Parameter learning rate selanjutnya yang digunakan sebagai pelatihan adalah 0.01. Hasil grafik konvergensi untuk parameter ini dapat dilihat pada Gambar 4.8.

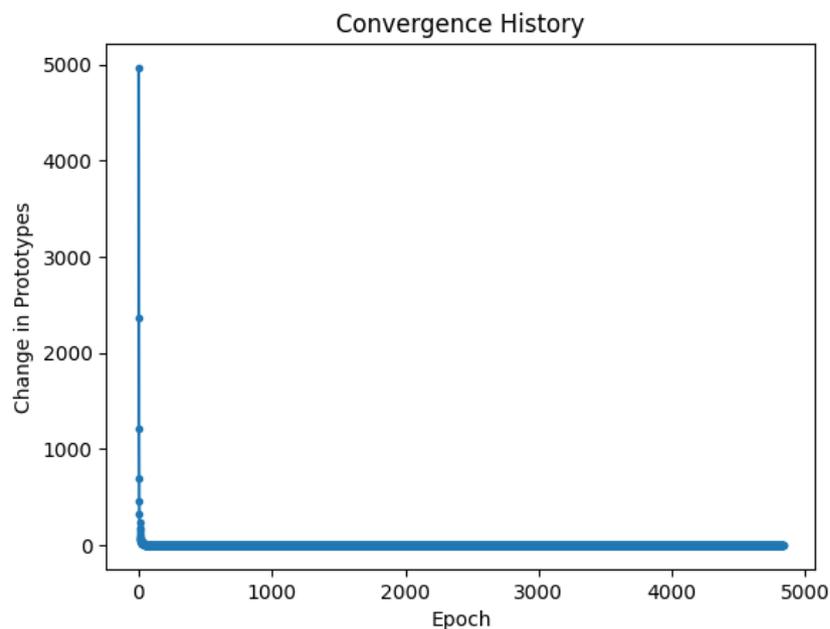


Gambar 4. 8 Grafik konvergensi dengan *learning rate* 0.01

Berdasarkan Gambar 4.7 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.7 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak diatas 2000. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2,

“Perubahan pada Prototipe” terdapat pada nilai diatas 900. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.01 dengan rasio data 70:30 berhenti pada epoch 20510.

Parameter learning rate terakhir yang digunakan untuk pelatihan pada rasio data 70:30 adalah 0.03. Hasil grafik konvergensi dari pengujian ini ditunjukkan pada Gambar 4.8.



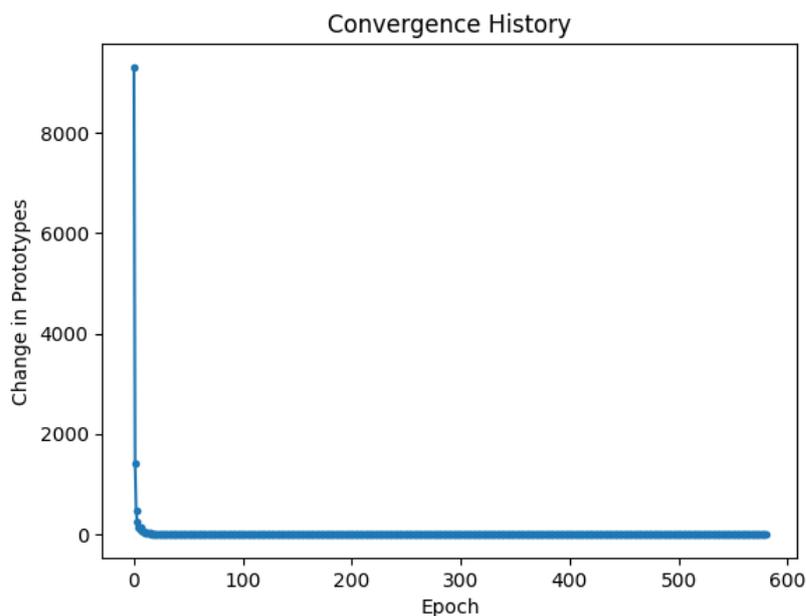
Gambar 4. 9 Grafik konvergensi dengan *learning rate* 0.03

Berdasarkan Gambar 4.8 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot

prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.8 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak 4967.140989841807. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai 2363.9022715757947. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.01 dengan rasio data 70:30 berhenti pada epoch 4835.

4.2.3 Hasil Training Rasio Data 80:20

Skenario ketiga dilakukan dengan menerapkan rasio data 80:20 terhadap learning rate 0.01, 0.2, 0.01, dan 0.03. Pelatihan menggunakan parameter learning rate 0.1 menghasilkan grafik konvergensi seperti pada Gambar 4.9.

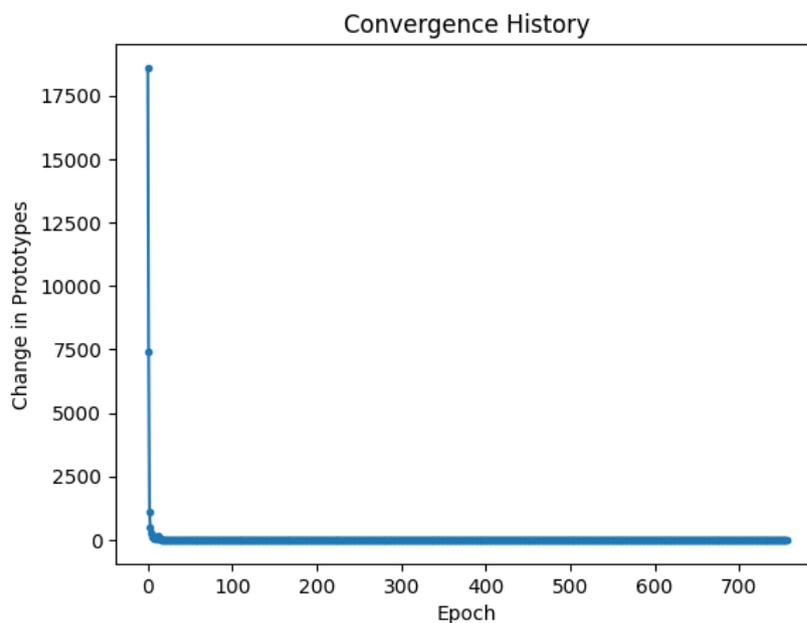


Gambar 4. 10 Grafik konvergensi dengan learning rate 0.1

Berdasarkan Gambar 4.9 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.9 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak 9313.247994879217. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai 1411.2409248643216. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga

mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.1 dengan rasio data 80:20 berhenti pada epoch 581.

Pelatihan selanjutnya dilakukan dengan mengubah parameter learning rate menjadi 0.2 untuk menguji hasil akurasi yang dihasilkan. Gambar 4.10 berikut menunjukkan grafik kovergensi dari pengujian yang dilakukan menggunakan parameter 0.2.

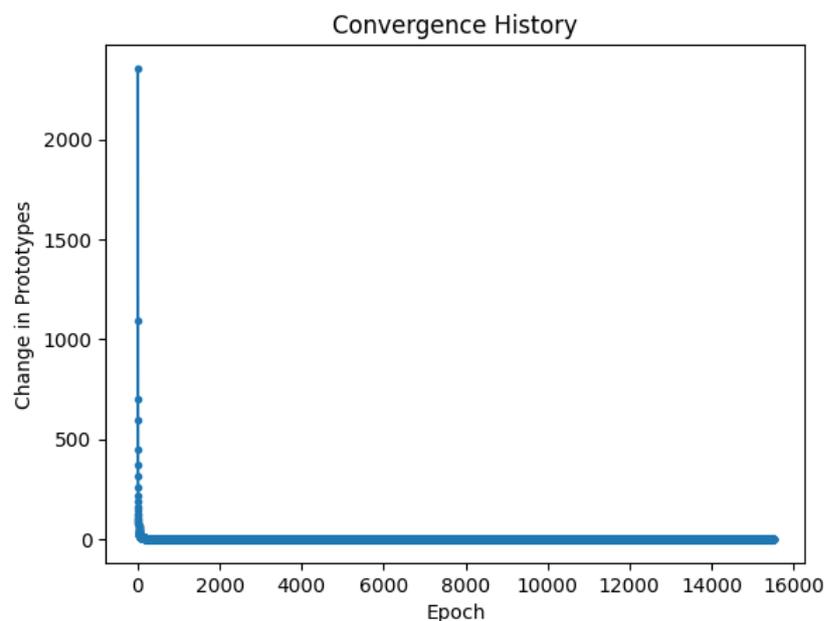


Gambar 4. 11 Grafik konvergensi dengan *learning rate* 0.2

Berdasarkan Gambar 4.10 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.10 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak 18591.242401352967. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ

menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai 7416.224558091433. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.2 dengan rasio data 80:20 berhenti pada epoch 757.

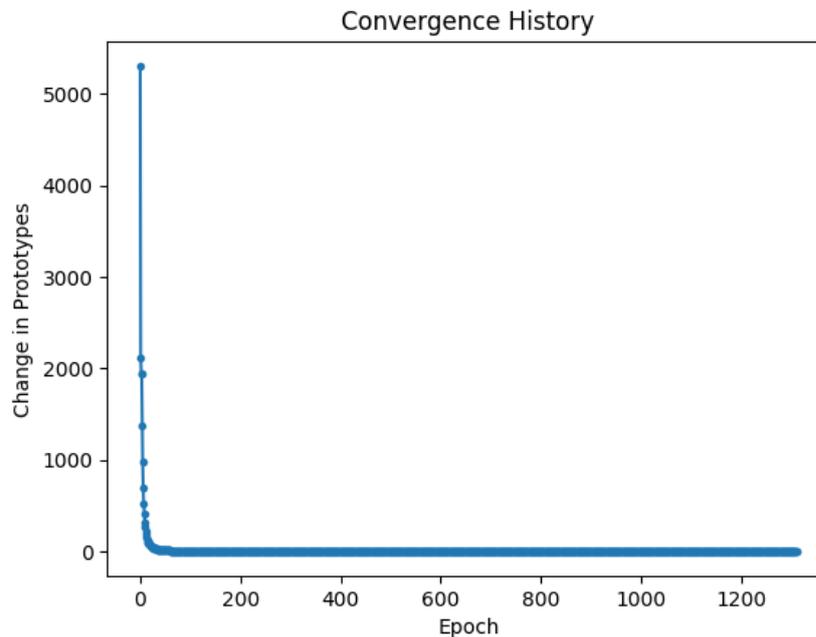
Pelatihan berikutnya dengan mengubah parameter learning rate 0.01 untuk mengetahui tingkat akurasi yang mampu dihasilkan apakah lebih baik daripada parameter learning rate sebelumnya. Gambar 4.11 berikut menunjukkan perubahan prototipe dalam bentuk grafik konvergensi.



Gambar 4. 12 Grafik konvergensi dengan *learning rate* 0.01

Berdasarkan Gambar 4.11 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.11 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak 2355.595898579644. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai 1094.1629439146488. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.01 dengan rasio data 80:20 berhenti pada epoch 15509.

Pelatihan berikutnya dengan mengubah parameter learning rate menjadi 0.03. Berikut ditunjukkan pada Gambar 4.12 grafik konvergensi dari perubahan prototipe yang terjadi pada proses pelatihan LVQ.



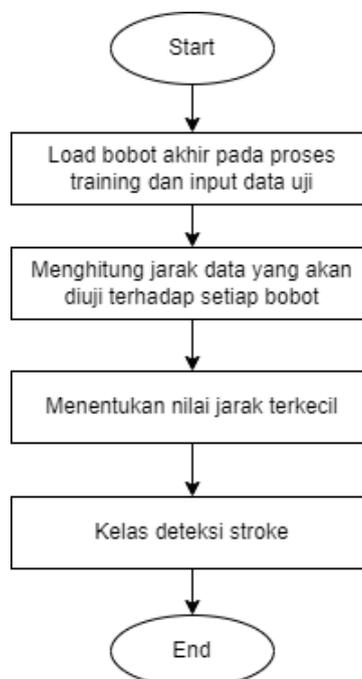
Gambar 4. 13 Grafik konvergensi dengan *learning rate* 0.03

Berdasarkan Gambar 4.12 dapat dilihat bahwa, bobot prototipe mengalami perubahan pada setiap epoch. Perubahan yang terjadi diukur dengan menggunakan “Perubahan Prototipe”, yang mana diukur berdasarkan perbedaan antara bobot prototipe pada epoch sebelumnya dan bobot prototipe saat ini. Pada Gambar 4.12 menunjukkan “Perubahan pada Prototipe” pada epoch 1 dimana nilai bobot terletak 5309.265141014205. Hal ini menunjukkan perubahan besar pada bobot prototipe antara epoch 1 dengan epoch sebelumnya yang mana menunjukkan model LVQ menyesuaikan bobot berdasarkan data pelatihan. Sedangkan pada epoch 2, “Perubahan pada Prototipe” terdapat pada nilai 2112.7329680474413. Perubahan ini lebih kecil dibandingkan dengan yang terjadi pada epoch 1, yang menunjukkan bahwa model LVQ semakin mendekati konvergensi. Semakin kecil perubahan bobot prototipe, semakin dekat model dengan konvergensi. “Perubahan pada Prototipe” akan berkurang pada setiap epoch ketika proses terus berlanjut hingga

mencapai tingkat konvergensi yang lebih stabil. Pada learning rate 0.03 dengan rasio data 80:20 berhenti pada epoch 1312.

4.3 Proses Testing

Bobot akhir yang telah diperoleh pada proses training kemudian digunakan sebagai input pada proses testing. Proses testing pada algoritma LVQ ini bertujuan untuk mengetahui apakah model yang telah dilatih dapat digunakan pada data yang belum pernah dilihat sebelumnya. Langkah-langkah yang dilakukan pada proses testing LVQ melibatkan input data uji, melakukan perhitungan jarak menggunakan dij sebagai jarak Euclidian, menentukan neuron pemenang berdasarkan jarak terkecil dengan data uji, dan melakukan pelabelan terhadap data uji sebagai hasil deteksi. Adapun alur proses testing algoritma LVQ dijabarkan pada Gambar 4.13 berikut.

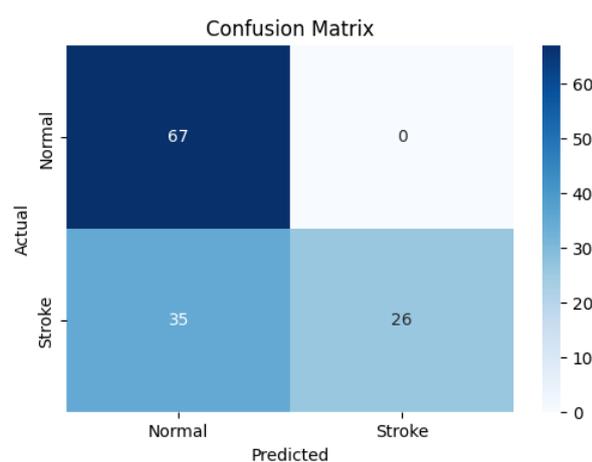


Gambar 4. 14 Flowchart proses testing algoritma LVQ

Salah satu bagian penting dalam deteksi stroke adalah pengukuran sistem. Tujuan dilakukannya proses ini adalah untuk mengumpulkan informasi penting seperti apakah pasien menderita stroke atau tidak berdasarkan pada data yang ada. Namun, tujuan lain dilakukannya pengukuran sistem ini adalah untuk menilai akurasi yang mampu dihasilkan oleh sistem. Sejauh mana sistem mampu mendeteksi kondisi stroke dengan tepat dan konsisten merupakan tolak ukur dari akurasi ini. Semakin tinggi nilai akurasi yang dihasilkan, semakin baik sistem dapat mendeteksi stroke dengan benar dan mengurangi salah diagnosis yang dapat terjadi.

4.3.1 Hasil Testing Rasio Data 60:40

Uji coba yang dilakukan pada penelitian ini menggunakan parameter learning rate 0.1, 0.2, 0.01, dan 0.03 dengan convergence threshold sebesar $1e-2$ dan patience sebesar 10. Hasil deteksi yang dilakukan pada pengujian dengan parameter 0.1 dapat dilihat pada Lampiran 1. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.14.



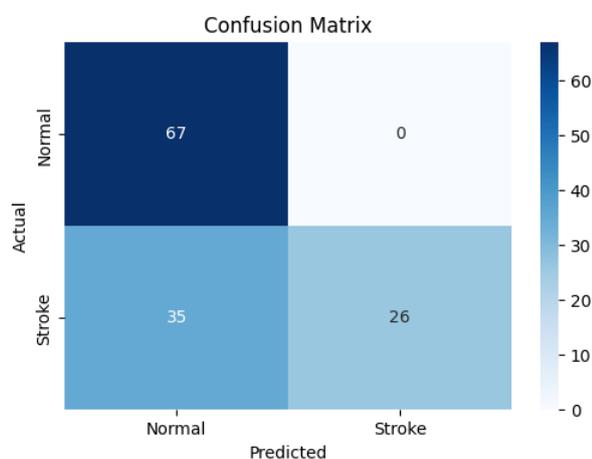
Gambar 4. 15 *Confusion matrix 60:40 learning rate 0.1*

Gambar 4.14 menunjukkan hasil pengujian menggunakan confusion matrix dengan rasio data 60:40 dan learning rate 0.1. Hasil yang diberikan menunjukkan terdapat 26 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 35 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 67 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediksikan sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil prediksi tersebut, diperoleh hasil akurasi sebesar 0.726 dengan menggunakan skenario 1. Tabel 4.5 berikut menunjukkan hasil confusion matrix untuk precision, recall, dan f1-score dari deteksi 2 kelas.

Tabel 4. 4 Nilai *precision*, *recall*, *f1-score* pada *learning rate* 0.1

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.66	1.00	0.79
Stroke	1.00	0.43	0.60

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.2 dapat dilihat pada Lampiran 1. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.15.



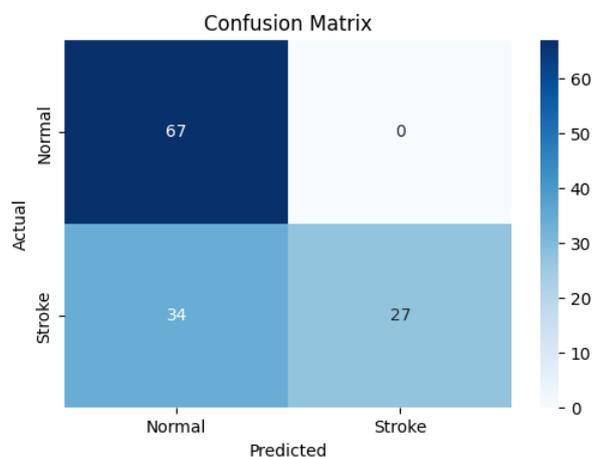
Gambar 4. 16 *Confusion matrix* 60:40 *learning rate* 0.2

Gambar 4.15 menunjukkan hasil deteksi menggunakan confusion matrix dengan learning rate sebesar 0.2 mampu menghasilkan akurasi sebanyak 0.726 yang menunjukkan terdapat 26 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 35 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 67 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediksi sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Hasil perhitungan aauntuk precision, recall, dan f1-score untuk masing-masing kelas dapat ditunjukkan pada Tabel 4.6 berikut.

Tabel 4. 5 Nilai *Precision*, *Recall*, dan *F1-Score* learning rate 0.2

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.66	1.00	0.79
Stroke	1.00	0.43	0.60

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.01 dapat dilihat pada Lampiran 1. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.16.



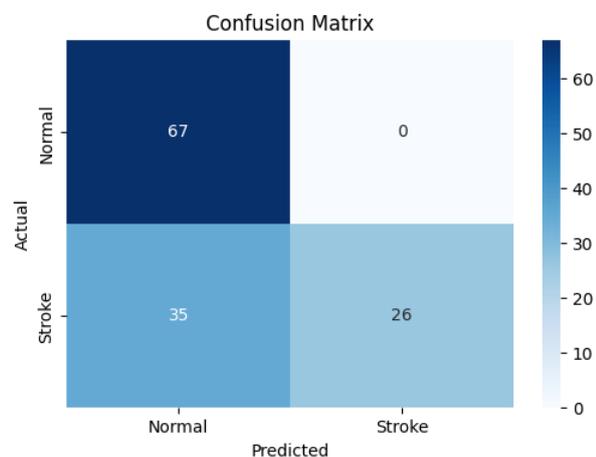
Gambar 4. 17 *Confusion matrix* 60:40 learning rate 0.01

Berdasarkan Gambar 4.16 mampu menghasilkan akurasi sebesar 0.734 dengan hasil yang diberikan menunjukkan terdapat 27 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 34 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 67 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediksikan sebagai data normal. Selain itu terdapat 1 data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil tersebut, diperoleh nilai precision, recall, dan f1-score jika menggunakan parameter learning rate sebesar 0.01 yang dapat ditunjukkan pada Tabel 4.7.

Tabel 4. 6 Nilai *Precision*, *Recall*, dan *F1-Score* learning rate 0.01

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.66	1.00	0.80
Stroke	1.00	0.44	0.61

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.03 dapat dilihat pada Lampiran 1. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.17.



Gambar 4. 18 *Confusion Matrix* 60:40 learning rate 0.03

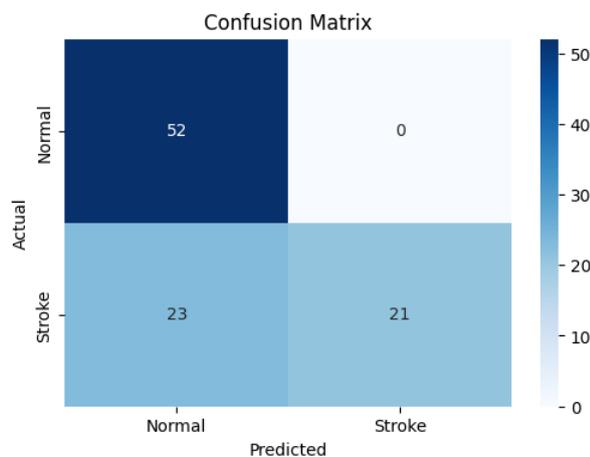
Nilai akurasi yang diperoleh menggunakan parameter ini sebesar 0.726. Hasil yang diberikan menunjukkan terdapat 26 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 35 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 67 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediski sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan Gambar 4.17 diperoleh nilai precision, recall, f1-score yang dapat ditunjukkan pada Tabel 4.8 berikut.

Tabel 4. 7 Nilai *Precision*, *Recall*, dan *F1-Score learning rate 0.03*

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.66	1.00	0.79
Stroke	1.00	0.43	0.60

4.3.2 Hasil Testing Rasio Data 70:30

Hasil deteksi yang dilakukan pada pengujian dengan parameter 0.1, 0.2, 0.01, dan 0.03 dapat dilihat pada Lampiran 2. Adapun hasil dari pengujian tersebut menggunakan confusion matrix untuk parameter learning rate 0.1 ditunjukkan seperti pada Gambar 4.18.



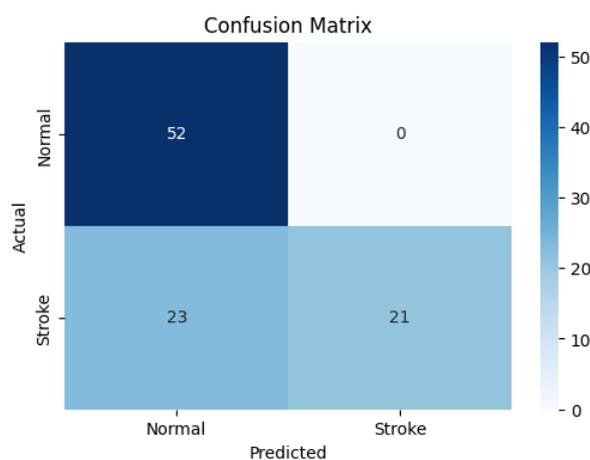
Gambar 4. 19 *Confusion matrix 70:30 learning rate 0.1*

Gambar 4.18 menunjukkan terdapat 21 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 23 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 52 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediksi sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil confusion matrix tersebut diperoleh hasil akurasi sebesar 0.760 dengan nilai precision, recall, dan f1-score untuk masing-masing kelas seperti ditunjukkan pada Tabel 4.9.

Tabel 4. 8 Nilai *precision*, *recall*, dan *f1-score* learning rate 0.1

Kelas	Precision	Recall	F1-Score
Normal	0.69	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil deteksi yang dilakukan pada pengujian dengan parameter 0.2 dapat dilihat pada Lampiran 2. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.19.



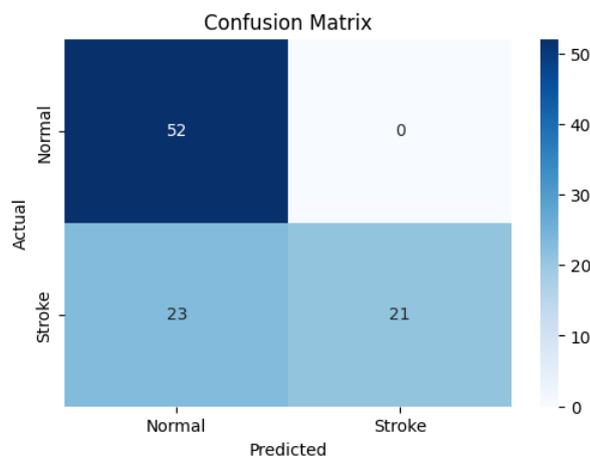
Gambar 4. 20 *Confusion matrix 70:30 learning rate 0.2*

Berdasarkan hasil confusion matrix pada Gambar 4.19 menunjukkan terdapat 21 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 23 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 52 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediski sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil prediksi tersebut, diperoleh hasil akurasi sebesar 0.760. Nilai precision, recall, dan f1-score untuk masing-masing kelas pengujian learning rate 0.2 ditunjukkan pada Tabel 4.10.

Tabel 4. 9 Nilai *precision*, *recall*, dan *f1-score learning rate 0.2*

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.69	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.01 dapat dilihat pada Lampiran 2. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.20.



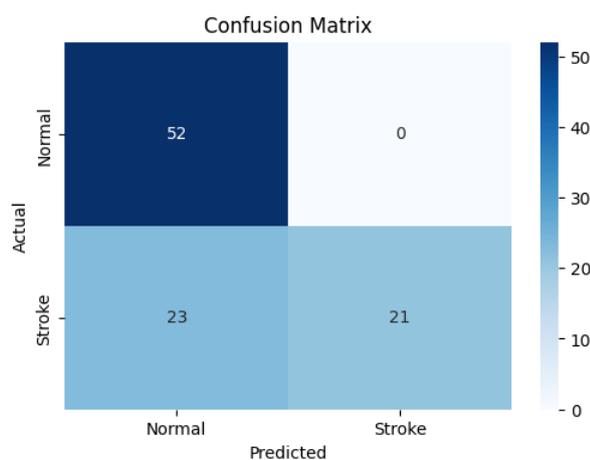
Gambar 4. 21 *Confusion matrix 70:30 learning rate 0.01*

Hasil confusion matrix yang diperoleh dari perubahan learning rate 0.01 menunjukkan terdapat 21 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 23 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 52 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediski sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil tersebut, diperoleh hasil akurasi sebesar 0.760. Nilai untuk precision, recall, dan f1-score berdasarkan hasil tersebut ditunjukkan pada Tabel 4.11.

Tabel 4. 10 Nilai *precision*, *recall*, dan *f1-score learning rate 0.01*

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.74	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.03 dapat dilihat pada Lampiran 2. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.21.



Gambar 4. 22 Confusion matrix 70:30 learning rate 0.03

Berdasarkan Gambar 4.21 diperoleh hasil akurasi sebesar 0.760. Hasil akurasi tersebut diperoleh berdasarkan hasil confusion matrix yang menunjukkan terdapat 21 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 23 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 52 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediski sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Selain itu, nilai precision, recall, dan f1-score diperoleh berdasarkan hasil confusion matrix tersebut yang dapat ditunjukkan pada Tabel 4.12.

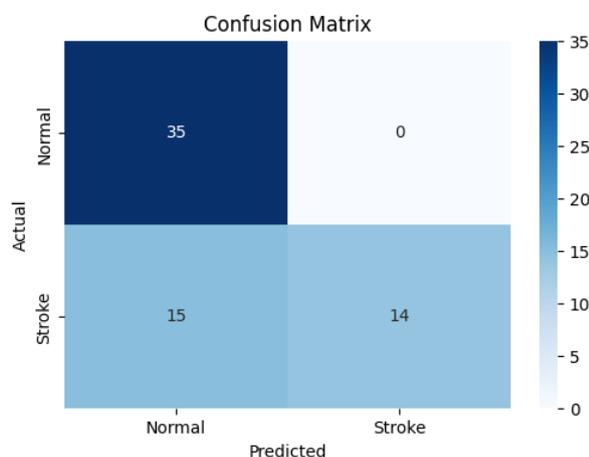
Tabel 4. 11 Nilai *Precision*, *recall*, dan *f1-score* learning rate 0.03

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.69	1.00	0.82
Stroke	1.00	0.48	0.65

4.3.3 Hasil Testing Rasio Data 80:20

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.1, 0.2, 0.01, dan 0.03 dapat dilihat pada Lampiran 3. Adapun hasil dari

pengujian parameter learning rate 0.1 menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.22.



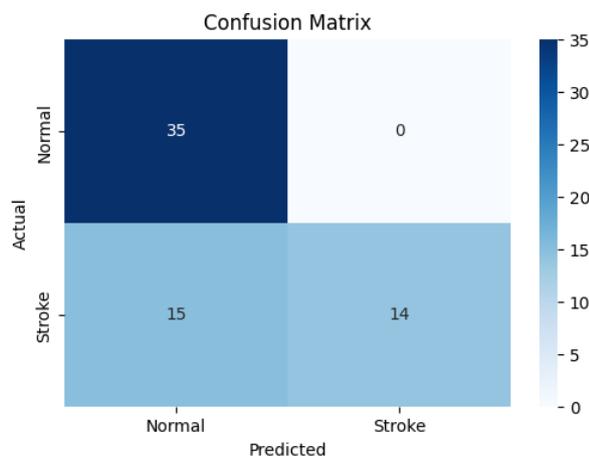
Gambar 4. 23 Confusion matrix 80:20 learning rate 0.1

Berdasarkan Gambar 4.22 yang menunjukkan hasil confusion matrix yang terdapat 14 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 15 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 35 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediski sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil tersebut, diperoleh hasil akurasi sebesar 0.765. Selain itu diperoleh nilai precision, recall, dan f1-score seperti pada Tabel 4.13.

Tabel 4. 12 Nilai *precision*, *recall* dan *f1-score* learning rate 0.1.

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.70	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.2 dapat dilihat pada Lampiran 3. Adapun hasil dari pengujian tersebut menggunakan confusion matrix ditunjukkan seperti pada Gambar 4.23.



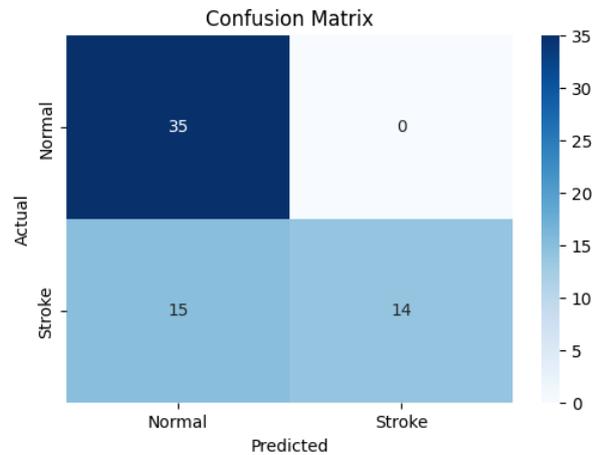
Gambar 4. 24 *Confusion matrix 80:20 learning rate 0.2*

Berdasarkan hasil confusion matrix yang telah dilakukan, diperoleh tingkat akurasi sebesar 0.765 yang mana hasil akurasi ini lebih rendah dibandingkan dengan menggunakan learning rate sebelumnya. Hasil yang diberikan menunjukkan terdapat 14 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 15 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 35 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil dipredisksi sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Adapun nilai precision, recall, dan f1-score yang dihasilkan pada masing-masing kelas ditunjukkan pada Tabel 4.14.

Tabel 4. 13 Nilai *precision*, *recall*, dan *f1-score* learning rate 0.2

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.70	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil deteksi yang dilakukan pada pengujian dengan parameter 0.01 dapat dilihat pada Lampiran 3. Perhitungan menggunakan learning rate 0.01 menghasilkan jumlah citra yang mampu dideteksi dengan baik pada Gambar 4.24.



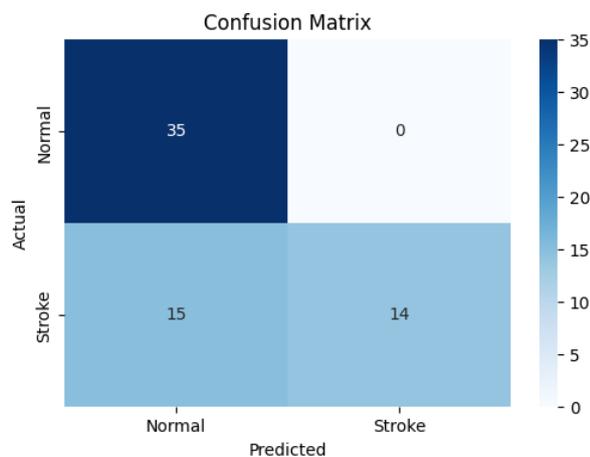
Gambar 4. 25 Confusion matrix 80:20 learning rate 0.01

Gambar 4.24 menunjukkan jumlah data yang mampu dideteksi oleh confusion matrix yang terdapat 14 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 15 data FN yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 35 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediksi sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil prediksi tersebut, diperoleh hasil akurasi sebesar 0.765. Berdasarkan hasil tersebut, kemudian diperoleh nilai untuk precision, recall, dan f1-score dari masing-masing label kelas seperti pada Tabel 4.15.

Tabel 4. 14 Nilai *Precision*, *recall*, dan *f1-score* learning rate 0.01

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.70	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil deteksi yang dilakukan pada pengujian dengan parameter learning rate 0.03 dapat dilihat pada Lampiran 3. Jumlah data yang mampu dideteksi oleh confusion matrix ditunjukkan pada Gambar 4.25.



Gambar 4. 26 Confusion matrix 80:20 learning rate 0.03

Gambar 4.25 menunjukkan hasil confusion matrix berupa 14 data TP (True Positive) yang mampu diprediksi dengan benar oleh sistem. Terdapat 15 data FN (False Negative) yaitu data stroke yang salah diprediksi sebagai normal. Sebaliknya terdapat 35 data TN (True Negative) yaitu data yang tidak memiliki stroke dan berhasil diprediksikan sebagai data normal. Selain itu tidak ada data yang salah diprediksi sebagai FP (False Positive). Berdasarkan hasil prediksi tersebut, diperoleh hasil akurasi sebesar 0.765. Tabel 4.16 berikut menunjukkan nilai precision, recall, dan f1-score yang diperoleh terhadap masing-masing kelas.

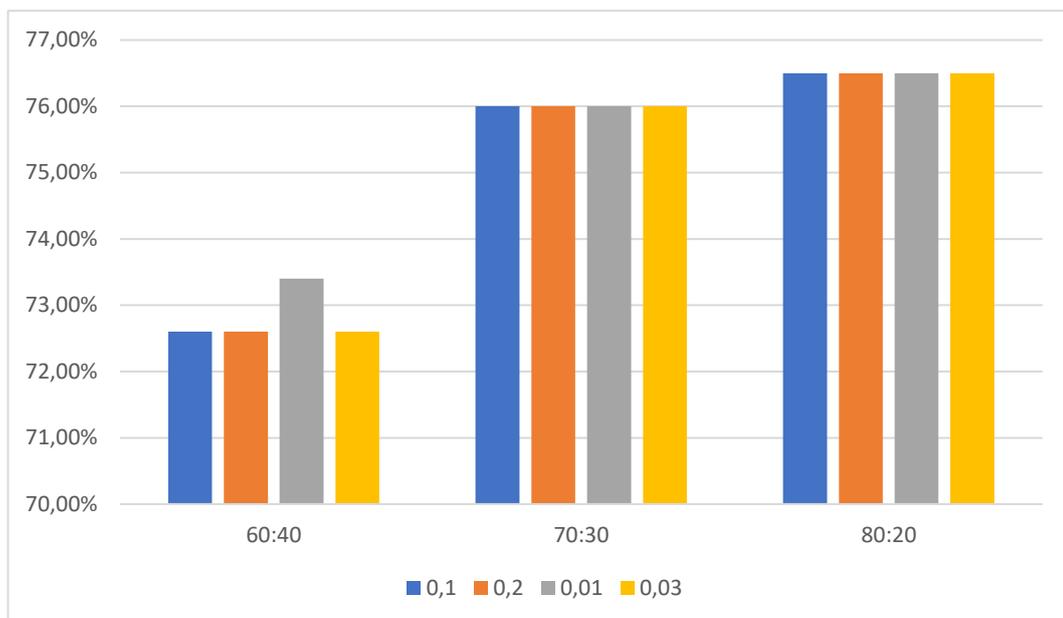
Tabel 4. 15 Nilai *Precision*, *recall*, dan *f1-score* learning rate 0.03

Kelas	<i>Precision</i>	<i>Recall</i>	<i>F1-Score</i>
Normal	0.70	1.00	0.82
Stroke	1.00	0.48	0.65

Hasil uji coba menggunakan ketiga rasio data yaitu 60:40, 70:30, dan 80:20 serta perubahan parameter learning rate 0.1, 0.2, 0.01, dan 0.03 pada epoch yang digunakan yaitu 50 mampu menghasilkan tingkat akurasi seperti pada Tabel 4.16.

Tabel 4. 16 Hasil Pengujian Parameter

Rasio Data	Learning Rate	Akurasi
60:40	0.1	72.6 %
	0.2	72.6 %
	0.01	73.4 %
	0.03	72.6 %
70:30	0.1	76.0 %
	0.2	76.0 %
	0.01	76.0 %
	0.03	76.0 %
80:20	0.1	76.5 %
	0.2	76.5 %
	0.01	76.5 %
	0.03	76.5 %



Gambar 4. 27 Perbandingan hasil pengujian parameter

Berdasarkan Tabel 4.17 diatas, akurasi terbaik yang mampu dihasilkan terdapat pada rasio data 80:20 dengan hasil akurasi yang cenderung stabil. Akurasi terbaik dihasilkan ketika menggunakan ratio data 80:20 dengan akurasi sebesar 76.5% untuk keempat parameter learning rate yang digunakan. Pada pengujian rasio data 60:40 mampu menghasilkan akurasi rata-rata sebesar 72.8%, rasio data

70:30 mampu menghasilkan akurasi rata-rata terbaik sebesar 76%. Sedangkan pada pengujian ratio data 80:20, diperoleh akurasi rata-rata sebesar 76.5%. Hasil akurasi yang dihasilkan cenderung stabil dan tidak terdapat perubahan yang cukup signifikan pada akurasi model. Hasil pengujian parameter pada Tabel 4.17 dapat divisualisasikan seperti pada Gambar 4.25.

4.4 Pembahasan

Berdasarkan hasil uji coba yang dilakukan dengan menggunakan 3 jenis pembagian rasio data dan juga 4 jenis learning rate yang berbeda mampu menghasilkan akurasi terbaik pada rasio data 80:20 mampu menghasilkan akurasi tertinggi di setiap tingkat pembelajaran yang telah diuji. Akurasi yang dihasilkan mencapai 76.5% menggunakan keempat parameter learning rate yang digunakan sebagai uji coba. Hal ini menunjukkan bahwa pengalokasian data training yang lebih besar mampu memberikan hasil yang lebih optimal. Sedangkan penggunaan learning rate pada penelitian ini tidak berpengaruh secara signifikan untuk memberikan perbedaan pada akurasi yang dihasilkan. Hal ini menunjukkan bahwa model LVQ cenderung stabil dan tidak terpengaruh oleh perubahan rasio data dan variasi learning rate yang digunakan sebagai pengujian. Adapun dataset yang digunakan pada penelitian ini memiliki jumlah yang sama di masing-masing kelas. Sehingga menghindari terjadinya oversampling dan undersampling karena jumlah data yang tidak seimbang untuk pelatihannya. Penentuan penggunaan parameter untuk model ini memiliki pertimbangan yang bertujuan untuk membandingkan parameter yang paling optimal yang mampu diterapkan pada model LVQ dengan menggunakan epoch yang sudah ditetapkan dan jumlah data sekian.

Berdasarkan hasil uji coba menggunakan beberapa skenario yang telah diterapkan, sistem mampu mencapai hasil akurasi paling optimal dengan menggunakan rasio data 80:20 pada learning rate 0.1, 0.2, 0.01, dan 0.03 yang mana mampu menghasilkan akurasi stabil sebesar 76.5%. Pada rasio data 80:20 mampu menghasilkan precision, recall, dan f1-score sebesar 85%, 74%, dan 74%.

Tabel 4.18 menunjukkan perbandingan penelitian antara penelitian terdahulu dengan penelitian ini yang berjudul Deteksi Stroke Berdasarkan Citra CT SCAN dengan menggunakan metode LVQ. Perbandingan dilakukan berdasarkan hasil akurasi dari metode yang digunakan pada setiap penelitian.

Tabel 4. 17 Perbandingan penelitian

No.	Peneliti	Metode	Akurasi
1.	(Tazin et al., 2021)	Random Forest, Logistic Regression, Decision Tree, Voting Classifier	96% RF.
2.	(Hokkinen et al., 2021)	CNN	1.00 sensitivity dan 0.94 specificity.
3.	(Rachman, 2021)	Case Based Reasoning, Probabilistic Symmetric, dan KNN	66.66%
4.	(Z & N, 2023)	KNN	93.54%
5.	(Peng et al., 2020)	Artificial Neural Network (ANN) + Levenberg Marquardt dan Scaled Conjugate Gradient	98%
6.	(Subudhi et al., 2020)	Expectation-Maximization (EM) + Fractional-Order Darwinian Particle Swarm Optimization (FODPSO) + RF	93.4%
7.	(Hendriyani, 2020)	Perbandingan LVQ dan Bckpropagation	91.43% Backpropagation, 65.71% LVQ
8.	(Santony et al., 2020)	Minutiae Extraction + LVQ	83.34%
9.	(Nurpadillah et al., 2019)	LVQ	66.66%
10.	(Zheng & Xie, 2018)	CNN + PCA + LVQ	94.62%
	Metode yang diusulkan	LVQ	76.5%

Berdasarkan Tabel 4.18, hasil akurasi yang diperoleh pada penelitian ini cenderung lebih rendah dibandingkan dengan penelitian terdahulu yang menggunakan metode lebih kompleks seperti CNN. Selain itu, variasi data yang digunakan juga berpengaruh pada hasil akurasi yang dihasilkan. Pada penelitian terdahulu juga melakukan penambahan metode untuk menghasilkan variasi dalam fitur-fitur agar lebih relevan sebelum dilakukannya proses deteksi stroke.

Tujuan dilakukannya penelitian ini yaitu untuk mendeteksi stroke melalui CT Scan berdasarkan dua kelas yaitu Stroke dan Normal. Penelitian ini dilakukan agar penyakit stroke yang diderita pasien dapat dilakukan deteksi cepat untuk melakukan pengobatan agar nyawa pasien dapat segera diselamatkan dan mencegah kerusakan lain yang kemungkinan akan ditimbulkan. Sistem yang dibangun masih memerlukan beberapa perbaikan sehingga performa yang dihasilkan dapat lebih optimal.

Sistem pendeteksi penyakit stroke ini dapat digunakan dalam bidang teknologi medis sebagai sarana untuk mencegah dan mengobati penyakit stroke. Sehingga penyakit yang diderita mampu dideteksi dengan lebih cepat jika manusia mengalami gejala yang mengarah kepada penyakit stroke atau penyakit lainnya dan bisa mendapatkan pertolongan dengan lebih cepat.

Model deteksi yang digunakan pada penelitian terkait deteksi stroke ini menggunakan metode Learning Vector Quantization dimana metode ini merupakan salah satu metode jaringan syaraf tiruan dimana penerapannya dengan mempelajari sistem kerja syaraf manusia. LVQ dianggap efektif untuk mengklasifikasikan data

berdasarkan kemiripan dan memahami pola seperti yang dilakukan oleh syaraf manusia. Model ini memiliki kemampuan untuk mengenali pola dan melakukan penyesuaian terhadap perubahan dataset. Sehingga, hasil evaluasi menunjukkan bahwa LVQ memiliki tingkat konsistensi yang baik dalam memberikan akurasi menggunakan berbagai skenario rasio data dan variasi learning rate yang diujikan. Hal ini menunjukkan bagaimana kuasa Allah dalam menciptakan sesuatu dengan berbagai ukuran.

وَلَقَدْ خَلَقْنَا الْإِنْسَانَ مِنْ سُلَالَةٍ مِنْ طِينٍ . ثُمَّ جَعَلْنَاهُ نُطْفَةً فِي قَرَارٍ مَكِينٍ . ثُمَّ خَلَقْنَا النُّطْفَةَ عَلَاقَةً فَخَلَقْنَا الْعَلَقَةَ مُضْغَةً فَخَلَقْنَا الْمُضْغَةَ عِظَامًا فَكَسَوْنَا الْعِظَامَ لَحْمًا ثُمَّ أَنْشَأْنَاهُ خَلْقًا آخَرَ ۖ فَتَبَارَكَ اللَّهُ أَحْسَنُ الْخَالِقِينَ .

“Dan sesungguhnya Kami telah menciptakan manusia dari suatu saripati (berasal) dari tanah. (12). Kemudian Kami jadikan saripati itu air mani (yang disimpan) dalam tempat yang kokoh (rahim). (13). Kemudian air mani itu Kami jadikan segumpal darah, lalu segumpal darah itu Kami jadikan segumpal daging, dan segumpal daging itu Kami jadikan tulang belulang, lalu tulang belulang itu Kami bungkus dengan daging. Kemudian Kami jadikan dia makhluk yang (berbentuk) lain. Maka Maha sucilah Allah, Pencipta Yang Paling Baik. (14).” (Q.S Al-Mu'minin: 12-14)

Menurut Tafsir Jalalain terkait Q.S Al-Mu'minin Ayat 12 dijelaskan bahwa Allah SWT menciptakan manusia, yang berasal dari Adam, dengan menggunakan air mani yang ada dalam rahim untuk menghasilkan segumpal darah, segumpal daging, dan akhirnya tulang-belulang yang terbungkus daging. Pada tafsir tersebut menjelaskan tentang proses penciptaan manusia yang melibatkan tahapan pembentukan segumpal darah, segumpal daging, dan tulang-belulang yang dilapisi dengan daging.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Berdasarkan penelitian yang dilakukan terhadap deteksi penyakit stroke menunjukkan bahwa pengalokasian data pelatihan yang lebih besar mampu memberikan hasil yang lebih optimal. Adapun dataset yang digunakan pada penelitian ini memiliki jumlah yang sama pada masing-masing kelas. Skenario pengujian yang dilakukan dengan menggunakan 3 rasio data yaitu 60:40, 70:30, dan 80:20. Ketiga rasio data yang digunakan mampu memberikan hasil akurasi yang berbeda. Selain itu, pada penelitian ini dilakukan pengujian juga terhadap parameter *learning rate* yang digunakan, 0.1, 0.2, 0.01, dan 0.03. Berdasarkan hasil uji coba yang dilakukan terhadap rasio data dan *learning rate* diperoleh hasil akurasi paling optimal pada rasio data 80:20 dengan keempat *learning rate* memberikan hasil akurasi yang sama sebesar 76.5%.

5.2 Saran

Setelah melakukan uji coba, peneliti menyadari bahwa masih terdapat kekurangan pada penelitian ini. Sehingga perlu dilakukan perbaikan agar dapat meningkatkan performa model. Berikut saran untuk peneliti selanjutnya :

1. Penggunaan dataset yang lebih bervariasi. Penggunaan dataset yang kecil dan tidak bervariasi mampu mempengaruhi akurasi yang dihasilkan.
2. Penambahan skenario untuk pengujian menggunakan parameter lain agar mampu membantu melakukan deteksi dengan lebih baik.

3. Penggunaan metode lain dapat dikombinasikan dengan metode LVQ yang saat ini digunakan sehingga dapat meningkatkan performa deteksi sehingga hasil yang diperoleh juga semakin baik.

DAFTAR PUSTAKA

- Apriani, P., Rusdianto, I., & Ristian, U. (2020). Aplikasi Deteksi Objek Bergerak Berbasis Citra Dengan Metode Background Substraction dan Blob Detection (Studi Kasus: Mami Mart Kubu Raya). *Coding: Jurnal Komputer Dan Aplikasi*, 08(01), 132–141.
- Arifianto, A. S., Sarosa, M., & Setyawati, O. (2014). Klasifikasi Stroke Berdasarkan Kelainan Patologis dengan Learning Vector Quantization. *Jurnal EECCIS*, 8(2), 117–122.
- Dewi, E. M., Purwanti, E., & Apsari, R. (2019). Cervical cell classification using Learning Vector Quantization (LVQ) based on shape and statistical features. *International Journal of Online and Biomedical Engineering*, 15(2), 91–98. <https://doi.org/10.3991/ijoe.v15i02.9796>
- Eshmawi, A. A., Khayyat, M., Algarni, A. D., & Hilali-Jaghdam, I. (2022). An Ensemble of Deep Learning Enabled Brain Stroke Classification Model in Magnetic Resonance Images. *Journal of Healthcare Engineering*, 2022. <https://doi.org/10.1155/2022/7815434>
- Hariri, F. R. (2018). Klasifikasi Jenis Golongan Darah Menggunakan Fuzzy C-Means Clustering (FCM) dan Learning Vector Quantization (LVQ). *MATICS*, 10(1), 26. <https://doi.org/10.18860/mat.v10i1.5356>
- Hendriyani, Y. (2020). Perbandingan Algoritma Backpropagation Dan Learning Vector Quantization (LVQ) dalam Pengenalan Pola Bangun Ruang Geometri. *INVOTEK: Jurnal Inovasi Vokasional Dan Teknologi*, 20(2), 59–66. <https://doi.org/10.24036/invotek.v20i2.746>
- 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). <https://doi.org/10.1186/s41747-021-00225-1>
- Kusumadewi, S. (2004). *Membangun Jaringan Syaraf Tiruan Menggunakan MATLAB & EXCEL LINK*. Graha Ilmu.
- Li, K., Wan, G., Cheng, G., Meng, L., & Han, J. (2020). Object detection in optical remote sensing images: A survey and a new benchmark. *ISPRS Journal of Photogrammetry and Remote Sensing*, 159, 296–307. <https://doi.org/10.1016/j.isprsjprs.2019.11.023>
- Nurpadillah, D. I., Haviluddin, Pakpahan, H. S., Islamiyah, & Setyadi, H. J. (2019). Pengenalan Karakter Tulisan Menggunakan Metode Learning Vector

Quantization. *Sains, Aplikasi, Komputasi Dan Teknologi Informasi*, 1(2), 23–29.

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*, 130–133.

Pratomo, A. H., Kaswidjanti, W., & Mu'arifah, S. (2020). Implementasi Algoritma Region Of Interest (ROI) Untuk Meningkatkan Performa Algoritma Deteksi Dan Klasifikasi Kendaraan. *Jurnal Teknologi Informasi Dan Ilmu Komputer (JTIK)*, 7(1), 155–162. <https://doi.org/10.25126/jtiik.202071718>

Rachman, R. (2021). Implementasi Case Based Reasoning Mendiagnosa Penyakit Stroke Menggunakan Algoritma Probabilistic Symmetric. *JURNAL INFORMATIKA*, 8(1), 10–16. <http://ejournal.bsi.ac.id/ejurnal/index.php/ji>

Rahmat, R. F., Pulungan, A. F., Faza, S., & Budiarto, R. (2017). Image Classification of Ribbed Smoked Sheet using Learning Vector Quantization. *Journal of Physics: Conference Series*, 801(1). <https://doi.org/10.1088/1742-6596/801/1/012050>

Santony, J., Devita, R., Hadi, A. F., & Zain, R. H. (2020). Peningkatan Kualitas Citra Sidik Jari Dengan Algoritma Minutiae Extraction dan Learning Vector Quantization. *Jurnal Teknologi Informasi Dan Pendidikan*, 13(1), 54–60.

Sari, L. M., Yuliano, A., & Almudriki. (2019). Hubungan Pengetahuan dan Sikap Keluarga Terhadap Kemampuan Deteksi Dini Serangan Stroke Iskemik Akut Pada Penanganan Pre Hospital. *Jurnal Kesehatan Perintis (Perintis's Health Journal)*, 6(1), 74–80.

Setyowati, E., & Mariani, S. (2021). Penerapan Jaringan Syaraf Tiruan dengan Metode Learning Vector Quantization (LVQ) Untuk Klasifikasi Penyakit Infeksi Saluran Pernapasan Akut (ISPA). *PRISMA, Prosiding Seminar Nasional Matematika*, 4, 514–523. <https://journal.unnes.ac.id/sju/index.php/prisma/>

Shaban, W. M., Morsy, M., & ELdin Abo, M. A. (2018). Preprocessing of Brain Hemorrhage Using CT scan. *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, 4(1), 397–404. www.ijraset.com

Subudhi, A., Dash, M., & Sabut, S. (2020). Automated segmentation and classification of brain stroke using expectation-maximization and random forest classifier. *Biocybernetics and Biomedical Engineering*, 40(1), 277–289. <https://doi.org/10.1016/j.bbe.2019.04.004>

- Sundararajan, D. (2017). *Digital Image Processing A Signal Processing and Algorithmic Approach*. <https://doi.org/10.1007/978-981-10-6113-4>
- Tazin, T., Alam, M. N., Dola, N. N., Bari, M. S., Bourouis, S., & Monirujjaman Khan, M. (2021a). Stroke Disease Detection and Prediction Using Robust Learning Approaches. *Journal of Healthcare Engineering*, 2021. <https://doi.org/10.1155/2021/7633381>
- Tazin, T., Alam, M. N., Dola, N. N., Bari, M. S., Bourouis, S., & Monirujjaman Khan, M. (2021b). Stroke Disease Detection and Prediction Using Robust Learning Approaches. *Journal of Healthcare Engineering*, 2021. <https://doi.org/10.1155/2021/7633381>
- Widiyanto, D. (2020). Tinjauan Algoritma RoI (Region of Interest) dengan Metode Pengembangan Otsu dan Klasterisasi K-Mean; Hasil dan Tantangannya. *JURNAL INFORMATIK*, 16(2), 75–86.
- Yu, C., Qi, X., Ma, H., He, X., Wang, C., & Zhao, Y. (2020). LLR: Learning learning rates by LSTM for training neural networks. *Neurocomputing*, 394, 41–50. <https://doi.org/10.1016/j.neucom.2020.01.106>
- Z, Z., & N, Q. (2023). Klasifikasi Penyakit Stroke Menggunakan Algoritma K-Nearest Neighbor (KNN). *ROUTERS: Jurnal Sistem Dan Teknologi Informasi*, 1(1), 1–8. <https://doi.org/10.xxxxx>
- Zheng, Q., & Xie, X. (2018). Traffic Sign Recognition Based on Learning Vector Quantization and Convolution Neural Network. *ACM International Conference Proceeding Series*, 178–183. <https://doi.org/10.1145/3232116.3232145>

LAMPIRAN

Lampiran 1 Tabel Hasil deteksi stroke rasio 60:40

No.	<i>Learning Rate</i>							
	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
1	1	1	1	1	1	1	1	1
2	1	0	1	0	1	1	1	0
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	1	0	1	0	1	0	1	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	1	0	1	0	1	0	1	0
20	1	0	1	0	1	0	1	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	1
25	1	0	1	0	1	0	1	0
26	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	1	1	1	1	1	1	1	1
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	1	0	1	0	1	0	1	0
34	0	0	0	0	0	0	0	0
35	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1
37	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
41	1	0	1	0	1	0	1	0

<i>Learning Rate</i>								
No.	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
42	0	0	0	0	0	0	0	0
43	1	0	1	0	1	0	1	0
44	1	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1	1
46	1	1	1	1	1	1	1	1
47	1	0	1	0	1	0	1	0
48	1	1	1	1	1	1	1	1
49	1	0	1	0	1	0	1	0
50	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0
52	1	0	1	0	1	0	1	0
53	1	0	1	0	1	0	1	0
54	1	0	1	0	1	0	1	0
55	1	1	1	1	1	1	1	1
56	0	0	0	0	0	0	0	0
57	1	0	1	0	1	0	1	0
58	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0
62	1	1	1	1	1	1	1	1
63	0	0	0	0	0	0	0	0
64	1	0	1	0	1	0	1	0
65	1	0	1	0	1	0	1	0
66	0	0	0	0	0	0	0	0
67	1	1	1	1	1	1	1	1
68	1	1	1	1	1	1	1	1
69	0	0	0	0	0	0	0	0
70	1	0	1	0	1	0	1	0
71	0	0	0	0	0	0	0	0
72	1	0	1	0	1	0	1	0
73	1	0	1	0	1	0	1	0
74	1	0	1	0	1	0	1	0
75	1	1	1	1	1	1	1	1
76	1	1	1	1	1	1	1	1
77	1	0	1	0	1	0	1	0
78	1	0	1	0	1	0	1	0
79	0	0	0	0	0	0	0	0
80	0	0	0	0	0	0	0	0
81	0	0	0	0	0	0	0	0
82	0	0	0	0	0	0	0	0
83	1	0	1	0	1	0	1	0

<i>Learning Rate</i>								
No.	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
84	1	1	1	1	1	1	1	1
85	0	0	0	0	0	0	0	0
86	0	0	0	0	0	0	0	0
87	0	0	0	0	0	0	0	0
88	1	1	1	1	1	1	1	1
89	0	0	0	0	0	0	0	0
90	0	0	0	0	0	0	0	0
91	0	0	0	0	0	0	0	0
92	0	0	0	0	0	0	0	0
93	0	0	0	0	0	0	0	0
94	0	0	0	0	0	0	0	0
95	0	0	0	0	0	0	0	0
96	1	1	1	1	1	1	1	1
97	0	0	0	0	0	0	0	0
98	0	0	0	0	0	0	0	0
99	1	0	1	0	1	0	1	0
100	1	0	1	0	1	0	1	0
101	1	0	1	0	1	0	1	0
102	1	0	1	0	1	0	1	0
103	1	0	1	0	1	0	1	0
104	0	0	0	0	0	0	0	0
105	0	0	0	0	0	0	0	0
106	1	1	1	1	1	1	1	1
107	1	1	1	1	1	1	1	1
108	1	1	1	1	1	1	1	1
109	0	0	0	0	0	0	0	0
110	1	0	1	0	1	0	1	0
111	0	0	0	0	0	0	0	0
112	0	0	0	0	0	0	0	0
113	0	0	0	0	0	0	0	0
114	1	0	1	0	1	0	1	0
115	1	0	1	0	1	0	1	0
116	0	0	0	0	0	0	0	0
117	0	0	0	0	0	0	0	0
118	1	0	1	0	1	0	1	0
119	1	1	1	1	1	1	1	1
120	0	0	0	0	0	0	0	0
121	1	0	1	0	1	0	1	0
122	0	0	0	0	0	0	0	0
123	1	1	1	1	1	1	1	1
124	0	0	0	0	0	0	0	0
125	1	0	1	0	1	0	1	0

<i>Learning Rate</i>								
No.	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
126	0	0	0	0	0	0	0	0
127	0	0	0	0	0	0	0	0
128	1	0	1	0	1	0	1	0

Lampiran 2 Tabel Hasil deteksi stroke rasio data 70:30

<i>Learning Rate</i>								
No.	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
1	1	1	1	1	1	1	1	1
2	1	0	1	0	1	0	1	0
3	1	1	1	1	1	1	1	1
4	1	1	1	1	1	1	1	1
5	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	0
9	0	0	0	0	0	0	0	0
10	1	1	1	1	1	1	1	1
11	0	0	0	0	0	0	0	0
12	0	0	0	0	0	0	0	0
13	0	0	0	0	0	0	0	0
14	1	0	1	0	1	0	1	0
15	0	0	0	0	0	0	0	0
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	1	0	1	0	1	0	1	0
20	1	0	1	0	1	0	1	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	1
25	1	0	1	0	1	0	1	0
26	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	1	1	1	1	1	1	1	1
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	1	0	1	0	1	0	1	0

<i>Learning Rate</i>								
No.	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
16	0	0	0	0	0	0	0	0
17	0	0	0	0	0	0	0	0
18	0	0	0	0	0	0	0	0
19	1	0	1	0	1	0	1	0
20	1	0	1	0	1	0	1	0
21	0	0	0	0	0	0	0	0
22	0	0	0	0	0	0	0	0
23	0	0	0	0	0	0	0	0
24	1	1	1	1	1	1	1	1
25	1	0	1	0	1	0	1	0
26	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0
28	0	0	0	0	0	0	0	0
29	1	1	1	1	1	1	1	1
30	0	0	0	0	0	0	0	0
31	0	0	0	0	0	0	0	0
32	0	0	0	0	0	0	0	0
33	1	0	1	0	1	0	1	0
34	0	0	0	0	0	0	0	0
35	1	1	1	1	1	1	1	1
36	1	1	1	1	1	1	1	1
37	0	0	0	0	0	0	0	0
38	0	0	0	0	0	0	0	0
39	0	0	0	0	0	0	0	0
40	0	0	0	0	0	0	0	0
41	1	0	1	0	1	0	1	0
42	0	0	0	0	0	0	0	0
43	1	0	1	0	1	0	1	0
44	1	1	1	1	1	1	1	1
45	1	1	1	1	1	1	1	1
46	1	1	1	1	1	1	1	1
47	1	0	1	0	1	0	1	0
48	1	1	1	1	1	1	1	1
49	1	0	1	0	1	0	1	0
50	0	0	0	0	0	0	0	0
51	0	0	0	0	0	0	0	0
52	1	0	1	0	1	0	1	0
53	1	0	1	0	1	0	1	0
54	1	0	1	0	1	0	1	0
55	1	1	1	1	1	1	1	1
56	0	0	0	0	0	0	0	0
57	1	0	1	0	1	0	1	0

<i>Learning Rate</i>								
No.	0.1		0.2		0.01		0.03	
	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi	Aktual	Prediksi
58	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0
60	0	0	0	0	0	0	0	0
61	0	0	0	0	0	0	0	0
62	1	1	1	1	1	1	1	1
63	0	0	0	0	0	0	0	0
64	1	0	1	0	1	0	1	0