

**SEGMENTASI TUMOR OTAK MENGGUNAKAN MODEL *LIGHTWEIGHT* U-NET
BERBASIS CITRA MRI (*MAGNETIC RESONANCE IMAGING*) OTAK**

SKRIPSI

Oleh:
MUHAMMAD HAFIDLUL QOLBI
NIM. 220605110063



**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

**SEGMENTASI TUMOR MENGGUNAKAN MODEL *LIGHTWEIGHT U-
NET* BERBASIS CITRA MRI (*MAGNETIC RESONANCE IMAGING*)
OTAK**

SKRIPSI

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

Oleh :
MUHAMMAD HAFIDLUL QOLBI
220605110063

**PROGRAM STUDI TEKNIK INFORMATIKA
FAKULTAS SAINS DAN TEKNOLOGI
UNIVERSITAS ISLAM NEGERI MAULANA MALIK IBRAHIM
MALANG
2026**

HALAMAN PERSETUJUAN


SEGMENTASI TUMOR MENGGUNAKAN MODEL *LIGHTWEIGHT U-NET* BERBASIS CITRA MRI (*MAGNETIC RESONANCE IMAGING*) OTAK

SKRIPSI


**Oleh :
MUHAMMAD HAFIDLUL QOLBI
NIM. 220605110063**

Telah Diperiksa dan Disetujui untuk Diuji:
Tanggal: 9 April 2026

Pembimbing I,


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

Pembimbing II,


Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

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



Supriyono, M.Kom
NIP. 19841010 201903 1 012

HALAMAN PENGESAHAN

SEGMENTASI TUMOR MENGGUNAKAN MODEL *LIGHTWEIGHT U-NET* BERBASIS CITRA MRI (*MAGNETIC RESONANCE IMAGING*) OTAK

SKRIPSI

Oleh :
MUHAMMAD HAFIDLUL QOLBI
NIM. 220605110063

Telah Dipertahankan di Depan Dewan Penguji Skripsi
dan Dinyatakan Diterima Sebagai Salah Satu Persyaratan
Untuk Memperoleh Gelar Sarjana Komputer (S.Kom)
Tanggal: 16 April 2026

Susunan Dewan Penguji

Ketua Penguji : Prof. Dr. Suhartono, S. Si, M.Kom
NIP. 19680519 200312 1 001

Anggota Penguji I : Okta Qomaruddin Aziz, M.Kom
NIP. 19911019 201903 1 013

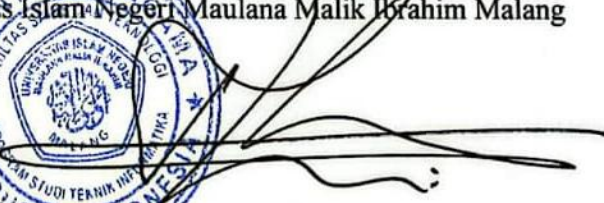
Anggota Penguji II : Dr. Irwan Budi Santoso, M.Kom
NIP. 19770103 201101 1 004

Anggota Penguji III : Shoffin Nahwa Utama, M.T
NIP. 19860703 202012 1 003

()
()
()
()

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




Supriyono, M.Kom
NIP. 19841010 201903 1 012

PERNYATAAN KEASLIAN TULISAN

Saya yang bertanda tangan di bawah ini:

Nama : Muhammad Hafidul Qolbi
NIM : 220605110063
Fakultas / Program Studi : Sains dan Teknologi / Teknik Informatika
Judul Skripsi : Segmentasi Tumor Otak Menggunakan Model
Lightweight U-Net Berbasis Citrak MRI (Magnetic Resonance Imaging) Otak

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, 16 April 2026
Yang membuat pernyataan,



Muhammad Hafidul Qolbi
NIM. 220605110063

MOTTO

“Happiness can be found even in the darkest of times, if one only remembers to turn on the light”

-Albus Dumbledore-

HALAMAN PERSEMBAHAN

Alhamdulillah rabbil 'alamin puji syukur kepada Allah SWT karena berkat rahmat dan petunjuk-Nya penulis dapat menyelesaikan skripsi ini. Shalawat serta salam selalu tercurahkan kepada Rasulullah SAW yang telah membawa kita dari zaman jahiliyah menuju addinul Islam.

Penulis mempersembahkan karya ini kepada diri orang tua, diri sendiri, kakak, para dosen, teman-teman, serta orang-orang yang telah membantu, mendoakan, serta menyemangati penulis dalam menuntaskan skripsi ini.

KATA PENGANTAR

Assalamu'alaikum warahmatullahi wabarakatuh

Segala puji dan rasa syukur penulis panjatkan ke hadirat Allah *Subhanahu wa Ta'ala* atas limpahan rahmat dan hidayah-Nya, sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Shalawat dan salam semoga senantiasa tercurah kepada Nabi Muhammad SAW, semoga kita semua kelak memperoleh syafaat beliau di hari kiamat. Aamiin.

Penulis ucapkan terima kasih yang sebesar-besarnya kepada seluruh pihak yang memberikan dukungan dan motivasi kepada penulis sehingga dapat menyelesaikan skripsi ini. Ucapan terima kasih ini penulis disampaikan kepada:

1. Prof. Dr. Hj. Ilfi Nur Diana, M.Si, selaku Rektor Universitas Islam Negeri Maulana Malik Ibrahim Malang.
2. Dr. H. Agus Mulyono, M.Si, selaku Dekan Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang.
3. Supriyono, M.Kom, selaku Ketua Program Studi Teknik Informatika Universitas Islam Negeri Maulana Malik Ibrahim Malang.
4. Dr. Irwan Budi Santoso, M.Kom, selaku dosen pembimbing 1 serta mentor yang telah sangat sabar memberikan bimbingan, saran, dan dukungan yang sangat berarti bagi penulis selama proses penelitian.
5. Shoffin Nahwa Utama, M.T, selaku dosen pembimbing 2, atas arahan dan pencerahan yang sangat membantu dalam menyempurnakan karya ini.

6. Prof. Dr. Suhartono, S.Si, M.Kom, selaku dosen Penguji I dan Okta Qomaruddin Aziz, M.Kom, selaku dosen Penguji II yang telah memberikan kritik dan saran yang membangun dalam penyusunan skripsi ini.
7. Seluruh staf dan dosen Program Studi Teknik Informatika yang telah memberikan ilmu, dukungan, dan fasilitas kepada penulis selama masa studi.
8. Ayahanda Supriadi dan Ibunda Anik Indrawati yang memiliki peran besar dalam perjalanan penulis selama menempuh pendidikan perkuliahan. Terima kasih atas segala perjuangan, dukungan, perhatian, semangat, pengorbanan, serta doa yang senantiasa diberikan kepada penulis dalam setiap proses penyelesaian skripsi ini. Semoga Allah SWT senantiasa melimpahkan kesehatan, keberkahan, kebahagiaan, serta perlindungan kepada beliau berdua, dan membalas seluruh kebaikan dengan pahala yang berlipat ganda.
9. M. Singgi Aditya Ramadhan, M. Yasril Adim Al-Amin, dan Lefa Maulana Alghifary yang senantiasa menjadi rekan berdiskusi dan bertukar pemikiran selama masa perkuliahan hingga proses penyelesaian skripsi ini. Terima kasih atas dukungan, semangat, motivasi, serta berbagai pandangan dan pengalaman yang telah dibagikan, baik dalam bidang akademik maupun kehidupan sehari-hari, sehingga memberikan banyak pelajaran dan dorongan bagi penulis dalam menyelesaikan studi ini.
10. Teman-teman yang selalu hadir dan menemani penulis selama masa perkuliahan, yaitu Salman, Hilmi, Ghiffari, Fairuz, Agam, Tiara, Ayu, Zaka,

Aji, Epen, Dimas, Mukhtar, Jayan, Fawwaz, dan Adhi, serta teman-teman lainnya yang tidak dapat disebutkan satu per satu. Terima kasih atas kebersamaan, dukungan, semangat, serta berbagai proses suka dan duka yang telah dilalui bersama selama masa perkuliahan. Semoga segala kebaikan dan pengalaman yang telah dilalui menjadi langkah menuju kesuksesan bagi kita semua.

11. Keluarga Besar UKM PSM Gema Gita Bahana. Terima kasih telah memberikan pengalaman yang sangat berharga berupa prestasi Nasional dan Internasional

12. Seluruh pihak yang telah terlibat, baik secara langsung maupun tidak langsung dari awal perkuliahan hingga akhir penulisan skripsi ini

Penulis berharap semoga skripsi ini dapat memberikan manfaat di masa yang akan datang. Penulis juga menyadari bahwa karya ini masih memiliki kekurangan. Oleh karena itu, penulis sangat terbuka terhadap segala bentuk masukan dan saran yang membangun untuk pengembangan lebih lanjut.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Malang, 16 April 2026

Penulis

DAFTAR ISI

HALAMAN PERSETUJUAN	iii
HALAMAN PENGESAHAN.....	iv
PERNYATAAN KEASLIAN TULISAN	v
MOTTO	vi
HALAMAN PERSEMBAHAN	vii
KATA PENGANTAR.....	viii
DAFTAR ISI.....	xi
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xiv
ABSTRAK	xv
ABSTRACT.....	xvi
مستخلص البحث	xvii
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Identifikasi Masalah.....	4
1.3 Batasan Penelitian.....	4
1.4 Tujuan Penelitian	4
1.5 Manfaat Penelitian	5
BAB II STUDI PUSTAKA	6
2.1 Segmentasi Tumor Otak.....	6
2.2 Tumor Otak.....	11
2.3 Model U-Net.....	11
BAB III DESAIN DAN IMPLEMENTASI.....	15
3.1 Prosedur Penelitian	15
3.2 Pengumpulan Data.....	15
3.3 Desain Sistem.....	17
3.3.1 <i>Input Image</i>	18
3.3.2 <i>Preprocessing Citra</i>	19
3.3.3 Arsitektur Usulan <i>Lightweight U-Net</i>	20
3.3.4 Fungsi Aktivasi	28
3.3.5 Model Pembelajaran	32
3.4 Skenario Uji coba.....	40
3.4.1 Pengelompokan Dataset.....	40
3.4.2 Pengujian Parameter Sistem.....	41
3.5 Evaluasi Kinerja Sistem.....	43
3.5.1 <i>Dice Coefficient</i>	44
3.5.2 Akurasi	44
3.5.3 Sensitifitas (<i>Recall</i>).....	45
3.5.4 <i>Precision</i>	46
BAB IV HASIL DAN PEMBAHASAN.....	47
4.1 Hasil <i>Training</i> Setiap Skenario	47
4.1.1 Hasil <i>Training</i> Skenario 1	48

4.1.2 Hasil <i>Training</i> Skenario 2	50
4.1.3 Hasil <i>Training</i> Skenario 3	51
4.2 Hasil <i>Testing</i> Setiap Skenario.....	53
4.2.1 Hasil <i>Testing</i> Skenario 1.....	54
4.2.2 Hasil <i>Testing</i> Skenario 2.....	57
4.2.3 Hasil <i>Testing</i> Skenario 3.....	60
4.2.4 Hasil Perbandingan <i>Testing</i> Setiap Skenario.....	63
4.3 Integrasi Islam.....	79
BAB V KESIMPULAN DAN SARAN	83
5.1 Kesimpulan	83
5.2 Saran	84
DAFTAR PUSTAKA	

DAFTAR GAMBAR

Gambar 2. 1 Arsitektur U-Net (Ronneberger dkk., 2015).....	12
Gambar 3. 1 Prosedur Penelitian.....	15
Gambar 3. 2 Sample Tumor Otak	16
Gambar 3. 3 Skema Training dan Testing.....	17
Gambar 3. 4 Resize Image	19
Gambar 3. 5 Arsitektur Usulan LLightweight U-Net.....	21
Gambar 3. 6 Algoritma 1 Convolution Operation 2D (ManualConv2D)	25
Gambar 3. 7 Max-Polling (Haq, 2022)	27
Gambar 3. 8 Algoritma 2 Max-Pooling Operation (ManualMaxPooling2D)	27
Gambar 3. 9 Diagram ReLU (Sharma dkk., 2020)	29
Gambar 3. 10 Diagram Sigmoid (Sharma dkk., 2020)	30
Gambar 3. 11 Algoritma 3 Membangun U-Net Construction (build_manual_unet).....	31
Gambar 3. 12 Algoritma 4 Forward Pass U-Net.....	34
Gambar 3. 13 Algoritma 5 Backward Pass U-Net	38
Gambar 4. 1 Hasil Training Skenario 1.....	49
Gambar 4. 2 Hasil Training Skenario 2.....	51
Gambar 4. 3 Hasil Training Skenario 3.....	53
Gambar 4. 4 Hasil Segmentasi Skenario 1.....	55
Gambar 4. 5 Hasil Segmentasi Skenario.....	58
Gambar 4. 6 Hasil Segmentasi Skenario 3.....	61
Gambar 4. 7 Hasil Perbandingan Setiap Skenario	66
Gambar 4. 8 Hasil Perbandingan Nilai Dice Coefficient.....	68
Gambar 4. 9 Sample Outlier Skenario 1	69
Gambar 4. 10 Sample Outlier Skenario 2	70
Gambar 4. 11 Sample Outlier Skenario 3	71

DAFTAR TABEL

Tabel 2. 1 Segmentasi Tumor Otak.....	10
Tabel 3. 1 Pengelompokan Tumor	16
Tabel 3. 2 Proporsi Split Data	40
Tabel 3. 3 Label Area Tumor	41
Tabel 3. 4 Parameter yang Diuji.....	42
Tabel 3. 5 Skenario Pengujian.....	43
Tabel 4. 1 Confusion Matrix Skenario 1 Gambar 47.png	56
Tabel 4. 2 Confusion Matrix Skenario 1 Gambar 232.png	56
Tabel 4. 3 Confusion Matrix Skenario 1 Gambar 2812.png	57
Tabel 4. 4 Confusion Matrix Skenario 2 Gambar 47.png	59
Tabel 4. 5 Confusion Matrix Skenario 2 Gambar 232.png	59
Tabel 4. 6 Confusion Matrix Skenario 2 Gambar 2812.png	60
Tabel 4. 7 Confusion Matrix Skenario 3 Gambar 47.png	62
Tabel 4. 8 Confusion Matrix Skenario 3 Gambar 232.png	62
Tabel 4. 9 Confusion Matrix Skenario 3 Gambar 2812.png	63
Tabel 4. 10 Hasil Performa Setiap Skenario	65
Tabel 4. 11 Hasil p-value Uji T-Test Dice Coefficient Antar Skenario Learning Rate	74
Tabel 4. 12 Keputusan Uji T-Test Dice Coefficient Antar Learning Rate.....	75
Tabel 4. 13 Hasil p-value Uji T-Test Pixel Accuracy Antar Skenario Learning Rate	76
Tabel 4. 14 Keputusan Uji T-Test Pixel Accuracy Antar Learning Rate.....	76
Tabel 4. 15 Hasil p-value Uji T-Test Dice Coefficient Antar Epoch Learning Rate	77
Tabel 4. 16 Keputusan Uji T-Test Dice Coefficient Antar Epoch Learning Rate.....	78
Tabel 4. 17 Hasil p-value Uji T-Test Pixel Accuracy Antar Epoch Learning Rate	79
Tabel 4. 18 Keputusan Uji T-Test Pixel Accuracy Antar Epoch Learning Rate.....	79

ABSTRAK

Qolbi, Muhammad Hafidul. 2026. **Segmentasi Tumor Otak Menggunakan Model *Lightweight* U-Net Berbasis MRI (*Magnetic Resonance Imaging*) Otak**. Skripsi. Program Studi Teknik Informatika Fakultas Sains dan Teknologi Universitas Islam Negeri Maulana Malik Ibrahim Malang. Pembimbing: (I) Dr. Irwan Budi Santoso, M.Kom. (II) Shoffin Nahwa Utama, M.T.

Kata Kunci: *Lightweight* U-Net, Segmentasi Tumor Otak, MRI, *Dice Coefficient*, *Pixel Accuracy*.

Segmentasi tumor otak pada citra MRI merupakan tahapan penting dalam membantu diagnosis dan perencanaan pengobatan, namun masih menghadapi tantangan dalam menghasilkan segmentasi yang akurat dan efisien. Penelitian ini bertujuan untuk menganalisis performa model *Lightweight* U-Net dalam melakukan segmentasi tumor otak menggunakan citra MRI berdasarkan metrik *Dice Coefficient* dan *Pixel Accuracy*. Metode yang digunakan adalah *Lightweight* U-Net dengan beberapa skenario pengujian yang melibatkan variasi *Learning Rate* dan jumlah *Epoch*. Dataset yang digunakan berupa citra MRI tumor otak yang telah melalui tahap *preprocessing*. Evaluasi dilakukan menggunakan *Dice Coefficient* untuk mengukur tingkat kesesuaian area segmentasi dan *Pixel Accuracy* untuk mengukur ketepatan klasifikasi piksel, serta didukung dengan uji statistik *paired sample T-Test* untuk mengetahui signifikansi perbedaan performa antar skenario. Hasil penelitian menunjukkan bahwa model mampu menghasilkan nilai *Pixel Accuracy* di atas 0,99 pada seluruh skenario, serta nilai *Dice Coefficient* terbaik sebesar 0,7483. Hasil uji *T-Test* menunjukkan bahwa *Learning Rate* 0,001 dan 0,0001 tidak memiliki perbedaan performa yang signifikan, sedangkan *Learning Rate* 0,00001 menghasilkan performa yang secara signifikan lebih rendah. Selain itu, peningkatan jumlah *Epoch* hingga 200 memberikan pengaruh terhadap peningkatan performa model. Dengan demikian, model *Lightweight* U-Net yang diusulkan mampu memberikan hasil segmentasi yang baik dan konsisten, dengan kombinasi *Learning Rate* 0,001 dan 200 *Epoch* sebagai konfigurasi yang paling optimal.

ABSTRACT

Qolbi, Muhammad Hafidlul. 2026. **Brain Tumor Segmentation Using a Lightweight U-Net Model Based on Brain MRI (Magnetic Resonance Imaging)**. Thesis. Informatics Engineering Study Program, Faculty of Science and Technology, Universitas Islam Negeri Maulana Malik Ibrahim Malang. Advisors: (I) Dr. Irwan Budi Santoso, M.Kom. (II) Shoffin Nahwa Utama, M.T.

Keywords: *Lightweight U-Net, Brain Tumor Segmentation, MRI, Dice Coefficient, Pixel Accuracy.*

Brain tumor segmentation in MRI images is an important step in supporting diagnosis and treatment planning; however, it still faces challenges in producing accurate and efficient segmentation. This research aims to analyze the performance of the Lightweight U-Net model in segmenting brain tumors using MRI images based on the Dice Coefficient and Pixel Accuracy metrics. It employs Lightweight U-Net with several testing scenarios involving variations in Learning Rate and number of Epochs. The dataset consists of brain tumor MRI images that have undergone preprocessing. Evaluation is conducted using the Dice Coefficient to measure the overlap between segmented areas and Pixel Accuracy to assess pixel classification accuracy, supported by a paired sample T-Test to determine the significance of performance differences between scenarios. The result shows that the model is able to achieve Pixel Accuracy values above 0.99 across all scenarios, with the best Dice Coefficient reaching 0.7483. The T-Test results indicate that learning rates of 0.001 and 0.0001 do not show significant performance differences, while a learning rate of 0.00001 results in significantly lower performance. Additionally, increasing the number of Epochs up to 200 contributes to improved model performance. Thus, the proposed Lightweight U-Net model is capable of producing good and consistent segmentation results, with a learning rate of 0.001 and 200 Epochs identified as the most optimal configuration.

مستخلص البحث

القلب، محمد حافظ. 2026. تقسيم الأورام الدماغية باستخدام نموذج يو-نت خفيف الوزن القائم على التصوير بالرنين المغناطيسي للدماغ (MRI). البحث الجامعي. قسم الهندسة المعلوماتية، كلية العلوم والتكنولوجيا بجامعة مولانا مالك إبراهيم الإسلامية الحكومية مالانج. المشرف الأول: د. إيروان بودي سانتوسو، الماجستير؛ المشرف الثاني: صفين نحو أوتاما، الماجستير.

الكلمات الرئيسية: يو-نت خفيف وزن، تقسيم أورام دماغية، تصوير رنين مغناطيسي، معامل دايس، دقة بكسل.

تقسيم الأورام الدماغية في صور التصوير بالرنين المغناطيسي يُعد خطوة مهمة في المساعدة على التشخيص وتخطيط العلاج، ومع ذلك لا يزال يواجه تحديات في تحقيق تقسيم دقيق وفعال. هدف هذا البحث إلى تحليل أداء نموذج يو-نت خفيف الوزن في إجراء تقسيم الأورام الدماغية باستخدام التصوير بالرنين المغناطيسي بناءً على مؤشر معامل دايس (*Dice Coefficient*) ودقة البكسل (*Pixel Accuracy*). تعتمد الطريقة المستخدمة على نموذج يو-نت خفيف الوزن مع بعض سيناريوهات الاختبار التي تشمل تغييرات في معدل التعلم (*Learning Rate*) وعدد الحقبات (*Epoch*). تستخدم مجموعة البيانات صور التصوير بالرنين المغناطيسي للأورام الدماغية التي خضعت لمرحلة المعالجة المسبقة. تم التقييم باستخدام معامل دايس لقياس درجة توافق منطقة التقسيم ودقة البكسل لقياس دقة تصنيف البكسل، بالإضافة إلى دعمها بالاختبار الإحصائي المزدوج لعينة الاختبار ت لمعرفة دلالة الفروقات في الأداء بين السيناريوهات. أظهرت نتائج البحث أن النموذج قادر على تحقيق قيمة دقة البكسل أكثر من 0.99 في جميع السيناريوهات، وكذلك أفضل قيمة لمعامل دايس بلغت 0.7483. أظهرت نتائج اختبار ت أن معدل التعلم 0.001 و 0.0001 ليس لها فرق أداء كبير، بينما معدل التعلم 0.00001 أدى إلى أداء أقل بشكل ملحوظ. بالإضافة إلى ذلك، فإن زيادة عدد الحقبات إلى 200 كان لها تأثير على تحسين أداء النموذج. وبالتالي، فإن نموذج يو-نت خفيف الوزن المقترح قادر على تقديم نتائج تقسيم جيدة ومتسقة، مع الجمع بين معدل التعلم 0.001 وحقبة 200 كأفضل تكوين.

BAB I

PENDAHULUAN

1.1 Latar Belakang

Tumor otak merupakan salah satu jenis tumor pada sistem saraf pusat yang menimbulkan beban kesehatan serius secara global. Berdasarkan estimasi GLOBOCAN 2020, terdapat sekitar 308.000 kasus baru dan 251.000 kematian akibat kanker otak, dengan angka kejadian sebesar 3,9 per 100.000 penduduk serta angka kematian sebesar 3,3 per 100.000 penduduk (Sung dkk., 2021). Proyeksi global menunjukkan bahwa jumlah kasus kanker otak akan terus meningkat seiring pertumbuhan dan penuaan populasi, sehingga beban penyakit ini diperkirakan semakin berat pada dekade mendatang (Zhang dkk., 2025).

Magnetic Resonance Imaging (MRI) telah menjadi standar emas dalam diagnosis tumor otak, namun interpretasi dan segmentasi area tumor pada MRI masih sangat bergantung pada keahlian radiolog. Kondisi ini menyebabkan proses diagnosis berlangsung lambat, rentan bias, dan sulit diakses secara merata, khususnya di negara berkembang (Diana-Albelda dkk., 2025). Hal ini menegaskan kebutuhan mendesak terhadap metode deteksi dan segmentasi tumor otak yang lebih cepat, akurat, dan skalabel, terutama dalam menentukan letak serta ukuran tumor guna mendukung penanganan yang tepat dan menekan risiko kematian.

Allah SWT berfirman:

يَا أَيُّهَا الَّذِينَ آمَنُوا اسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ ۚ إِنَّ اللَّهَ مَعَ الصَّابِرِينَ

“Hai orang-orang yang beriman, jadikanlah sabar dan salat sebagai penolongmu, sesungguhnya Allah bersama orang-orang yang sabar.” (QS. Al-Baqarah: 153).

Adapun tafsir ayat menurut laman NU Online yang dituliskan oleh Ustadz Alwi Jamalulel Ubab menjelaskan bahwa Allah memerintahkan orang beriman untuk memohon pertolongan melalui sabar dan salat. Kesabaran yang paling utama adalah kesabaran dalam meninggalkan maksiat, karena menahan diri dari dosa lebih berat daripada menanggung ujian. Setelah itu, kesabaran melaksanakan ketaatan secara konsisten menempati urutan kedua, dan kesabaran menghadapi musibah berada pada tingkat ketiga. Salat di sini dipahami sebagai sarana untuk memperkuat kesabaran, baik wajib maupun sunnah, sedangkan janji Allah “Sesungguhnya Allah bersama orang-orang yang sabar” menunjukkan pertolongan dan penjagaan-Nya bagi mereka yang sabar (Ubab, 2022). Di sisi lain hadist Rasulullah ﷺ juga bersabda:

خَيْرُ النَّاسِ أَنْفَعُهُمْ لِلنَّاسِ

"Sebaik-baik manusia adalah yang paling bermanfaat bagi manusia lain." (HR. Ahmad, ath-Thabrani, ad-Daruqutni).

Hadis ini menegaskan bahwa penelitian yang berkontribusi pada peningkatan diagnosis dan penanganan tumor otak merupakan bentuk nyata menghadirkan manfaat bagi sesama. Dari kedua dalil tersebut mengandung *Maqashit Syariah* diantaranya *Hifzh ad-Din* yang berarti mengajarkan manusia untuk bersandar kepada Allah SWT melalui ibadah (shalat) serta selalu menguatkan keimanan dan sabar dalam menghadapi ujian berupa penyakit, *Hifzh an-Nafs* dengan maksud penelitian di bidang kesehatan secara langsung menjaga dan menyelamatkan jiwa manusia, dan terakhir *Hifzh al-'Aql* yang bermakna penggunaan akal secara optimal untuk kemaslahatan demi menjaga kehidupan.

Salah satu tantangan utama dalam penanganan tumor otak adalah segmentasi wilayah tumor dari citra MRI secara cepat, akurat, dan konsisten. Segmentasi manual memerlukan waktu lama, bergantung pada keahlian radiolog, serta rawan subjektivitas. Metode otomatis berbasis *deep learning*, khususnya U-Net, telah terbukti efektif dalam segmentasi citra medis. Namun, sebagian besar penelitian masih berfokus pada dataset publik seperti BRATS dengan model berparameter besar yang membutuhkan data pelatihan lebih banyak dan berisiko *overfitting* pada dataset berukuran kecil (Matuszewski & Sintorn, 2019).

Penelitian ini menggunakan Lightweight U-Net karena model yang lebih sederhana mampu menjaga stabilitas pelatihan, mengurangi risiko *overfitting*, dan tetap memberikan akurasi yang baik pada data terbatas. Walsh dkk., mengusulkan *Lightweight* U-Net dengan menyederhanakan struktur jaringan tanpa menghilangkan detail penting, sehingga segmentasi dapat dilakukan lebih cepat dengan kebutuhan komputasi dan data yang lebih rendah. Efektivitas pendekatan ini juga didukung sejumlah penelitian yang menunjukkan nilai *Pixel Accuracy* dan *Dice Coefficient* tinggi pada segmentasi tumor otak berbasis U-Net (Arora et al., 2021; Sangui et al., 2022; Suta et al., 2020). Dengan karakteristik tersebut, *Lightweight* U-Net berpotensi menjadi solusi yang tepat untuk mengatasi keterbatasan sumber daya dan ukuran dataset, terutama di fasilitas kesehatan dengan perangkat keras terbatas.

Berdasarkan potensi penelitian sebelumnya, penelitian ini mengusulkan metode segmentasi tumor otak menggunakan arsitektur *Lightweight* U-Net usulan berbasis citra MRI data publik dari Kaggle yang bersumber dari figshare untuk

mengetahui performa evaluasi *Dice Coefficient* dan *Pixel Accuracy* dalam mendeteksi serta memetakan area tumor. Data MRI akan melalui tahap pra-pemrosesan untuk menstandarkan kualitas dan format citra agar hasil segmentasi lebih konsisten. *Lightweight* U-Net usulan berperan sebagai model utama untuk mengekstraksi fitur spasial serta membedakan jaringan sehat dan jaringan tumor secara detail.

1.2 Identifikasi Masalah

Berdasarkan latar belakang yang telah dipaparkan, permasalahan utama yang diangkat oleh peneliti adalah bagaimana performa evaluasi *Dice Coefficient* dan *Pixel Accuracy* pada model *Lightweight* U-Net usulan dalam segmentasi tumor otak menggunakan citra MRI.

1.3 Batasan Penelitian

Batasan yang digunakan dalam penelitian ini adalah penggunaan datasets publik *Brain Tumor Segmentasi* dengan format gambar (.png) dari Kaggle oleh Nikhil Tomar pada 2022 (Cheng, 2015).

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah untuk mengetahui performa evaluasi *Dice Coefficient* dan *Pixel Accuracy* pada model *Lightweight* U-Net usulan dalam segmentasi tumor otak menggunakan citra MRI.

1.5 Manfaat Penelitian

Peneliti Penelitian ini diharapkan dapat memberikan manfaat sebagai berikut:

1. Menyediakan alat alternatif segmentasi tumor otak berbasis MRI yang mampu mempercepat alur pemeriksaan, membantu penentuan lokasi serta bentuk tumor, dan meningkatkan kualitas layanan diagnostik kepada pasien bagi lembaga kesehatan seperti rumah sakit atau klinik.
2. Memberikan dukungan dalam menganalisis citra MRI dengan hasil segmentasi yang konsisten, sehingga mengurangi potensi kesalahan interpretasi, memperingan beban kerja manual, serta membantu dokter atau tenaga ahli dalam menentukan rencana penanganan yang lebih tepat bagi pasien.

BAB II

STUDI PUSTAKA

2.1 Segmentasi Tumor Otak

Sejumlah penelitian terdahulu telah membahas model U-Net dalam segmentasi tumor otak, sebagaimana dirangkum dalam tabel berikut. Misbullah dkk., membandingkan dua arsitektur *Convolutional Neural Network* (CNN), yaitu U-Net dan Res-UNet, untuk segmentasi citra MRI tumor otak menggunakan dataset BraTS 2020. Kedua model dilatih dengan konfigurasi serupa menggunakan optimizer Adam dan *categorical crossentropy loss*, lalu dievaluasi menggunakan metrik akurasi dan *Dice Coefficient*. Hasil menunjukkan bahwa U-Net mencapai performa optimal pada 35 epoch, *batch size* 64, dan *learning rate* 0,001, dengan *loss* terendah 0,014 dan akurasi tertinggi 99,5%, mengungguli Res-UNet yang menghasilkan akurasi 99,3% dengan *loss* lebih besar.

Penelitian Walsh dkk., mengembangkan *Lightweight* U-Net, yaitu arsitektur U-Net yang dimodifikasi menjadi lebih ringan, untuk segmentasi tumor otak pada citra MRI secara efisien dan *real-time* tanpa membutuhkan dataset besar maupun augmentasi data tambahan. Model diuji pada dataset BITE menggunakan tiga bidang perspektif (*coronal, sagittal, transversal*) yang disederhanakan menjadi citra 2D. Hasil menunjukkan bahwa *Lightweight* U-Net mencapai rata-rata *Intersection over Union* (IoU) hingga 89% dan secara konsisten mengungguli metode lain seperti *thresholding, K-Means, Fuzzy C-Means*, dan *LinkNet*,

membuktikan bahwa model ini merupakan solusi praktis sekaligus akurat dalam segmentasi tumor otak berbasis MRI.

Penelitian menurut Lin & Lin menggunakan arsitektur U-Net yang dipadukan dengan EfficientNetV2 sebagai encoder untuk meningkatkan akurasi segmentasi tumor otak pada citra MRI, khususnya pada kasus *pleomorphic glioblastoma*. Hasil eksperimen menunjukkan performa tinggi dengan nilai *loss* = 0,0866, akurasi = 0,9977, dan *Dice Similarity Coefficient* (DSC) = 0,9133, membuktikan bahwa kombinasi ini lebih efektif dibandingkan pendekatan konvensional dalam menghasilkan segmentasi yang presisi.

Menurut penelitian Suta dkk., menerapkan arsitektur U-Net untuk segmentasi otomatis tumor otak pada citra MRI menggunakan dataset BRATS 2017 dengan konfigurasi *learning rate* 0,0001, *batch size* 5, 80 *epoch*, fungsi aktivasi ReLU, dan *optimizer Adam*. Hasil menunjukkan nilai *Dice Coefficient* sebesar 90,22% pada *Full Tumor*, 78,09% pada *Core Tumor*, dan 80,20% pada *Enhancing Tumor*, membuktikan bahwa U-Net efektif digunakan untuk segmentasi otomatis tumor otak berbasis citra MRI.

Penelitian yang dilakukan oleh Arora dkk. (2021) menggunakan pendekatan *deep learning* dengan arsitektur U-Net untuk segmentasi tumor otak pada citra MRI. Fokus penelitian ini adalah mengoptimalkan performa segmentasi otomatis pada dataset BraTS 2018 dengan memanfaatkan U-Net dalam membedakan area tumor dari jaringan otak sehat. Model ini dilatih dan diuji menggunakan *dice coefficient* sebagai metrik utama untuk menilai kualitas segmentasi. Hasil penelitian menunjukkan bahwa U-Net mampu mencapai nilai *dice coefficient* yang tinggi pada

berbagai tipe tumor, membuktikan bahwa arsitektur ini efektif dan andal dalam tugas segmentasi citra medis (Arora dkk., 2021).

Penelitian menurut Sanguia dkk., mengembangkan arsitektur U-Net yang dimodifikasi untuk segmentasi tumor otak dari citra 3D MRI menggunakan dataset BRATS 2020, dievaluasi dengan metrik akurasi, *Dice Coefficient*, *Mean IoU*, sensitivitas, dan spesifisitas. Hasil menunjukkan akurasi validasi 99,39%, *Dice Coefficient* 65,67, dan *Mean IoU* 83,26, secara konsisten mengungguli model VGG16 dan ResNet50, membuktikan efektivitas pendekatan ini dalam mendukung diagnosis medis berbasis citra MRI.

Pourmahboubi dkk. (2025) mengembangkan U-Net berbasis VGG19 dengan transfer learning dan fungsi *loss Focal Tversky* untuk segmentasi tumor otak pada citra MRI, khususnya dalam mendeteksi batas tumor yang kompleks. Hasil eksperimen pada dataset TCGA-LGG menunjukkan performa unggul dibandingkan U-Net standar maupun variasi backbone lain, dengan AUC 0,9957, *F1-score* 0,9679, *Dice Coefficient* 0,9679, *Precision* 0,9541, *Recall* 0,9821, dan *IoU* 0,9378.

Mestre dkk., mengoptimalkan inferensi model *deep learning* untuk perangkat IoT dengan keterbatasan sumber daya melalui *layer fusion* dan kuantisasi 8-bit. Pendekatan ini menyatukan lapisan Conv2D, ReLU, dan *Batch Normalization* menjadi satu operator baru bernama FConv2D. Model dievaluasi menggunakan prosesor *Arduino Nano 33 BLE Sense* (Cortex-M4) dan *Arduino Portenta H7 Lite* (Cortex-M4/M7) pada dataset CIFAR-10. Hasil menunjukkan pengurangan waktu inferensi hingga $1,53\times$ secara keseluruhan dan percepatan

2,95× pada lapisan tertentu, dengan ukuran model lebih kecil dan akurasi tetap stabil ($\approx 83,7\%$).

Menurut Abbas dkk., mengusulkan model *hybrid* U-Net yang dikombinasikan dengan EfficientNetV2-S untuk segmentasi dan klasifikasi tumor otak menggunakan dataset Figshare, yang terdiri dari 3.064 citra MRI T1-*weighted* dari 233 pasien mencakup tiga jenis tumor: *glioma*, *meningioma*, dan *pituitary*. Model menggunakan pendekatan *double hybrid encoder* untuk ekstraksi fitur pada tahap segmentasi, sementara klasifikasi dilakukan menggunakan EfficientNetV2-S yang telah di-*fine-tuning*, dilatih dengan *optimizer Adam* dan *learning rate* 0,0001. Hasil menunjukkan *Dice Score* 95,27% dan IoU 86,89%, mengungguli U-Net dan U-Net++ konvensional, membuktikan bahwa kombinasi arsitektur *hybrid* dan dataset beranotasi lengkap mampu meningkatkan akurasi segmentasi dan klasifikasi tumor otak secara signifikan.

Penelitian menurut Preetha dkk., mengusulkan model *Multiscale Attention* U-Net dengan EfficientNetB4 sebagai *encoder* untuk segmentasi citra MRI tumor otak menggunakan dataset Figshare. Model memanfaatkan mekanisme *attention* dan ekstraksi fitur multiskala untuk meningkatkan kemampuan menangkap fitur spasial dan kontekstual pada citra tumor, dievaluasi menggunakan metrik akurasi dan *Dice Coefficient*. Hasil menunjukkan akurasi 99,79% dan *Dice Score* 93,39%, membuktikan bahwa kombinasi EfficientNetB4 dengan pendekatan *Multiscale Attention* U-Net mampu meningkatkan akurasi segmentasi tumor otak secara signifikan dibandingkan metode konvensional.

Tabel 2. 1 Segmentasi Tumor Otak

No.	Peneliti	Metode	Dataset	Hasil	Perbedaan
1.	(Misbullah dkk., 2024)	<i>Deep Learning</i> ; U-Net dan Res-UNet	BraTS Dataset	U-Net: akurasi tertinggi 99,5%, Res-UNet: akurasi 99,3%	Perbandingan U-Net dan Res-UNet
2.	(Walsh dkk., 2022)	<i>Lightweight U-Net</i>	BITE Dataset	Rata-rata IoU: 89%	U-Net versi ringan (<i>lightweight</i>)
3.	(Lin & Lin, 2024)	U-Net (<i>encoder EfficientNetV2 + majority voting</i>)	BraTS Dataset	<i>Akurasi 0,9977; Dice Similarity Coefficient (DSC) 0,9133; Loss 0,0866</i>	Mengatasi Kelemahan U-Net
4.	(Suta dkk., 2020)	<i>Deep Learning</i> : U-Net	BraTS Dataset	Full Tumor: 90,22%, Core Tumor: 78,09%, Enhancing Tumor: 80,20%	Menerapkan Segmentasi Otomatis
5.	(Arora dkk., 2021)	<i>Deep Learning</i> : U-Net	BraTS Dataset	Mencapai nilai <i>dice coefficient</i> tinggi	Membuktikan konsistensi U-Net dalam tugas segmentasi
6.	(Sangui dkk., 2022)	U-Net	BraTS Dataset	Akurasi validasi 99,39%, <i>Dice Coefficient</i> 65,67	Modifikasi dalam U-Net 3D
7.	(Pourmahboubi dkk., 2025)	U-Net (backbone VGG19, menggunakan <i>Focal Tversky loss</i>)	TCGA-LGG	AUC: 0.9957 dan F1-score: 0.9679	Memanfaatkan VGG19 dan menggunakan <i>Focal Tversky loss</i>
8.	(Dolz1, 2025)	<i>Deep Learning</i> melalui Conv2D-ReLU-BN	<i>Benchmark Deep Learning Models</i>	Meningkatkan pengurangan memori 50%, kecepatan inferensi naik 40%, dan akurasi stabil (<1%)	Fokus Optimasi Efisiensi Model
9.	(Abbas dkk., 2026)	<i>Hybrid U-Net (double hybrid encoder) + EfficientNetV2-S</i>	Figshare Version 5	<i>Accuracy</i> (klasifikasi): 99%, <i>Dice Coefficient</i> : 95,27%, dan IoU: 86,89%	Penggunaan Variasi Model
10.	(Preetha dkk., 2025)	EfficientNetB4 + Multiscale Attention U-Net	Figshare	<i>Accuracy</i> : 99,79% dan <i>Dice</i> : 93,39%	Penggunaan Variasi Model

2.2 Tumor Otak

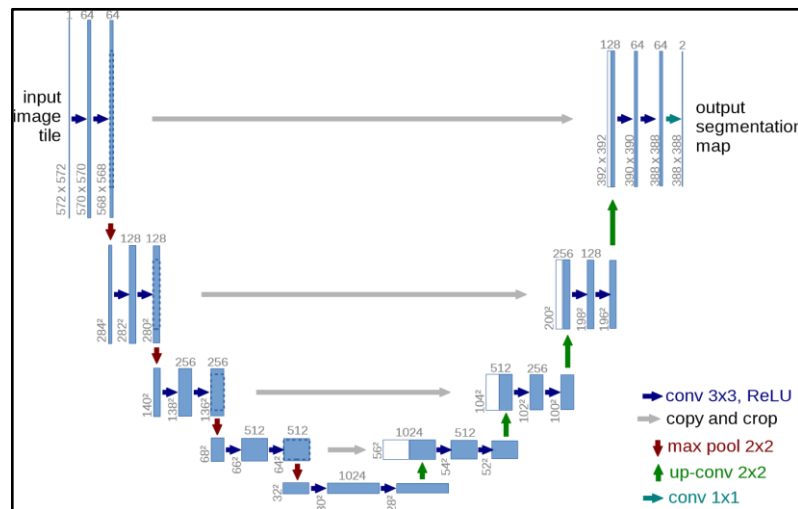
Tumor otak adalah massa atau pertumbuhan sel atau jaringan yang tidak normal pada otak. Tumor ini dapat bersifat jinak (non-kanker) maupun ganas (kanker). Berdasarkan asalnya, tumor otak dibedakan menjadi dua jenis, yaitu tumor otak primer yang berkembang langsung di otak atau jaringan sekitarnya, dan tumor otak sekunder (metastasis) yang berasal dari organ lain lalu menyebar ke otak (McFaline-Figueroa & Lee, 2018). Pertumbuhan dan penyebaran tumor sangat bervariasi, bergantung pada jenis, lokasi, dan tingkat keganasannya. Kondisi ini berpengaruh langsung terhadap fungsi sistem saraf serta menentukan pilihan terapi yang akan diberikan (McFaline-Figueroa & Lee, 2018).

Penyebab tumor otak belum sepenuhnya diketahui. Tumor otak primer biasanya berawal dari mutasi DNA sel normal di otak yang menyebabkan sel-sel tersebut tumbuh dan membelah secara tidak terkendali hingga membentuk massa abnormal. Tumor sekunder lebih sering dijumpai pada orang dewasa, di mana kanker dari organ lain menyebar ke otak (Haq, 2022). Selain faktor genetik, risiko tumor otak juga dapat meningkat akibat paparan radiasi, bahan kimia berbahaya, serta riwayat keluarga dengan penyakit serupa (Haq, 2022). Beberapa jenis tumor otak primer yang umum antara lain *glioma*, *meningioma*, *acoustic neuroma*, *pituitary adenoma*, *medulloblastoma*, *germ cell tumor*, dan *craniopharyngioma*.

2.3 Model U-Net

U-Net merupakan salah satu arsitektur *deep learning* berbasis *Convolutional Neural Network* (CNN) yang dirancang khusus untuk segmentasi citra biomedis. Arsitektur ini pertama kali dikembangkan untuk segmentasi sel

mikroskopis, namun kini banyak digunakan pada citra MRI otak karena kemampuannya dalam menghasilkan segmentasi spasial yang detail. Keunggulan U-Net terletak pada strukturnya yang simetris berbentuk *encoder-decoder*, di mana bagian *encoder* mengekstraksi fitur citra, sedangkan bagian *decoder* merekonstruksi peta fitur menjadi segmentasi akhir (Wardhani dkk., 2023).



Gambar 2. 1 Arsitektur U-Net (Ronneberger dkk., 2015)

Pada Gambar 2. 1 merupakan contoh arsitektur U-Net terdiri atas beberapa komponen utama:

1. *Encoder (Contracting Path)*

Encoder berfungsi untuk menangkap konteks dan mengekstraksi fitur-fitur penting dari citra *input*. Pada bagian ini, setiap blok *encoder* terdiri atas dua lapisan konvolusi Conv2D berukuran 3×3 yang diikuti oleh *Batch Normalization* untuk menjaga kestabilan pelatihan dan mempercepat konvergensi. Setelah proses konvolusi, dilakukan Max Pooling (2×2) untuk mengurangi resolusi spasial citra sekaligus meningkatkan jumlah channel fitur. Melalui tahapan ini, jaringan dapat mempelajari representasi yang

semakin abstrak dan kompleks seiring bertambahnya kedalaman lapisan. Dengan demikian, jalur *encoder* berperan sebagai bagian yang menangkap pola-pola global dan fitur semantik dari citra yang dianalisis (Sasongko dkk., 2024; Walsh dkk., 2022).

2. *Bottleneck* (Jembatan *Encoder-Decoder*)

Bagian *bottleneck* merupakan area terdalam dalam arsitektur U-Net yang menghubungkan jalur *encoder* dan *decoder*. Pada tahap ini, resolusi citra berada pada titik terendah namun mengandung representasi fitur paling padat secara semantik. Strukturnya terdiri dari dua lapisan Conv2D (3×3) dan *Batch Normalization* untuk memperkuat fitur yang diekstraksi dari *encoder*, sekaligus menyimpan representasi semantik global yang digunakan kembali pada proses rekonstruksi di jalur *decoder*.

3. *Decoder* (*Expansive Path*)

Decoder berfungsi untuk mengembalikan resolusi spasial citra ke ukuran semula sambil mempertahankan informasi semantik yang diperoleh dari *encoder*. Pada setiap blok *decoder*, dilakukan proses Conv2DTranspose (2×2) untuk melakukan *upsampling* terhadap fitur, diikuti oleh operasi *Concatenate* (*skip connection*) yang menggabungkan fitur dari jalur *encoder* pada level resolusi yang sama. Fitur gabungan ini kemudian diproses oleh dua lapisan Conv2D (3×3) dan *Batch Normalization* untuk memperhalus hasil rekonstruksi. *Skip connection* memiliki peran penting dalam arsitektur U-Net karena membantu mengembalikan detail spasial beresolusi tinggi yang hilang selama proses *downsampling*. Dengan adanya

koneksi ini, model dapat menghasilkan segmentasi dengan batas objek yang lebih akurat dan.

4. Lapisan *Output* (*Output Layer*)

Tahap akhir arsitektur U-Net adalah lapisan *output*, yang menggunakan Conv2D (1×1) untuk mengubah jumlah channel fitur menjadi jumlah kelas segmentasi yang diinginkan. Pada kasus segmentasi biner seperti deteksi tumor otak, lapisan ini menghasilkan satu channel yang mewakili probabilitas setiap piksel termasuk dalam kelas target (misalnya, tumor atau no tumor). Fungsi aktivasi Sigmoid digunakan untuk menghasilkan nilai probabilitas antara 0 dan 1 bagi setiap piksel *output*, sedangkan untuk kasus multi-kelas biasanya digunakan Softmax. Lapisan ini berperan penting untuk memetakan hasil akhir jaringan ke bentuk segmentasi citra yang dapat diinterpretasikan secara langsung.

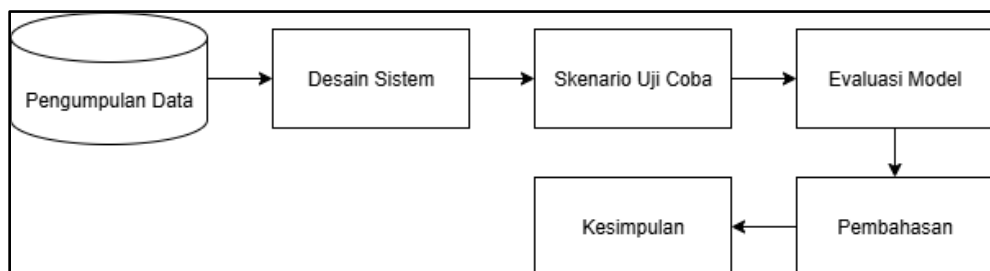
Dengan mekanisme ini, U-Net mampu menghasilkan segmentasi dengan presisi tinggi meskipun pada citra yang memiliki noise atau kontras rendah (Sasongko dkk., 2024).

BAB III

DESAIN DAN IMPLEMENTASI

3.1 Prosedur Penelitian

Prosedur penelitian yang diusulkan bertujuan untuk memudahkan dalam penyampaian gambaran terkait alur penelitian yang akan dilakukan. Tahap pertama dalam prosedur penelitian ini yakni tahap pengumpulan data. Dilanjutkan dengan desain sistem, image citra MRI akan melalui tahap penelitian sehingga dapat diproses untuk segmentasi tumor otak. Tahap berikutnya ada skenario uji coba dan evaluasi model yang dilakukan dengan tujuan agar dapat memaksimalkan hasil dari penelitian ini. Usulan prosedur penelitian dapat dilihat pada Gambar 3. 1.



Gambar 3. 1 Prosedur Penelitian

3.2 Pengumpulan Data

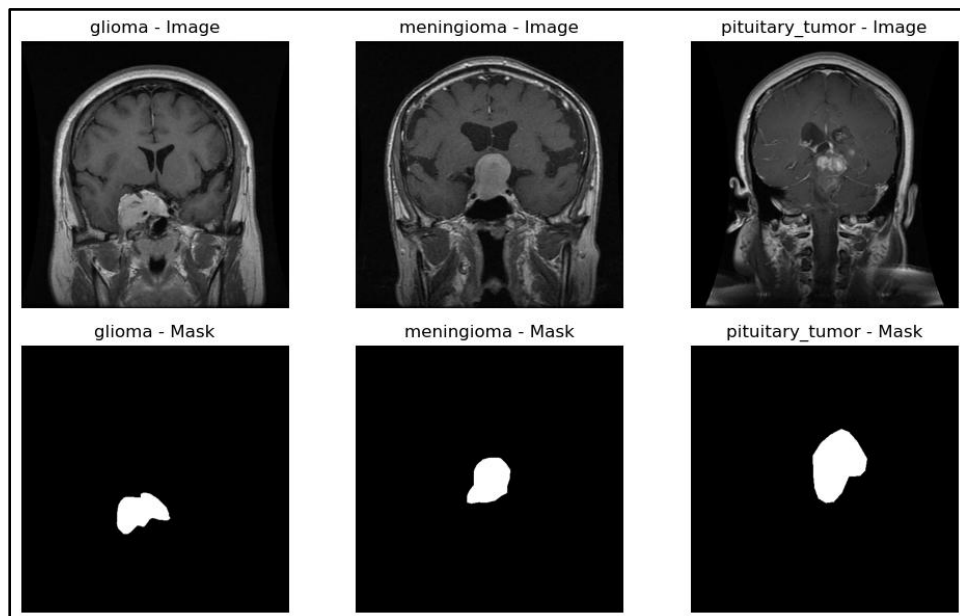
Tahap pertama adalah tahap pengumpulan data *Brain Tumor Segmentation*. Data tersebut diambil dari Kaggle dengan *author* Nikhil Tomar 2022 yang bersumber dari *Figsahre Version 1* Tahun 2015 milik Jun Cheng dengan total banyak data adalah 3064 image (Cheng, brain tumor segmentation, 2015). Dataset tersebut memiliki 2 folder yaitu image dan mask. Dari kedua folder tersebut

dibagi atas 3 klasifikasi tumor yaitu dan disimpan dalam folder bernama *Glioma*, *Meningioma*, dan *Pituitary Tumor*.

Tabel 3. 1 Pengelompokan Tumor

No.	Jenis Tumor	<i>image</i>	<i>mask</i>
1.	<i>Glioma</i>	708	708
2.	<i>Meningioma</i>	718	718
3.	<i>Pituitary Tumor</i>	1638	1638

Berdasarkan Tabel 3. 1, tumor otak diklasifikasikan ke dalam tiga jenis utama yaitu *Glioma*, *Meningioma*, dan *Pituitary Tumor*. *Glioma* merupakan jenis tumor yang paling umum ditemukan di otak dan berasal dari sel glial yang berfungsi mendukung aktivitas neuron, sedangkan *Meningioma* tumbuh pada meninges, selaput pelindung otak dan sumsum tulang belakang yang umumnya bersifat jinak namun dapat menekan jaringan otak di sekitarnya. Masing-masing contoh tumor dapat dilihat pada gambar berikut.

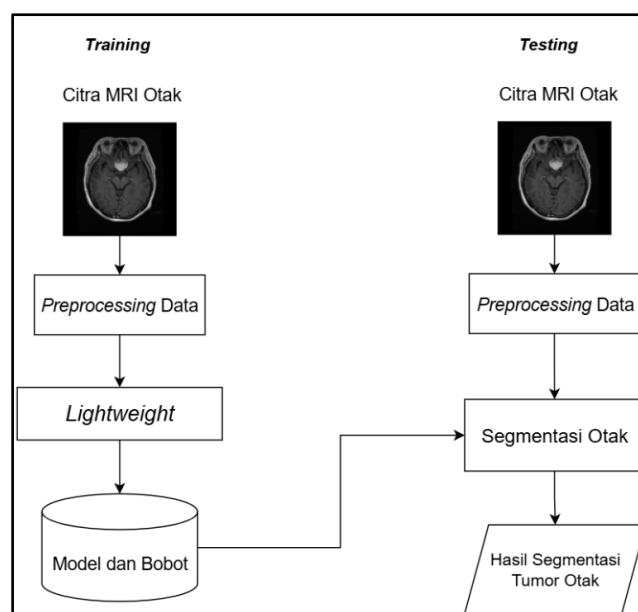


Gambar 3. 2 *Sample Tumor Otak*

Pada Gambar 3. 2 adalah contoh dari masing-masing jenis tumor dari *Pituitary* tumor berkembang pada kelenjar *pituitary*, yang berperan dalam mengatur keseimbangan hormon tubuh dan dapat menimbulkan gangguan hormonal akibat produksi berlebih atau kekurangan hormon tertentu (Hussain dkk., 2025). Dalam penelitian ini, dataset yang digunakan terdiri atas 708 citra dan mask untuk *Glioma*, 718 citra dan mask untuk *Meningioma*, serta 1638 citra dan mask untuk *Pituitary* tumor.

3.3 Desain Sistem

Dalam melakukan penelitian, diperlukan desain sistem yang dibangun sebelum penelitian dilaksanakan. Desain sistem ini menggambarkan alur penelitian yang dikembangkan, yaitu segmentasi tumor otak menggunakan model usulan *Lightweight* U-Net berbasis citra MRI otak. Data yang digunakan telah dikelompokkan menjadi tiga jenis, yaitu *Glioma*, *Meningioma*, dan *Pituitary* Tumor.



Gambar 3. 3 Skema *Training* dan *Testing*

Gambar 3. 3 merupakan Skema *Training* dan *Testing* yang nantinya akan digunakan dalam penelitian ini. Tahap pertama pada rancangan desain sistem adalah *preprocessing* data, di mana citra mengalami proses *resize image*. Dilakukan implementasi model usulan *Lightweight* U-Net yang mencakup beberapa langkah utama, antara lain *input* citra MRI otak dan ekstraksi fitur pada proses *encoder* melalui operasi inti konvolusi dua dimensi (*2D Convolution*) dengan parameter *filters* dan fungsi aktivasi ReLU, serta terjadi proses Max-Pooling. Tahap terakhir meliputi proses *Training* dan *Testing* model usulan *Lightweight* U-Net, di mana data dibagi menjadi tiga bagian, yaitu data *Training*, data *Validasi*, dan data *Testing*, serta dilakukan optimasi bobot terbaik menggunakan *Optimizer Adam*. Desain sistem yang diusulkan dalam penelitian ini.

3.3.1 Input Image

Data citra MRI yang berupa gambar digunakan sebagai data utama dalam sistem yang akan dibangun. Setiap citra tersebut berisi berbagai jenis atau kondisi tumor otak yang akan dianalisis untuk mendeteksi keberadaan tumor. Sebelum digunakan, data citra terlebih dahulu melalui tahap pengumpulan dan pengelompokan berdasarkan jenis atau kondisi tumor otak. Setelah itu, data yang telah siap digunakan menjadi sumber *input* utama pada proses *Training* dan *Testing* model, sehingga sistem dapat mempelajari pola serta melakukan prediksi secara optimal.

3.3.2 Preprocessing Citra

Pada tahap pra-pemrosesan citra, proses perubahan ukuran (*resize*) dilakukan menggunakan metode *bilinear interpolation* yang diimplementasikan melalui fungsi `cv2.resize()` pada pustaka OpenCV dengan parameter `INTER_LINEAR`. Metode ini memperkirakan nilai intensitas piksel baru berdasarkan rata-rata berbobot dari empat piksel terdekat pada arah horizontal dan vertikal. Secara matematis, nilai intensitas hasil interpolasi dapat dituliskan pada Persamaan 3.1 berikut:

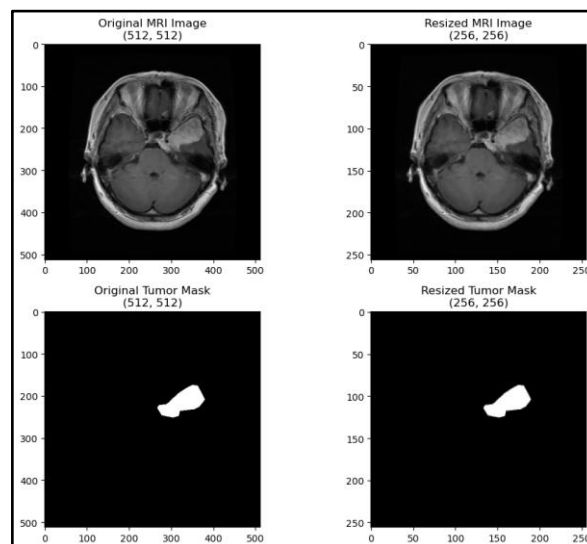
$$P(x) = (1 - \alpha)(1 - \beta)A + \alpha(1 - \beta)B + (1 - \alpha)\beta C + \alpha\beta D \quad (3.1)$$

Keterangan:

A, B, C, D = nilai intensitas empat piksel pada posisi; kiri atas, kanan atas, kiri bawah, kanan bawah

$\alpha = \frac{x-x_1}{x_2-x_1}$ = jarak relatif posisi piksel hasil terhadap arah horizontal

$\beta = \frac{y-y_1}{y_2-y_1}$ = jarak relatif posisi piksel hasil terhadap arah vertikal



Gambar 3. 4 *Resize Image*

Gambar 3. 4 merupakan hasil dari *resize* dengan menggunakan metode *bilinear interpolation*. Metode ini bekerja dengan menghitung kontribusi bobot dari

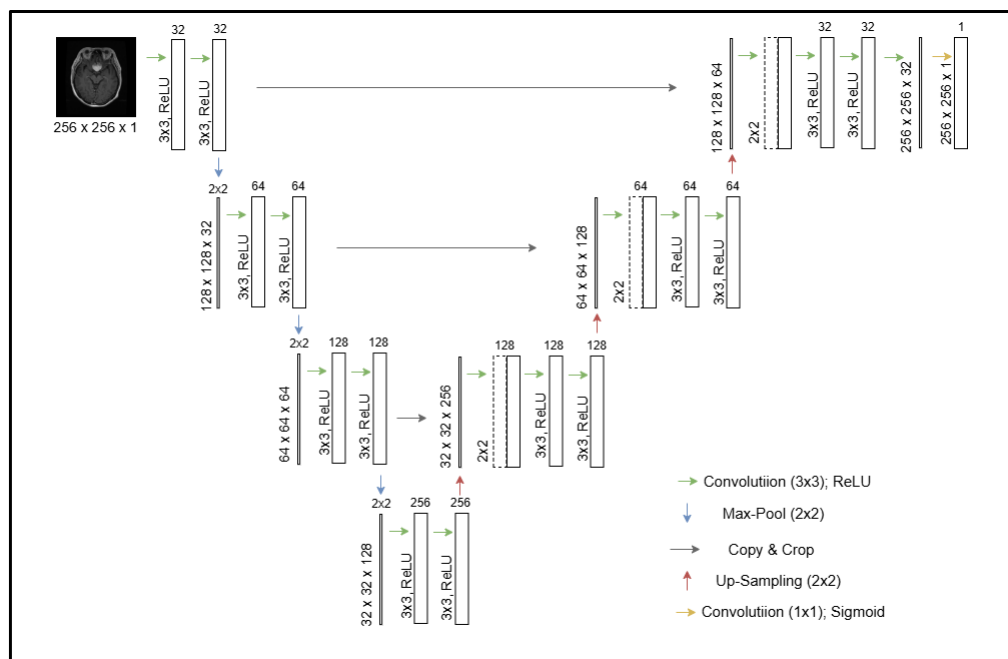
keempat piksel tetangga, kemudian mengombinasikannya untuk menghasilkan nilai intensitas piksel baru yang halus. Proses interpolasi ini mampu mempertahankan kualitas visual citra tanpa menyebabkan distorsi tepi yang signifikan, sehingga sering digunakan pada proses penskalaan gambar digital. Metode *bilinear interpolation* merupakan pengembangan dari interpolasi linear satu dimensi yang diterapkan dua kali pada arah x dan y untuk memperkirakan nilai intensitas piksel pada koordinat non-integer.

3.3.3 Arsitektur Usulan *Lightweight* U-Net

Lightweight U-Net yang digunakan dalam penelitian ini merupakan pengembangan dari arsitektur U-Net untuk segmentasi tumor otak pada citra MRI. Model ini bekerja berdasarkan prinsip jaringan saraf konvolusional dengan kernel berukuran 3×3 pada setiap lapisan konvolusi, sebagaimana diimplementasikan pada kelas `ManualConv2D` dengan parameter `kernel_size=3`. Setiap lapisan konvolusi menggunakan fungsi aktivasi ReLU dan inisialisasi bobot He Normal untuk mendukung stabilitas proses pelatihan. Struktur arsitektur disusun secara simetris antara encoder dan decoder dengan memanfaatkan mekanisme *skip connection*, sesuai dengan konsep U-Net yang diperkenalkan oleh Ronneberger dkk., (2015), yang menggabungkan jalur kontraksi dan ekspansi untuk mempertahankan informasi spasial pada berbagai tingkat resolusi (Ronneberger dkk., 2015).

Pada Gambar 3. 5 ditunjukkan bahwa encoder menerima masukan berupa citra MRI berukuran $256 \times 256 \times 1$ dalam format grayscale. Encoder terdiri atas tiga blok, di mana setiap blok memiliki dua lapisan konvolusi 3×3 dengan padding

SAME sehingga dimensi spasial fitur tetap terjaga sebelum dilakukan proses reduksi ukuran. Jumlah filter meningkat secara bertahap yaitu 32, 64, dan 128 agar model mampu mempelajari fitur yang semakin kompleks. Setelah dua kali konvolusi, dilakukan operasi max-pooling 2×2 dengan stride 2 untuk mengurangi dimensi spasial menjadi setengahnya. Setelah melalui tiga blok encoder, model memasuki bagian bottleneck yang terdiri dari dua lapisan konvolusi dengan 256 filter untuk menangkap representasi fitur terdalam sebelum proses decoding.



Gambar 3. 5 Arsitektur Usulan *Lightweight U-Net*

Pada tahap decoder, fitur direkonstruksi kembali menuju resolusi awal citra menggunakan operasi Conv2DTranspose dengan kernel 2×2 dan stride 2. Fitur hasil upsampling kemudian digabungkan dengan fitur dari encoder melalui *skip connection* sebelum diproses kembali oleh dua lapisan konvolusi 3×3 dengan aktivasi ReLU. Pada lapisan akhir digunakan konvolusi 1×1 dengan fungsi aktivasi

sigmoid untuk menghasilkan peta segmentasi biner. Proses pelatihan model dilakukan menggunakan optimizer Adam, yang memperbarui bobot jaringan berdasarkan estimasi momen pertama dan kedua dari gradien sehingga proses optimisasi menjadi lebih stabil dan efisien (Ramadhan & Hernadi, 2025). Secara matematis, ukuran keluaran dari operasi *transposed convolution* ini dapat ditentukan dengan Persamaan 3.2:

$$O = S \times (I - 1) + K - 2P \quad (3.2)$$

Keterangan:

O = nilai ukuran *output height/weight*

I = nilai ukuran *input height/weight*

S = nilai strider

K = nilai kernel size

P = nilai padding

Hasil *upsampling* kemudian digabungkan dengan *feature map* dari tahapan encoder sebelumnya untuk mengembalikan detail spasial yang sempat berkurang akibat proses penyusutan ukuran. Penggabungan ini dilakukan melalui mekanisme *skip connection*, yaitu penghubung langsung antara encoder dan decoder pada tingkat kedalaman fitur yang sama. Pada mekanisme tersebut digunakan operasi *concatenate*, yaitu proses menyatukan dua *feature map* agar informasi detail dari encoder dapat dimanfaatkan kembali pada tahap rekonstruksi citra. Setelahnya, dilakukan dua kali konvolusi dengan nama *method* ManualConv2D berukuran 3×3 dengan aktivasi ReLU dan jumlah *filter* yang menurun (128, 64, dan 32) pada tiap blok *decoder*. Lapisan akhir Conv2D 1×1 beraktivasi Sigmoid menghasilkan citra segmentasi biner $256 \times 256 \times 1$ yang merepresentasikan probabilitas area tumor. Model dikompilasi dengan *Optimizer* Adam, *loss Binary Cross-Entropy*, dan metrik

Dice Coefficient. Rancangan ini tetap menjaga simetri antara *encoder* dan *decoder* khas arsitektur U-Net namun dengan parameter lebih ringan, sehingga efisien untuk dataset berukuran sedang hingga terbatas.

Struktur *U-shape* pada *Lightweight* U-Net memungkinkan aliran informasi berlangsung secara efisien melalui dua arah, yaitu *forward pass* dan *backward pass*. Pada *forward pass*, data masukan diproses melalui operasi konvolusi di *encoder* hingga mencapai bagian terdalam jaringan, kemudian direkonstruksi kembali pada *decoder* untuk menghasilkan peta segmentasi. Pada tahap ini, informasi detail dari *encoder* dikembalikan ke *decoder* melalui jalur khusus yang dikenal sebagai *skip connection*, dan di dalam jalur tersebut digunakan operasi *concatenate* untuk menggabungkan *feature map* dari *encoder* dan *decoder* sehingga detail spasial tetap terjaga. Nilai keluaran dibandingkan dengan *ground-truth* menggunakan fungsi *loss*, seperti *Binary Cross-Entropy* atau *Dice Loss* untuk menghitung kesalahan prediksi. Proses *backward pass* kemudian dilakukan untuk menghitung gradien kesalahan yang mengalir mundur melalui seluruh jaringan, termasuk jalur *skip connection* dan operasi *concatenate*, sehingga bobot pada *encoder* dan *decoder* dapat diperbarui secara bersamaan. Dengan mekanisme ini, U-Net mampu mempertahankan keseimbangan antara pemahaman konteks yang luas dan detail spasial yang presisi selama pelatihan.

Adapun rumus dari operasi konvolusi, *max-pooling*, dan fungsi aktivasi diimplementasikan pada setiap lapisan untuk melakukan proses ekstraksi fitur dari citra *input*. Tujuan dari lapisan konvolusi adalah untuk menerapkan *filter* atau

kernel pada citra *input* sehingga menghasilkan peta fitur (*feature map*). Berikut ini merupakan rumus dari Persamaan 3.3 operasi konvolusi:

$$Y_{(i,j,k)} = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} \sum_{c=0}^{C_{in}-1} X(i+m, j+n, c) \cdot W(m, n, c, k) + b_k \quad (3.3)$$

Keterangan:

$Y(i, j, k)$ = nilai piksel *output* pada posisi (i, j) dan filter k
 $X(i + m, j + n, c)$ = nilai piksel *input* pada posisi $(i + m, j + n, c)$ di sekitar titik (i, j)
 $W(m, n, c, k)$ = bobot kernel dengan ukuran $M \times N$ untuk channel c dan channel k
 b_k = bias untuk channel k

Operasi Persamaan 3.3 konvolusi dua dimensi (*2D Convolution*) merupakan proses utama dalam jaringan saraf konvolusional (CNN) yang berfungsi untuk mengekstraksi fitur spasial dari citra masukan. Secara matematis, proses konvolusi dapat dinyatakan dari Persamaan 3.3 di mana $Y_{(i,j,k)}$ merupakan nilai keluaran pada posisi spasial (i, j) dan channel ke- k ; $X(i + m, j + n, c)$ adalah nilai piksel citra masukan pada posisi $(i + m, j + n)$ dengan channel ke- c ; $W(m, n, c, k)$ merupakan bobot kernel berukuran $M \times N$ yang menghubungkan channel masukan ke- c dengan filter ke- k ; dan b_k adalah bias untuk channel ke- k . Dengan cara ini, setiap piksel keluaran merupakan hasil perhitungan berbobot dari piksel-piksel di sekitar posisi tertentu pada citra masukan. Proses ini memungkinkan jaringan untuk mengenali fitur penting seperti tepi, tekstur, dan pola visual lainnya.

Input: inputs $\in \mathbb{R}[H \times W \times C_{in}]$, filters, kernel_size, activation
Output: conv — hasil konvolusi 2D dengan bias dan aktivasi
1: Procedure ManualConv2D (filters, kernel_size, activation)
2: filters \leftarrow jumlah filter konvolusi
3: kernel_size \leftarrow ukuran kernel
4: activation \leftarrow fungsi aktivasi (opsional)
5:
6: {Parameter Initialization}
7: build(input_shape)
8: Cin \leftarrow input_shape[-1]
9: kernel \leftarrow HeNormal([kernel_size, kernel_size, Cin, filters])
10: bias \leftarrow Zeros([filters])

```

11:
12: {Forward Pass}
13:  call(inputs)
14:  conv ← Conv2D (inputs, kernel, strides=1, padding='SAME')
15:  conv ← conv + bias
16:  if activation ≠ None then
17:    conv ← activation(conv)
18:  end if
19:  return conv
20: End Procedure

```

Gambar 3. 6 Algoritma 1 *Convolution Operation* 2D (ManualConv2D)

Hubungan antara persamaan tersebut dan Algoritma 1 *Convolution Operation* 2D terletak pada tahapan implementasi algoritmiknya. Pada bagian inisialisasi parameter, variabel C_{in} merepresentasikan jumlah channel masukan (C_{in}), sedangkan kernel dan bias diinisialisasi menggunakan metode *HeNormal* dan *Zeros* yang secara matematis setara dengan parameter W dan b_k dalam persamaan konvolusi. Tahap *forward pass*, instruksi Conv2D (inputs, kernel, strides=1, padding='SAME') melaksanakan operasi penjumlahan bertingkat sesuai rumus, di mana setiap posisi (i, j) dan filter k menghasilkan nilai konvolusi. Perintah $conv \leftarrow conv + bias$ menambahkan nilai bias b_k ke hasil konvolusi, sedangkan $activation(conv)$ (jika digunakan) menerapkan fungsi aktivasi non-linear untuk memperkuat kemampuan representasi model. Penggunaan $padding='SAME'$ menunjukkan adanya *zero-padding* di sekitar citra masukan, sehingga ukuran keluaran sama dengan ukuran masukan dan setiap operasi konvolusi tetap terdefinisi. Dengan demikian, Algoritma 1 *Convolution Operation* 2D merupakan representasi komputasional langsung dari rumus matematis konvolusi dua dimensi, di mana aspek teoritis dan algoritmik saling melengkapi untuk menggambarkan mekanisme ekstraksi fitur pada jaringan konvolusional.

Setelah melalui proses aktivasi ReLU, hasil konvolusi kemudian diproses menggunakan operasi *max-pooling 2D* untuk menyederhanakan representasi fitur. Pada tahap ini, nilai maksimum dari setiap area tertentu pada peta fitur (*feature map*) diambil untuk mempertahankan informasi yang paling dominan. Langkah ini tidak hanya mengurangi dimensi data, tetapi juga membantu menjaga fitur penting yang telah diaktifkan oleh ReLU. Dengan demikian, model menjadi lebih efisien dalam mengenali pola utama dan lebih robust terhadap pergeseran posisi objek pada citra. Selain itu, *max-pooling* juga mempercepat proses komputasi dan mengurangi risiko model terlalu bergantung pada detail kecil yang tidak signifikan. Berikut merupakan Persamaan 3.4 dari *Max-Pooling 2D*:

$$u_{i,j}^{(k)} = \max_{p,q \in P_{i,j}} z_{p,q}^{(k)} \quad (3.4)$$

Keterangan:

$u_{i,j}^{(k)}$ = nilai hasil pooling (*ouput*) pada posisi (i, j) di channel k

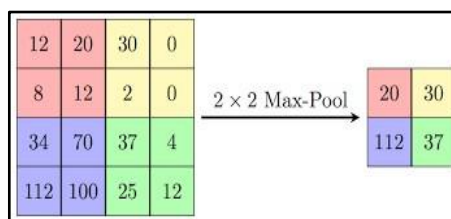
$P_{i,j}$ = *region* (area) atau *window* pada *input* yang berkaitan dengan posisi *input* (i, j)

Operator *max* = memiliki nilai maksimum dari area $P_{i,j}$

$z_{p,q}^{(k)}$ = nilai *input* dari *feature map* pada posisi (p, q)

Operasi *max pooling* merupakan salah satu tahap penting dalam jaringan saraf konvolusional (CNN) yang berfungsi untuk mereduksi dimensi spasial dari *feature map* tanpa menghilangkan informasi penting ilustrasi terdapat pada Gambar 3. 6. Secara matematis, operasi *max pooling* dapat dinyatakan dari Persamaan 3.4 di mana $u_{i,j}^{(k)}$ adalah nilai hasil *pooling* pada posisi spasial (i, j) di channel ke- k ; $z_{p,q}^{(k)}$ merupakan nilai piksel dari *feature map* input pada posisi (p, q) ; dan $P_{i,j}$ menunjukkan area (*region* atau *window*) pada input yang berhubungan dengan

posisi output (i, j) . Operator berfungsi memilih nilai maksimum dari seluruh elemen dalam area $P_{i,j}$, sehingga hanya nilai paling dominan yang dipertahankan pada hasil *pooling*. Dengan demikian, operasi *max pooling* dapat menyaring fitur penting, meningkatkan ketahanan terhadap translasi kecil, dan mengurangi beban komputasi pada lapisan berikutnya karena ukuran data menjadi lebih kecil.



Gambar 3. 7 Max-Polling (Haq, 2022)

Input: $inputs \in \mathbb{R}[H \times W \times Cin]$, $pool_size$, $strides$
Output: $pooled_output$ — hasil pooling citra

- 1: Procedure ManualMaxPooling2D ($pool_size$, $strides$)
- 2: $pool_size \leftarrow$ ukuran jendela pooling (default = 2)
- 3: $strides \leftarrow$ langkah pergeseran pooling (default = 2)
- 4:
- 5: {Pooling Process}
- 6: call(inputs)
- 7: $pooled_output \leftarrow$ MaxPool2D(inputs)
- 8: dengan $ksize = pool_size$,
- 9: $strides = strides$,
- 10: $padding = 'SAME'$
- 11: return $pooled_output$
- 12: End Procedure

Gambar 3. 8 Algoritma 2 Max-Pooling Operation (ManualMaxPooling2D)

Algoritma 2 *Max-Pooling Operation* merepresentasikan implementasi algoritmik langsung dari persamaan matematis tersebut. Pada tahap inisialisasi parameter, variabel $pool_size$ menentukan ukuran jendela *pooling* ($P_{i,j}$), sedangkan $strides$ menunjukkan langkah pergeseran jendela saat proses ekstraksi area fitur. Bagian *Pooling Process* kemudian memanggil fungsi $MaxPool2D(inputs)$ dengan parameter $ksize = pool_size$, $strides = strides$, dan $padding = 'SAME'$. Langkah ini

merealisasikan proses pencarian nilai maksimum dari setiap area lokal pada *feature map* input sesuai rumus $u_{i,j}^{(k)} = \max(z_{p,q}^{(k)})$. Nilai hasil *pooling* disimpan dalam variabel `pooled_output`, yang kemudian menjadi keluaran fungsi. Penggunaan parameter `padding='SAME'` memastikan ukuran keluaran tetap proporsional terhadap input, terutama pada tepi citra. Dengan demikian, pseudocode *Max-Pooling Operation* merupakan representasi komputasional dari prinsip dasar *max pooling* yang secara matematis dijabarkan dalam Persamaan 3.4, di mana proses pemilihan nilai maksimum dari setiap area lokal diimplementasikan secara sistematis untuk menghasilkan peta fitur beresolusi lebih rendah namun tetap informatif.

3.3.4 Fungsi Aktivasi

Penelitian ini menerapkan dua fungsi aktivasi utama dalam arsitektur *Lightweight U-Net*, yaitu ReLU dan Sigmoid. Fungsi-fungsi ini berperan dalam memperkenalkan *non-linearitas* ke dalam model serta mengatur hasil keluaran pada setiap tahap pemrosesan. Penjelasan lebih lanjut mengenai kedua fungsi aktivasi tersebut disajikan berikut ini.

a. Fungsi ReLU (*Rectified Linear Unit*)

Fungsi aktivasi ReLU (*Rectified Linear Unit*) digunakan setelah proses konvolusi untuk menambahkan sifat *non-linear* pada model. Fungsi ini berperan penting dalam mengubah nilai negatif pada hasil konvolusi menjadi nol, sementara nilai positif dipertahankan. Dengan cara ini, ReLU membantu model mempelajari representasi data yang lebih kompleks dan mempercepat proses pelatihan karena sifatnya yang sederhana namun efektif. Secara

matematis, fungsi aktivasi ReLU dinyatakan dengan Persamaan 3.5 dan grafik pada Gambar 3. 7:

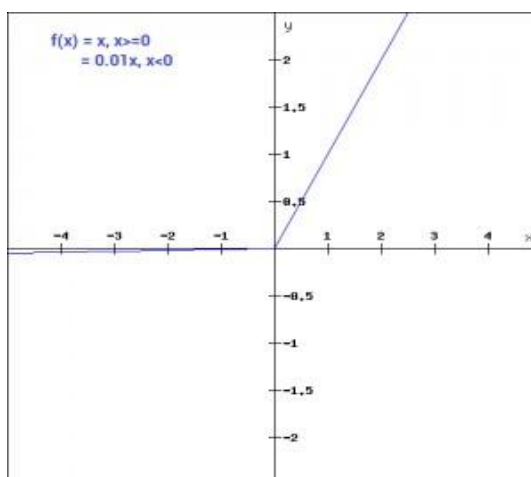
$$ReLU(x) = f(x) = \max(0, x) \quad (3.5)$$

Keterangan:

x = nilai *input* yang berasal dari hasil operasi konvolusi

$f(x)$ = nilai hasil aktivasi yang akan diteruskan pada layer berikutnya

$\max(0, x)$ = nilai maksimum antara 0 dan x : jika x positif akan bernilai x ; jika x negative bernilai 0



Gambar 3. 9 Diagram ReLU (Sharma dkk., 2020)

b. Fungsi Sigmoid

Selain fungsi aktivasi ReLU, penelitian ini juga menggunakan fungsi aktivasi Sigmoid pada lapisan *output*. Fungsi Sigmoid merupakan bentuk khusus dari fungsi logistik yang digunakan untuk mengubah nilai *input* menjadi rentang antara 0 dan 1. Hal ini menjadikan Sigmoid sangat cocok digunakan pada tahap akhir model segmentasi, di mana nilai keluaran merepresentasikan probabilitas antara dua kelas, yaitu area yang mengandung tumor dan area normal. Secara matematis, fungsi aktivasi Sigmoid dapat dinyatakan dengan Persamaan 3.6 dan grafik Gambar 3. 8 berikut:

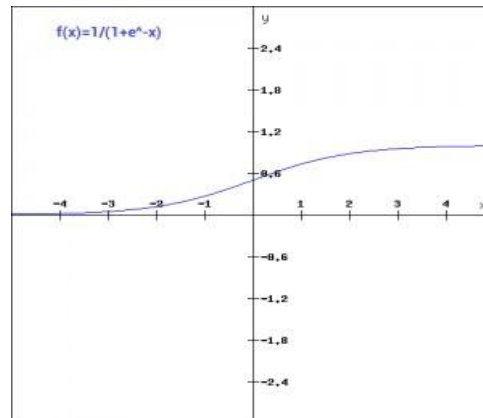
$$f(x) = \frac{1}{1 + e^{-x}} \quad (3.6)$$

Keterangan:

$f(x)$ = hasil aktivasi dari *neuron* setelah *input* x diproses

x = nilai *input* dari operasi konvolusi

e^{-x} = nilai eksponensial negatif dari *input* yang menunjukkan bahwa fungsi sigmoid menurun secara eksponensial.



Gambar 3. 10 Diagram Sigmoid (Sharma dkk., 2020)

Setelah seluruh proses *Convolution Operation 2D*, *Max-Pooling Operation*, dan penerapan fungsi aktivasi selesai dilakukan, disusul tahap membangun arsitektur model utama. Proses ini menghasilkan rancangan *Lightweight U-Net* yang merupakan hasil integrasi dari lapisan konvolusi, pooling, dan aktivasi yang telah dirancang secara manual sebelumnya. Melalui tahapan tersebut, terbentuklah struktur jaringan U-Net yang lengkap, di mana setiap lapisan memiliki peran tersendiri dalam mengekstraksi fitur dan merekonstruksi citra hasil segmentasi. Adapun rancangan akhir dari model usulan *Lightweight U-Net* yang diusulkan dapat dilihat pada algoritma berikut.

```

Input: input_shape = (256, 256, 1)
Output: model — Manual U-Net for image segmentation
1: Procedure build_manual_unet(input_shape)
2:   inputs ← Input(input_shape)
3: {Encoder Stage}
4: (s1, p1) ← build_encoder_block (inputs, 32)
5: (s2, p2) ← build_encoder_block (p1, 64)
6: (s3, p3) ← build_encoder_block (p2, 128)
7:
8: {Bottleneck Stage}
9:   b1 ← ManualConv2D (256, activation='relu') (p3)
10:  b1 ← ManualConv2D (256, activation='relu') (b1)
11:
12: {Decoder Stage}
13:  d1 ← build_decoder_block (b1, s3, 128)
14:  d2 ← build_decoder_block (d1, s2, 64)
15:  d3 ← build_decoder_block (d2, s1, 32)
16:
17: {Output Layer}
18:  outputs ← Conv2D (1, (1,1), activation='sigmoid')(d3)
19:
20: {Model Formation}
21:  model ← Model (inputs, outputs)
22:
23:  return model
24: End Procedure

```

Gambar 3. 11 Algoritma 3 Membangun U-Net *Construction* (*build_manual_unet*)

Pada tahap *encoder*, Algoritma 3 Membangun U-Net *Construction* menggunakan rangkaian operasi konvolusi 2D yang secara matematis mengikuti Persamaan 3.3 tentang konvolusi CNN. Operasi ini membaca pola penting pada citra dan menghasilkan fitur baru dengan detail yang lebih kaya. Setelah proses konvolusi selesai, setiap blok *encoder* melakukan *max-pooling*, yaitu teknik untuk mengecilkan ukuran citra sambil mempertahankan informasi terpenting, sebagaimana dijelaskan pada Persamaan 3.4. Kombinasi konvolusi dan *max-pooling* ini membuat model mampu menangkap fitur visual secara bertahap, mulai dari pola sederhana hingga representasi spasial yang lebih abstrak. Pada setiap tahap, diterapkan fungsi aktivasi ReLU pada Persamaan 3.5 yang berperan

mengenali pola non-linear dan mencegah hilangnya gradien selama proses pelatihan.

Model masuk ke *bottleneck*, yaitu lapisan terdalam yang memuat representasi fitur paling kompleks. Berikutnya, bagian *decoder* bekerja dengan melakukan *upsampling* untuk mengembalikan ukuran citra seperti semula, kemudian menggabungkan kembali fitur dari *encoder* agar detail yang sebelumnya diperkecil dapat dipulihkan. Operasi konvolusi kembali digunakan pada setiap blok *decoder* dengan prinsip yang sama seperti pada *encoder*. Pada tahap akhir, U-Net menggunakan fungsi aktivasi Sigmoid, yang dijelaskan dalam Persamaan 3.6, untuk menghasilkan nilai antara 0 dan 1 pada setiap piksel. Nilai ini membentuk peta segmentasi akhir yang membedakan area tumor dari jaringan normal. Melalui alur ini, Algoritma 3 Membangun U-Net *Construction* menjadi representasi langsung dari serangkaian persamaan matematis yang mendasari proses ekstraksi fitur, penyusutan ukuran citra, aktivasi *non-linear*, dan pembentukan mask segmentasi.

3.3.5 Model Pembelajaran

Model pembelajaran pada penelitian ini bertujuan untuk melatih arsitektur *Lightweight* U-Net agar mampu mengenali pola dan karakteristik area tumor otak pada citra MRI. Proses pembelajaran terdiri dari dua tahap utama, yaitu *forward pass* dan *backward pass*.

1. *Forward Pass*

Pada tahap *forward pass*, data citra masukan berukuran $256 \times 256 \times 1$ melewati beberapa blok konvolusi pada bagian *encoder* untuk mengekstraksi fitur penting

melalui lapisan konvolusi 2D dengan fungsi aktivasi ReLU, kemudian dilanjutkan dengan operasi *max-pooling* 2D untuk melakukan *downsampling*. Hasil ekstraksi fitur dari encoder diteruskan ke lapisan *bottleneck* untuk menangkap representasi global citra. Pada bagian *decoder*, proses rekonstruksi citra dilakukan melalui Conv2DTranspose sebagai mekanisme *upsampling* yang kemudian digabungkan dengan *feature map* dari *encoder* menggunakan *skip connection*. Penggabungan ini dilakukan dengan operasi *concatenate*, yaitu proses menyatukan dua *feature map* agar informasi detail dari *encoder* tetap dipertahankan selama rekonstruksi. Lapisan terakhir menggunakan Conv2D berukuran 1×1 dengan aktivasi Sigmoid untuk menghasilkan peta segmentasi biner yang merepresentasikan area tumor otak.

```

Input: Citra MRI otak X ( $256 \times 256 \times 1$ )
Output: Peta segmentasi Y ( $256 \times 256 \times 1$ )
1: Procedure Forward_Pass_Manual_UNet(X)
2: {Encoder Stage}
3:  s1  $\leftarrow$  ManualConv2D (X, 32, activation='relu')
4:  s1  $\leftarrow$  ManualConv2D (s1, 32, activation='relu')
5:  p1  $\leftarrow$  ManualMaxPooling2D(s1)

6:  s2  $\leftarrow$  ManualConv2D (p1, 64, activation='relu')
7:  s2  $\leftarrow$  ManualConv2D (s2, 64, activation='relu')
8:  p2  $\leftarrow$  ManualMaxPooling2D(s2)

9:  s3  $\leftarrow$  ManualConv2D (p2, 128, activation='relu')
10: s3  $\leftarrow$  ManualConv2D (s3, 128, activation='relu')
11: p3  $\leftarrow$  ManualMaxPooling2D(s3)

12: {Bottleneck Stage}
13: b1  $\leftarrow$  ManualConv2D(p3, 256, activation='relu')
14: b1  $\leftarrow$  ManualConv2D(b1, 256, activation='relu')

15: {Decoder Stage}
16: d1  $\leftarrow$  Conv2DTranspose (b1, 128, kernel=(2,2), stride=2, padding='same')
17: d1  $\leftarrow$  concatenate (d1, s3)
18: d1  $\leftarrow$  ManualConv2D (d1, 128, activation='relu')
19: d1  $\leftarrow$  ManualConv2D (d1, 128, activation='relu')

20: d2  $\leftarrow$  Conv2DTranspose (d1, 64, kernel=(2,2), stride=2, padding='same')
21: d2  $\leftarrow$  concatenate (d2, s2)

```

```

22: d2 ← ManualConv2D (d2, 64, activation='relu')
23: d2 ← ManualConv2D (d2, 64, activation='relu')

24: d3 ← Conv2DTranspose (d2, 32, kernel=(2,2), stride=2, padding='same')
25: d3 ← concatenate (d3, s1)
26: d3 ← ManualConv2D (d3, 32, activation='relu')
27: d3 ← ManualConv2D (d3, 32, activation='relu')

28: {Output Layer}
29: outputs ← Conv2D (d3, 1, kernel=(1,1), activation='sigmoid')

30: return outputs
31: End Procedure

```

Gambar 3. 12 Algoritma 4 *Forward Pass* U-Net

Algoritma 4 *Forward Pass* U-Net ini menggambarkan bagaimana model usulan *Lightweight* U-Net memproses citra MRI otak secara bertahap untuk menghasilkan peta segmentasi. Pertama, citra dimasukkan ke bagian *encoder*, yaitu rangkaian lapisan konvolusi yang bertugas membaca pola penting dalam gambar, seperti tepi, tekstur, atau area kontras tertentu. Setiap dua kali konvolusi diikuti dengan *max-pooling*, yaitu proses mengecilkan ukuran citra sembari mempertahankan informasi paling penting. Setelah melewati tiga tahap *encoder*, citra diproses di bagian *bottleneck*, yaitu lapisan terdalam yang merangkum informasi paling kompleks dan menjadi inti dari representasi citra. Dari titik ini, model bergerak ke *decoder*, yang berfungsi membesarkan kembali ukuran citra melalui *upsampling* (*Conv2DTranspose*) agar bentuknya kembali mendekati ukuran asli.

Pada proses pembesaran citra di *decoder*, hasil *upsampling* tidak bekerja sendirian. Model juga mengambil kembali fitur dari encoder dan menggabungkannya melalui operasi yang disebut *concatenate*. Secara sederhana, *concatenate* adalah proses menyatukan dua kelompok informasi:

satu dari tahap pembesaran (*decoder*), dan satu lagi dari tahap pengecilan (*encoder*). Penggabungan ini disebut sebagai *skip connection*, karena informasi “melompat” dari *encoder* langsung ke *decoder* tanpa berubah bentuk. Tujuannya adalah mempertahankan detail penting di awal proses yang mungkin hilang ketika citra diperkecil di *encoder*. Setelah fitur digabungkan, dua lapisan konvolusi kembali digunakan untuk memperbaiki detail dan menghasilkan citra yang lebih halus. Di bagian akhir, digunakan lapisan dengan fungsi aktivasi Sigmoid untuk menghasilkan peta segmentasi yang berisi nilai 0–1, yang menandakan apakah suatu piksel termasuk area tumor atau bukan.

2. *Backward Pass*

Tahap *backward pass* digunakan untuk memperbarui parameter model berdasarkan kesalahan prediksi terhadap *ground truth*. Nilai kesalahan dihitung menggunakan fungsi *loss Binary Cross-Entropy* (BCE) yang mengukur perbedaan antara hasil prediksi dan label sebenarnya. Fungsi *loss Binary Cross-Entropy* (Hussain et al., 2025). dirumuskan sebagai berikut oleh Persamaan 3.7:

$$BCE = \sum_{i=1}^N [y_i \cdot \log(p_i) + (1 - y_i) \cdot \log(1 - p_i)] \quad (3.7)$$

Keterangan:

N = jumlah pixel atau sampel yang dihitung

y_i = nilai *ground-truth* pada pixel ke- i (0: area no tumor, 1: area tumor)

p_i = nilai probabilitas prediksi model pada pixel i

$\log(.)$ = logaritma natural sebagai pengukur Tingkat kesalahan prediksi

Gradien dari fungsi *loss* dihitung dan dirambatkan ke seluruh lapisan jaringan menggunakan algoritma *backpropagation*. Optimisasi dilakukan dengan

Optimizer Adam, yang menggunakan dua rata-rata eksponensial dari gradien, yaitu pada Persamaan 3.8 dan Persamaan 3.9 adalah momen pertama (*first moment*) m_t dan momen kedua (*second moment*) u_t , dengan ρ_1 dan ρ_2 adalah koefisien peluruhan (bernilai 0.9 dan 0.999), $g \odot g$ adalah operasi elemen (perkalian Hadamard), g_t adalah gradien dari fungsi *loss* terhadap parameter pada iterasi ke- t (Soydaner, 2020). Persamaan 3.8 digunakan untuk menghitung rata-rata eksponensial dari gradien (*first moment estimate*) yang berfungsi menangkap arah pergerakan gradien untuk menghaluskan perubahan pembaruan bobot pada setiap iterasi dan Persamaan 3.9 menghitung rata-rata eksponensial kuadrat gradien (*second moment estimate*) yang digunakan untuk mengukur besarnya variansi gradien, sehingga pembaruan bobot tidak terlalu besar dan lebih stabil. kedua persamaan tersebut dapat dinyatakan:

$$m_t = \rho_1 m_{t-1} + (1 - \rho_1) g_t \quad (3.8)$$

$$u_t = \rho_2 m_{t-1} + (1 - \rho_2) g \odot g \quad (3.9)$$

Pada Persamaan 3.8 dan Persamaan 3.9 m_t merupakan estimasi momen pertama dari gradien, yaitu rata-rata eksponensial gradien yang merepresentasikan arah dominan perubahan nilai parameter, sedangkan u_t adalah estimasi momen kedua yang merepresentasikan rata-rata eksponensial kuadrat gradien untuk mengukur besarnya variansi gradien pada iterasi ke- t . Karena kedua nilai tersebut diinisialisasi dari nol, maka pada iterasi awal estimasinya masih bias ke nilai yang terlalu kecil. Jadi, Persamaan 3.10 dan Persamaan 3.11 ini adalah persamaan *Second Moment Update* dan *Bias-Corrected First Moment* yang

bertugas melakukan koreksi bias terhadap m_t dan u_t , sehingga diperoleh nilai momen pertama terkoreksi \hat{m}_t dan bias kedua terkoreksi \hat{u}_t yang lebih akurat dalam mencerminkan kondisi gradien sebenarnya dan siap digunakan untuk proses pembaruan bobot pada iterasi tersebut. Kedua momen tersebut memiliki bias terhadap nilai awal sehingga dilakukan koreksi bias menggunakan persamaan:

$$\hat{m}_t = \frac{m_t}{1-\rho_1^t} \quad (3.10)$$

$$\hat{u}_t = \frac{u_t}{1-\rho_2^t} \quad (3.11)$$

Pembaruan parameter model dilakukan berdasarkan nilai momen yang telah dikoreksi bias yaitu dipengaruhi nilai \hat{m}_t dan \hat{u}_t , dengan laju pembelajaran ϵ dan konstanta kecil δ , dimana ϵ adalah *learning rate* (LR), δ delta/konstanta kecil (10^{-7}) untuk mencegah pembagian dengan 0, $\Delta\theta$ merupakan perubahan nilai parameter yang dihitung pada iterasi ke- t . θ_t merupakan nilai bobot model pada iterasi ke- t , maka terbentuk persamaan:

$$\Delta\theta = -\epsilon \frac{\hat{m}_t}{\sqrt{\hat{u}_t + \delta}} \quad (3.12)$$

$$\theta_t = \theta_{t-1} + \Delta\theta \quad (3.13)$$

Maka, Persamaan 3.12 menghitung nilai perubahan bobot yang akan diterapkan. Pembaruan dihitung menggunakan rasio momen pertama yang telah dikoreksi terhadap akar kuadrat momen kedua yang dikoreksi, ditambah konstanta kecil untuk mencegah pembagian dengan nol, disebut dengan *Parameter Update Value* $\Delta\theta$ dan Persamaan 3.13 memperbarui bobot jaringan pada iterasi ke- t dengan

menambahkan nilai pembaruan bobot $\Delta\theta$ pada bobot sebelumnya, *Final Weight*

Update Rule.

<p>Input: (X_train, Y_train), Parameter (W, b), ϵ, ρ_1, ρ_2, δ, m_t, u_t, \hat{m}_t, \hat{u}_t, $\Delta\theta$</p> <p>Output: Parameter model yang diperbarui (W', b')</p> <ol style="list-style-type: none"> 1: Bagi data menjadi batch berukuran 16 2: Inisialisasi momen pertama $m_t = 0$ dan momen kedua $u_t = 0$ 3: Untuk setiap epoch: 4: Untuk setiap batch (X_batch, Y_batch): 5: Y_pred \leftarrow Forward_Pass_Manual_UNet(X_batch) 6: Hitung <i>loss</i> L menggunakan Persamaan (3.13) 7: Hitung gradien $g_t = \partial L / \partial Y_{\text{pred}}$ 8: Hitung arah pergerakan (momentum) $m_t \leftarrow \rho_1 * m_{(t-1)} + (1 - \rho_1) * g_t$ 9: Hitung besarnya variansi gradien $u_t \leftarrow \rho_2 * u_{(t-1)} + (1 - \rho_2) * (g_t \odot g_t)$ 10: Hitung momen pertama terkoreksi $\hat{m}_t \leftarrow m_t / (1 - \rho_1^t)$ 11: Hitung bias pertama terkoreksi $\hat{u}_t \leftarrow u_t / (1 - \rho_2^t)$ 12: Hitung nilai perubahan bobot $\Delta\theta \leftarrow -\epsilon * (\hat{m}_t / (\sqrt{\hat{u}_t} + \delta))$ 13: Hitung pembaruan nilai bobot pada indeks $\theta_t \leftarrow \theta_{(t-1)} + \Delta\theta$ 14: EndFor 15: Evaluasi performa (Dice Coefficient) 16: EndFor 17: Return W, b 18: EndAlgorithm

Gambar 3. 13 Algoritma 5 *Backward Pass* U-Net

Pada Algoritma 5 *Backward_Pass* U-Net, model belajar dari kesalahan prediksinya. Proses ini dimulai dengan membagi data menjadi *batch* kecil, kemudian setiap *batch* diproses melalui *forward pass* untuk menghasilkan prediksi. Setelah itu, model menghitung besar kesalahan (*loss*) menggunakan Persamaan 3.7, yaitu fungsi *Binary Cross-Entropy* yang membandingkan prediksi model dengan label asli. Nilai *loss* ini menunjukkan seberapa jauh prediksi model dari kebenaran—makin besar nilainya, makin besar kesalahan model. Dari sini dihitung gradien, yaitu informasi tentang bagaimana parameter model (bobot dan bias) harus diperbaiki agar kesalahan berkurang.

Gradien ini kemudian diolah menggunakan mekanisme *Optimizer Adam*, yang dijelaskan dalam Persamaan 3.8 hingga Persamaan 3.13. Pada tahap ini, Adam

menghitung dua nilai penting: m_t (momentum pertama) dan u_t (momentum kedua). Secara sederhana, m_t mewakili *arah rata-rata* perbaikan bobot, sedangkan u_t mewakili “seberapa kuat atau tidak stabil” pergerakan gradien tersebut. Namun, kedua nilai ini masih memiliki bias karena dihitung dari awal pelatihan. Sehingga, dilakukan koreksi menggunakan Persamaan 3.10 dan Persamaan 3.11 untuk menghasilkan \hat{m}_t dan \hat{u}_t , yaitu bentuk m_t dan u_t yang sudah dikoreksi sehingga lebih stabil dan akurat. Nilai-nilai inilah yang digunakan pada Persamaan 3.12 dan Persamaan 3.13 untuk menentukan perubahan bobot ($\Delta\theta$), yaitu seberapa besar parameter model perlu bergerak menuju solusi yang lebih baik. Setiap parameter lalu diperbarui berdasarkan $\Delta\theta$, dan proses ini diulang pada setiap *batch* dan setiap *epoch*.

Setelah seluruh *batch* selesai diproses pada satu *epoch*, performa model dievaluasi menggunakan Dice Coefficient untuk melihat apakah perubahan parameter membuat hasil prediksi semakin akurat. Proses ini terus berulang hingga model benar-benar stabil dan mampu menandai area tumor pada citra dengan baik.

3. Parameter dan Strategi Pelatihan

Pengujian ini bertujuan untuk mengetahui sejauh mana model mampu mendeteksi area tumor dengan tingkat akurasi yang tinggi berdasarkan variasi parameter dan konfigurasi sistem. Evaluasi dilakukan pada beberapa aspek, seperti pengaruh *learning rate*, *batch size*, dan jumlah *epoch* terhadap hasil segmentasi. Selain itu, dilakukan pula pengujian terhadap rasio pembagian dataset untuk memastikan keseimbangan antara data pelatihan, validasi, dan

pengujian. Proses pelatihan menggunakan fungsi aktivasi ReLU pada lapisan konvolusi untuk mempercepat pembelajaran, serta Sigmoid pada lapisan *output* untuk menghasilkan nilai probabilitas biner antara area tumor dan no tumor. Setiap pengujian dilakukan dengan pendekatan sistematis untuk menilai akurasi, konsistensi hasil, dan kemampuan model dalam mengenali pola citra secara menyeluruh.

3.4 Skenario Uji coba

Tahap skenario uji coba dirancang untuk menguji performa model usulan *Lightweight* U-Net. Skenario ini diintegrasikan pada saat pengujian sistem dengan memperhatikan berbagai aspek konfigurasi dan parameter guna mengevaluasi efektivitas serta keandalan model. Adapun skenario uji coba yang dilakukan meliputi pengelompokan dataset dan pengujian sistem.

3.4.1 Pengelompokan Dataset

Pengelompokan Penelitian ini menggunakan dataset tumor otak yang telah dikelompokkan menjadi tiga jenis, yaitu *Glioma*, *Meningioma*, dan *Pituitary Tumor*. Pengelompokan data dilakukan karena pada data asli hanya terdapat dua folder utama, yaitu folder image dan mask, tanpa adanya pembagian berdasarkan jenis tumor. Oleh karena itu, pada tahap pra-pemrosesan dilakukan pengelompokan ulang berdasarkan kategori kondisi tumor tersebut yang ditunjukkan oleh tabel berikut.

Tabel 3. 2 Proporsi *Split* Data

No.	Proses	Proporsi (%)	Jumlah Data
1.	<i>Training</i>	70	2143
2.	<i>Validation</i>	15	414

No.	Proses	Proporsi (%)	Jumlah Data
3.	<i>Testing</i>	15	307

Pada Tabel 3.2 secara keseluruhan, dataset terdiri atas 3.064 citra beserta mask-nya, yang kemudian dibagi menjadi tiga bagian untuk proses *Training* (70%), *Validation* (15%), dan *Testing* (15%). Proporsi pembagian data ini dipilih karena umum digunakan dalam penelitian-penelitian terdahulu, salah satunya oleh Pourmahboubi et al. (2025) yang menerapkan strategi split dataset dengan *Training* 70%, *Validasi* 15%, dan *Testing* 15% dalam pengembangan model segmentasi tumor otak berbasis U-Net dan *transfer learning*, sehingga dianggap efektif untuk menjaga keseimbangan antara performa pelatihan model dan akurasi evaluasi (Pourmahboubi et al., 2025).

Tabel 3. 3 Label Area Tumor

No.	Area	Label
1.	No Tumor	0
2.	Tumor	1

Selanjutnya pada Tabel 3. 3 merupakan tabel label area setelah proses *training* dan *testing* dilakukan menggunakan fungsi *loss Binary Cross-Entropy* (BCE), yang hanya dapat digunakan untuk data dengan label biner, yaitu 0 dan 1, yang masing-masing merepresentasikan area tumor dan area no tumor pada bagian segmentasi.

3.4.2 Pengujian Parameter Sistem

Untuk mengevaluasi kinerja sistem dalam melakukan segmentasi tumor otak berbasis citra MRI, dilakukan beberapa skenario pengujian terhadap model

usulan *Lightweight* U-Net. Rincian lengkap mengenai parameter dan konfigurasi yang diuji dapat dilihat pada tabel berikut.

Tabel 3. 4 Parameter yang Diuji

No.	Model	Batch Size	Learning Rate	Epoch
1.	<i>Lightweight</i> U-Net	16	0.001	100
2.	<i>Lightweight</i> U-Net	16	0.001	150
3.	<i>Lightweight</i> U-Net	16	0.001	200
4.	<i>Lightweight</i> U-Net	16	0.0001	100
5.	<i>Lightweight</i> U-Net	16	0.0001	150
6.	<i>Lightweight</i> U-Net	16	0.0001	200
7.	<i>Lightweight</i> U-Net	16	0.00001	100
8.	<i>Lightweight</i> U-Net	16	0.00001	150
9.	<i>Lightweight</i> U-Net	16	0.00001	200

Berdasarkan Tabel 3. 4 proses pengujian dirancang menggunakan konfigurasi *batch size* 16, *learning rate* (LR) 0.001; 0.0001; 0.00001 menggunakan eksperimen jumlah *epoch* sebesar 100, 150, dan 200 (Ardan & Indraswari, 2024; Haq, 2022; Lin & Lin, 2024; Matuszewski & Sintorn, 2019; Santoso, 2024). Dari *hyperparameter* tersebut telah mengalami penelitian terdahulu, misalnya pada pemilihan nilai *batch size* dan *learning rate* dengan usulan nilai *epoch*. Selain itu, bobot awal jaringan diinisialisasi menggunakan *He Normal Initialization* yang sesuai dengan fungsi aktivasi ReLU, guna menjaga kestabilan distribusi nilai dan mempercepat proses konvergensi selama pelatihan. Kinerja model juga dipantau

melalui data validasi agar model tidak mengalami *overfitting* serta mampu mempertahankan keseimbangan antara akurasi dan generalisasi.

Dengan adanya Tabel 3. 4 yang berisikan *hyperparameter* pengujian tersebut, disusun enam skenario pengujian untuk melatih model *Lightweight* U-Net. Seluruh skenario menggunakan *batch size* 16, sementara variasi dilakukan pada nilai *learning rate* 0.001; 0.0001; 0.00001 serta jumlah *epoch* 100, 150, dan 200.

Keenam skenario ini diperlukan untuk memperoleh hasil pelatihan yang optimal sehingga performa *Dice Coefficient* dan akurasi model dapat dievaluasi secara menyeluruh sesuai tujuan penelitian. Rincian lengkap terkait skenario pengujian dapat dilihat pada Tabel 3. 5.

Tabel 3. 5 Skenario Pengujian

No.	Skenario	Batch Size	Learning Rate	Epoch		
1.	Skenario 1	16	0.001	100		
				150		
				200		
2.	Skenario 2		16	0.0001	100	
					150	
					200	
3.	Skenario 3			16	0.00001	100
						150
						200

3.5 Evaluasi Kinerja Sistem

Langkah terakhir dalam prosedur penelitian ini adalah tahap evaluasi kinerja sistem. Evaluasi ini dilakukan untuk menilai sejauh mana model usulan *Lightweight* U-Net mampu melakukan segmentasi area tumor otak secara tepat berdasarkan hasil prediksi citra. Pengukuran kinerja dilakukan dengan menggunakan beberapa metrik evaluasi, yaitu *Dice Coefficient* dan Akurasi. Kedua metrik tersebut digunakan untuk menilai tingkat kesesuaian antara hasil segmentasi model dengan *ground truth* yang telah disediakan. Metode evaluasi ini bertujuan untuk

memastikan bahwa model tidak hanya mampu mengenali area tumor, tetapi juga memiliki kemampuan prediksi batas objek yang presisi. Data hasil pengujian yang diperoleh dari proses *Testing* kemudian dianalisis untuk mengukur kemampuan model dalam menghasilkan segmentasi yang konsisten dan akurat pada berbagai variasi citra MRI.

3.5.1 *Dice Coefficient*

Dice Coefficient merupakan metrik evaluasi yang umum digunakan dalam segmentasi citra medis untuk mengukur tingkat kesesuaian antara hasil segmentasi model dan *ground truth*. Metrik ini dihitung berdasarkan perbandingan area irisan terhadap total area gabungan antara hasil prediksi dan data acuan, sebagaimana dijelaskan pada penelitian sebelumnya (Suta et al., 2020). Rumus matematis *Dice Coefficient* ditunjukkan pada Persamaan 3.14:

$$Dice = \frac{2 \times |A \cap B|}{|A| + |B|} \quad (3.14)$$

Keterangan:

$|A \cap B|$ = area irisan (*Intersection*) antara hasil prediksi dan *ground truth*

$|A|$ = jumlah elemen dalam prediksi

$|B|$ = jumlah elemen dalam *ground truth*

3.5.2 Akurasi

Akurasi merupakan salah satu metrik evaluasi yang umum digunakan dalam pembelajaran mesin dan pembelajaran mendalam. Metrik ini mengukur kinerja keseluruhan model dengan menghitung perbandingan antara jumlah prediksi yang benar dan total jumlah sampel dalam dataset. Rumus akurasi dinyatakan pada Persamaan 3.15:

$$accuracy = \frac{\text{Number of True Positif (TP)} + \text{Number of True Negative (TN)}}{\text{Total Number of Sample (TP+TN+FP+FN)}} \quad (3.15)$$

Berdasarkan Tabel 3.3, True Positive (TP) mengacu pada area tumor otak yang benar-benar terdeteksi sebagai area tumor oleh model. True Negative (TN) adalah area otak normal yang benar-benar terdeteksi sebagai no tumor. Di sisi lain, False Positive (FP) menunjukkan area otak normal yang salah terdeteksi sebagai tumor, sedangkan False Negative (FN) merujuk pada area tumor yang salah terdeteksi sebagai no tumor (Alali & Hussein Ali, 2022). Keempat metrik ini digunakan untuk mengevaluasi kinerja model usulan *Lightweight U-Net Segmentation* dalam mengklasifikasikan setiap piksel citra MRI dengan benar atau salah, sehingga dapat diketahui sejauh mana model mampu mendeteksi area tumor secara akurat.

3.5.3 Sensitifitas (*Recall*)

Recall, atau yang juga dikenal dengan istilah *Sensitivity*, merupakan metrik evaluasi yang digunakan untuk mengukur seberapa banyak *true positives* yang diprediksi dari seluruh sampel positif aktual yang ada di dalam dataset. Metrik ini menghitung rasio prediksi *True Positive* (TP) terhadap total keseluruhan data yang secara aktual bernilai positif (Alali & Hussein Ali, 2022). Perhitungan *Recall* dinyatakan dalam Persamaan 3.16 sebagai berikut:

$$Recall = \frac{TP}{TP+FN} \quad (3.16)$$

Dalam penelitian segmentasi tumor otak ini, *Recall* menjadi parameter krusial untuk mengetahui sejauh mana model *Lightweight U-Net Segmentation* mampu mendeteksi seluruh area tumor tanpa melewatkan piksel penting. Nilai *Recall* yang tinggi

mengindikasikan bahwa model memiliki tingkat False Negative (FN) yang rendah, yang berarti sangat sedikit area tumor yang salah terklasifikasi sebagai jaringan normal (*no tumor*).

3.5.4 Precision

Precision merupakan metrik evaluasi yang digunakan untuk mengukur tingkat keakuratan dari sebuah prediksi positif yang dihasilkan oleh model. Dengan kata lain, metrik ini memberikan gambaran mengenai seberapa besar tingkat kepastian bahwa suatu data yang diprediksi sebagai positif memang benar-benar bernilai positif secara actual (Alali & Hussein Ali, 2022). Perhitungan *Precision* dinyatakan dalam Persamaan 3.17 sebagai berikut:

$$Precision = \frac{TP}{TP+FP} \quad (3.17)$$

Dalam konteks segmentasi tumor otak, nilai *Precision* yang tinggi menunjukkan kemampuan model *Lightweight U-Net Segmentation* dalam meminimalkan kesalahan deteksi pada jaringan otak normal. Hal ini sangat penting untuk memastikan bahwa area yang diidentifikasi sebagai tumor oleh model benar-benar merupakan jaringan tumor, sehingga dapat mengurangi risiko kesalahan diagnosis yang bersifat *False Positive* (FP).

BAB IV

HASIL DAN PEMBAHASAN

4.1 Hasil *Training* Setiap Skenario

Pada subbab ini akan dijelaskan tahapan-tahapan yang dilakukan selama proses *training* model segmentasi tumor otak, mulai dari tahap pra-pemrosesan data hingga *training* model *Lightweight* U-Net. Setiap tahapan dalam proses ini memiliki peran penting untuk memastikan model mampu melakukan segmentasi area tumor secara akurat pada citra MRI otak. Proses dimulai dengan persiapan dan pemrosesan data guna menghasilkan kualitas data yang optimal sebagai masukan bagi model. Tahap pra-pemrosesan meliputi penyesuaian ukuran citra (*resize*) serta pembagian dataset ke dalam data *training*, data *validation*, dan data *testing*. Langkah ini dilakukan untuk memastikan keseragaman data serta mendukung proses *training* dan evaluasi model secara sistematis.

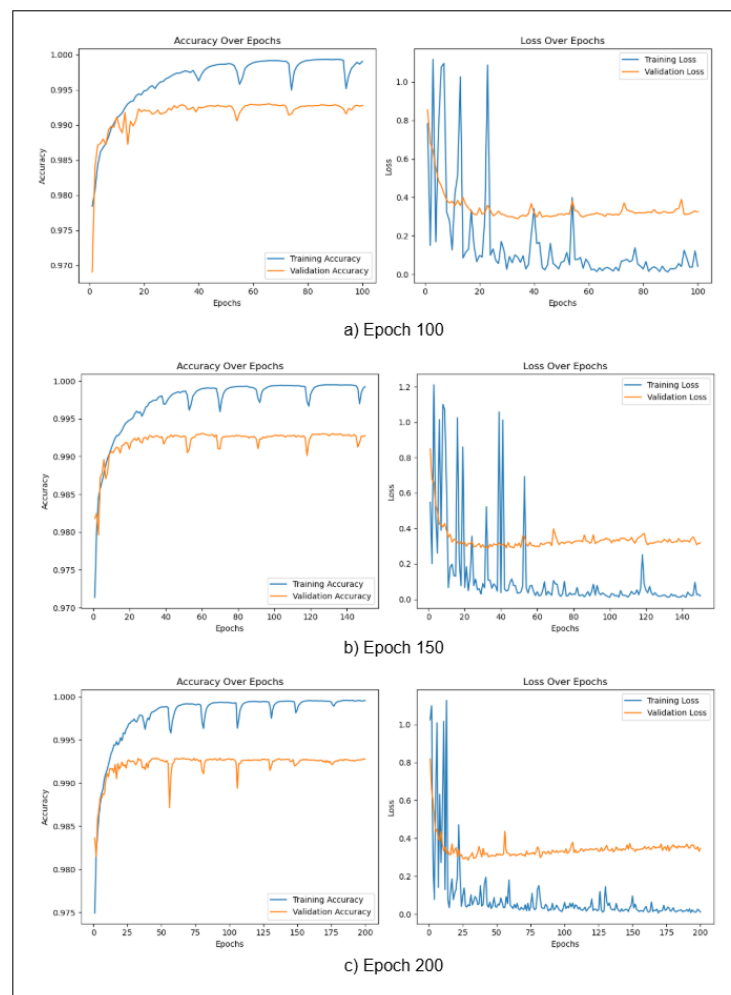
Tahap *training* model *Lightweight* U-Net. Model ini berperan sebagai arsitektur utama dalam mengekstraksi fitur spasial dan melakukan pemetaan area tumor pada citra MRI. Melalui mekanisme *encoder-decoder* dan *skip connection*, *Lightweight* U-Net memanfaatkan fitur-fitur penting yang diperoleh pada tahap ekstraksi untuk menghasilkan peta segmentasi yang merepresentasikan area tumor secara detail. Proses *training* model dilakukan menggunakan data *training* dan divalidasi menggunakan data *validation* untuk memantau performa model selama proses pembelajaran.

Tahapan-tahapan dalam proses *training* model ini disusun secara sistematis untuk menjaga keterkaitan yang jelas antara tahap persiapan data dan proses pembelajaran model. Dengan alur penelitian yang terstruktur, diharapkan model *Lightweight* U-Net mampu mempelajari representasi fitur yang relevan dari citra MRI otak dan menghasilkan segmentasi tumor yang akurat. Penjelasan lebih rinci mengenai setiap tahapan akan dibahas pada subbab berikutnya.

4.1.1 Hasil *Training* Skenario 1

Pada konfigurasi *learning rate* 0,001, hasil pelatihan model *Lightweight* U-Net pada Gambar 4.1 menunjukkan proses pembelajaran yang stabil dan konvergen. Pada Gambar 4.1(a) *Epoch* 100, *training accuracy* meningkat pesat di awal epoch dan mendekati nilai maksimum, sementara *validation accuracy* meningkat signifikan sebelum mencapai kondisi stabil. *Training loss* menurun tajam di awal pelatihan, diikuti penurunan *validation loss* secara bertahap hingga relatif stabil. Pola ini menunjukkan bahwa model mampu mempelajari fitur segmentasi tumor dengan baik dan memiliki kemampuan generalisasi yang memadai.

Pada Gambar 4.1(b) *Epoch* 150, *training accuracy* tetap tinggi dan stabil, namun *validation accuracy* tidak menunjukkan peningkatan signifikan dibandingkan *Epoch* 100. *Training loss* semakin kecil dan stabil, sedangkan *validation loss* cenderung stagnan dengan fluktuasi kecil. Kondisi ini mengindikasikan bahwa model telah mencapai batas optimal pembelajaran, sehingga penambahan epoch tidak memberikan peningkatan performa yang berarti pada data validasi.

Gambar 4. 1 Hasil *Training* Skenario 1

Pada Gambar 4.1(c) Epoch 200, *training accuracy* tetap berada pada nilai maksimum, namun *validation accuracy* tidak menunjukkan peningkatan signifikan dibandingkan konfigurasi sebelumnya. *Training loss* semakin rendah dan stabil, sementara *validation loss* relatif konstan dengan sedikit fluktuasi. Perbedaan yang konsisten antara kurva training dan validation pada Epoch 200 mengindikasikan kecenderungan *overfitting*. Berdasarkan analisis ketiga konfigurasi tersebut, model *Lightweight* U-Net mencapai performa optimal pada rentang 100–150 epoch,

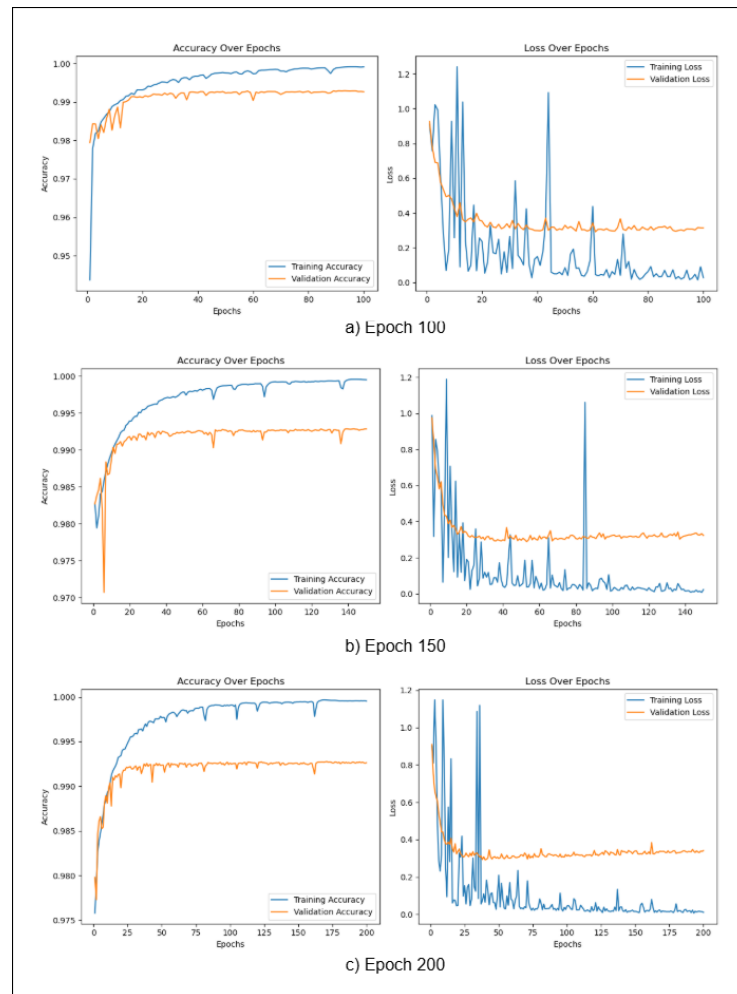
sehingga penambahan epoch hingga 200 dinilai kurang efisien dari segi waktu pelatihan dan sumber daya komputasi.

4.1.2 Hasil *Training* Skenario 2

Pada konfigurasi *learning rate* 0,0001, hasil pelatihan model *Lightweight* U-Net yang ditunjukkan pada Gambar 4.2 memperlihatkan proses pembelajaran yang berlangsung secara bertahap dan stabil. Pada Gambar 4.2(a) *Epoch* 100, *training accuracy* meningkat dengan cepat pada tahap awal pelatihan dan mencapai nilai yang tinggi, sementara *validation accuracy* juga mengalami peningkatan meskipun masih menunjukkan fluktuasi pada beberapa *epoch* awal. Grafik *loss* pada Gambar 4.2(a) menunjukkan penurunan *training loss* yang cukup tajam, sedangkan *validation loss* menurun secara bertahap hingga mencapai nilai yang relatif stabil. Pola ini menunjukkan bahwa model telah mampu mempelajari pola segmentasi tumor dengan baik, meskipun proses pembelajaran masih berada pada tahap penyesuaian awal.

Pada Gambar 4.2(b) *Epoch* 150, *training accuracy* terlihat semakin stabil dan berada mendekati nilai maksimum, sedangkan *validation accuracy* menunjukkan konsistensi yang lebih baik dengan fluktuasi yang lebih kecil dibandingkan *epoch* 100. Kurva *training loss* menjadi lebih rendah dan stabil, sementara *validation loss* berada pada rentang nilai yang relatif konstan. Pada Gambar 4.2(c) *Epoch* 200, baik *training accuracy* maupun *validation accuracy* tidak menunjukkan peningkatan yang signifikan dibandingkan hasil pada *epoch* 150. *Training loss* tetap rendah, sedangkan *validation loss* cenderung stagnan dengan perubahan yang sangat kecil. Hal ini mengindikasikan bahwa model telah

mencapai kondisi konvergen pada konfigurasi *learning rate* 0,0001, sehingga pelatihan hingga 150 *epoch* dapat dianggap cukup optimal, sementara penambahan *epoch* hingga 200 hanya memberikan peningkatan performa yang minimal.



Gambar 4. 2 Hasil *Training* Skenario 2

4.1.3 Hasil *Training* Skenario 3

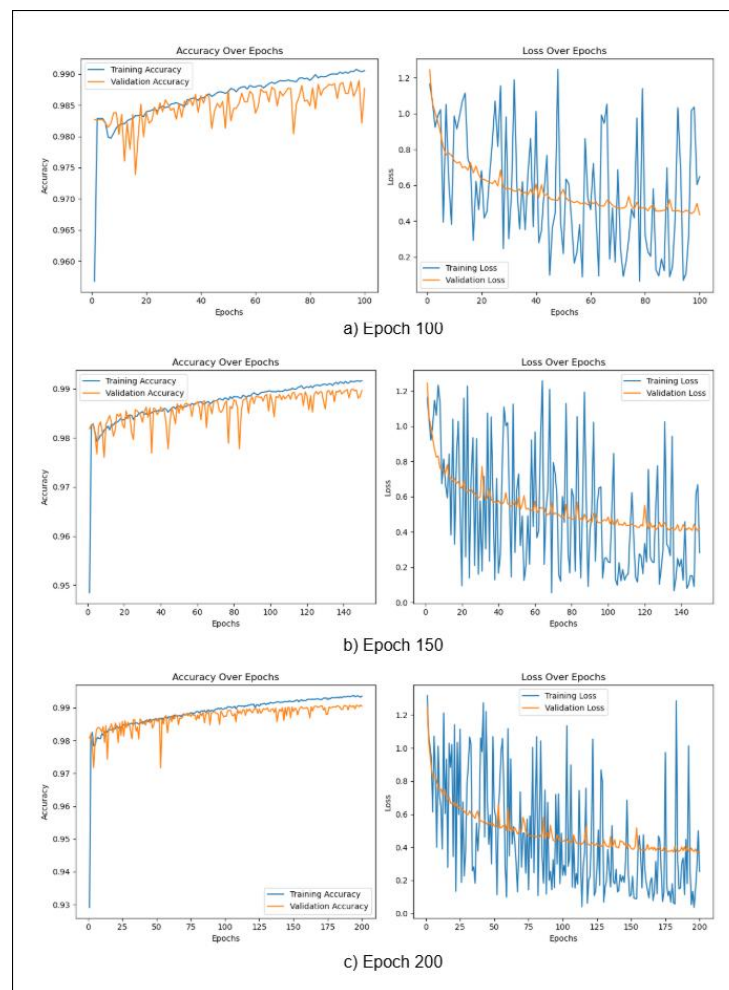
Pada konfigurasi *learning rate* 0,00001, hasil pelatihan model *Lightweight* U-Net pada Gambar 4.3 menunjukkan proses pembelajaran yang berlangsung lebih lambat dibandingkan skenario sebelumnya. Pada Gambar 4.3(a) *Epoch* 100, kurva *training accuracy* meningkat secara bertahap dari *epoch* awal hingga mencapai

nilai yang cukup tinggi, sedangkan *validation accuracy* juga mengalami peningkatan namun masih terlihat fluktuasi pada beberapa titik. Grafik *loss* pada Gambar 4.3(a) menunjukkan penurunan *training loss* yang relatif lebih lambat dan disertai fluktuasi yang cukup besar, sementara *validation loss* menurun secara perlahan menuju nilai yang lebih stabil. Pola ini menunjukkan bahwa model masih berada pada tahap penyesuaian parameter dan proses pembelajaran belum sepenuhnya mencapai kondisi optimal.

Pada Gambar 4.3(b) *Epoch 150*, *training accuracy* mengalami peningkatan dan mulai menunjukkan kestabilan, sedangkan *validation accuracy* menjadi lebih konsisten meskipun masih terdapat fluktuasi kecil. Kurva *training loss* terlihat menurun dibandingkan *epoch* sebelumnya, namun masih menunjukkan variasi nilai yang cukup tinggi. *Validation loss* cenderung menurun secara bertahap dan berada pada rentang nilai yang relatif stabil. Hal ini mengindikasikan bahwa penambahan jumlah *epoch* membantu model dalam memperbaiki proses pembelajaran, meskipun konvergensi belum sepenuhnya tercapai.

Gambar 4.3(c) *Epoch 200*, *training accuracy* mencapai nilai yang lebih tinggi dan cenderung stabil, sementara *validation accuracy* juga menunjukkan peningkatan meskipun tidak terlalu signifikan dibandingkan *epoch 150*. Kurva *training loss* masih menunjukkan fluktuasi yang cukup besar, sedangkan *validation loss* menurun secara perlahan dan berada pada nilai yang relatif konstan. Kondisi ini menunjukkan bahwa penggunaan *learning rate* 0,00001 menyebabkan proses pembelajaran berlangsung lebih lambat, sehingga model memerlukan jumlah *epoch* yang lebih banyak untuk mencapai kondisi konvergen. Berdasarkan hasil tersebut,

dapat disimpulkan bahwa performa model pada skenario ini belum menunjukkan peningkatan yang optimal dibandingkan skenario sebelumnya, sehingga learning rate yang terlalu kecil dapat menghambat kecepatan konvergensi model.



Gambar 4. 3 Hasil Training Skenario 3

4.2 Hasil *Testing* Setiap Skenario

Setelah proses pelatihan model selesai, tahap selanjutnya adalah pengujian untuk mengevaluasi kinerja model *Lightweight* U-Net dalam melakukan segmentasi tumor otak. Pengujian ini bertujuan untuk menilai kemampuan model dalam menghasilkan prediksi yang akurat dan konsisten terhadap data uji, serta

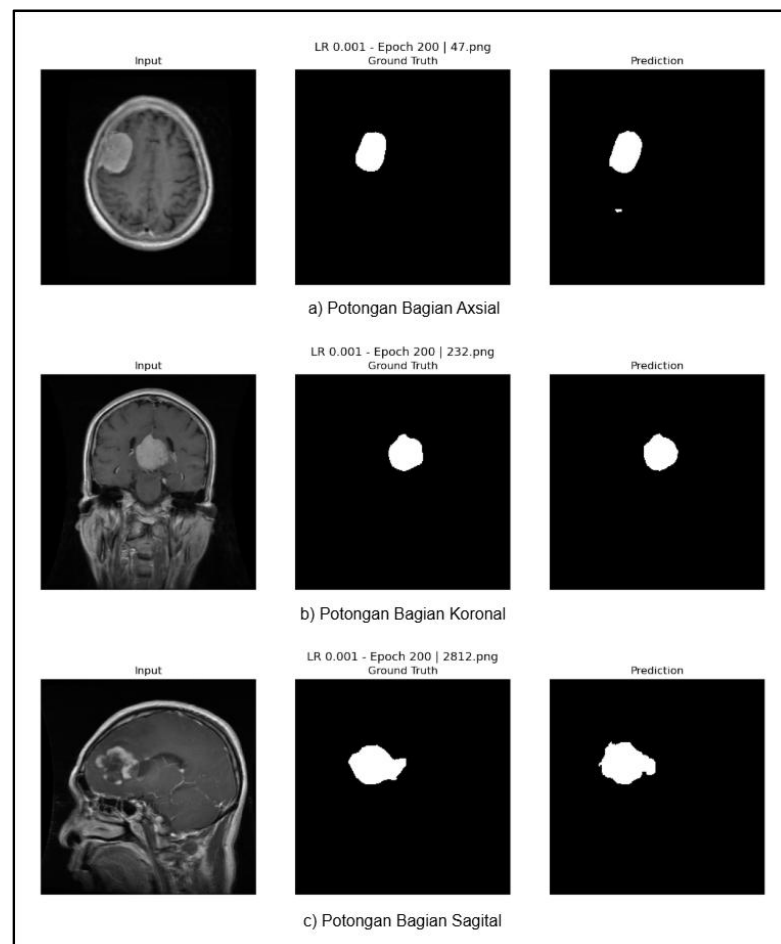
mengamati pengaruh pengaturan parameter pelatihan terhadap kualitas segmentasi yang dihasilkan. Pada penelitian ini, pengujian dilakukan menggunakan beberapa konfigurasi *learning rate* yang sama dengan skenario pelatihan, sehingga hasil yang diperoleh dapat digunakan untuk menilai hubungan antara parameter pelatihan dan performa model pada data yang belum pernah dilihat sebelumnya.

Pembahasan hasil *testing* pada bagian ini difokuskan pada analisis kualitatif dan kuantitatif kinerja model dalam melakukan segmentasi tumor otak. Analisis kualitatif ditunjukkan melalui visualisasi hasil prediksi model dibandingkan dengan *ground truth*, sedangkan analisis kuantitatif dilakukan menggunakan metrik evaluasi berupa *confusion matrix* dan nilai *Pixel Accuracy*. Selain itu, hasil pengujian juga disajikan dalam bentuk diagram batang untuk menampilkan perbandingan performa *testing* dari ketiga model berdasarkan variasi *learning rate* yang digunakan, sehingga memudahkan identifikasi model dengan performa terbaik. Untuk melengkapi analisis tersebut, digunakan pula visualisasi *boxplot* guna menggambarkan sebaran nilai *Dice Coefficient*, yang bertujuan untuk mengevaluasi konsistensi dan kestabilan kinerja model pada seluruh data uji. Penyajian hasil *testing* dilakukan secara terpisah untuk setiap konfigurasi *learning rate* agar memberikan gambaran yang komprehensif mengenai pengaruh parameter pelatihan terhadap kualitas segmentasi yang dihasilkan.

4.2.1 Hasil *Testing* Skenario 1

Pada skenario pertama, pengujian model dilakukan menggunakan *learning rate* sebesar 0,001. Hasil pengujian ditampilkan pada Gambar 4.4 yang terdiri dari tiga bagian, yaitu (a) potongan aksial, (b) potongan koronal, dan (c) potongan

sagital. Setiap bagian menampilkan citra input, *ground truth*, serta hasil prediksi segmentasi. Pada Gambar 4.4(a), model mampu mengidentifikasi area tumor dengan baik, dimana bentuk dan lokasi prediksi sudah mendekati *ground truth*. Hal serupa juga terlihat pada Gambar 4.4(b) dan Gambar 4.4(c), dimana area tumor berhasil tersegmentasi dengan cukup presisi dan hanya terdapat sedikit perbedaan pada batas tepi. Secara keseluruhan, hasil visual menunjukkan bahwa model mampu mengenali pola spasial tumor secara konsisten pada berbagai sudut pandang citra MRI.



Gambar 4. 4 Hasil Segmentasi Skenario 1

Analisis kuantitatif dilakukan menggunakan *confusion matrix* yang disajikan untuk masing-masing citra uji. Pada data (a) dalam Tabel 4. 1, jumlah *Actual Pixel* Tumor (total area tumor sebenarnya) adalah sebesar 1327 piksel (1315 *true positive* + 12 *false negative*), sedangkan *Predicted Pixel* Tumor adalah sebesar 1446 piksel (1315 *true positive* + 131 *false positive*). Pada data (b) dalam Tabel 4. 2, jumlah *Actual Pixel* Tumor sebesar 1342 piksel (1287 + 55), dan *Predicted Pixel* Tumor sebesar 1310 piksel (1287 + 23). Pada data (c) dalam Tabel 4. 3, jumlah *Actual Pixel* Tumor sebesar 2158 piksel (1942 + 216), dan *Predicted Pixel* Tumor sebesar 2276 piksel (1942 + 334). Perbandingan ini menunjukkan bahwa sebagian besar area tumor berhasil terdeteksi dengan baik oleh model, meskipun masih terdapat sedikit kelebihan prediksi (*over-segmentation*) maupun kekurangan prediksi (*under-segmentation*) pada beberapa citra.

Tabel 4. 1 *Confusion Matrix* Skenario 1 Gambar 47.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	64078	131
	Tumor	12	1315

Tabel 4. 2 *Confusion Matrix* Skenario 1 Gambar 232.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	64171	23
	Tumor	55	1287

Tabel 4. 3 *Confusion Matrix* Skenario 1 Gambar 2812.png

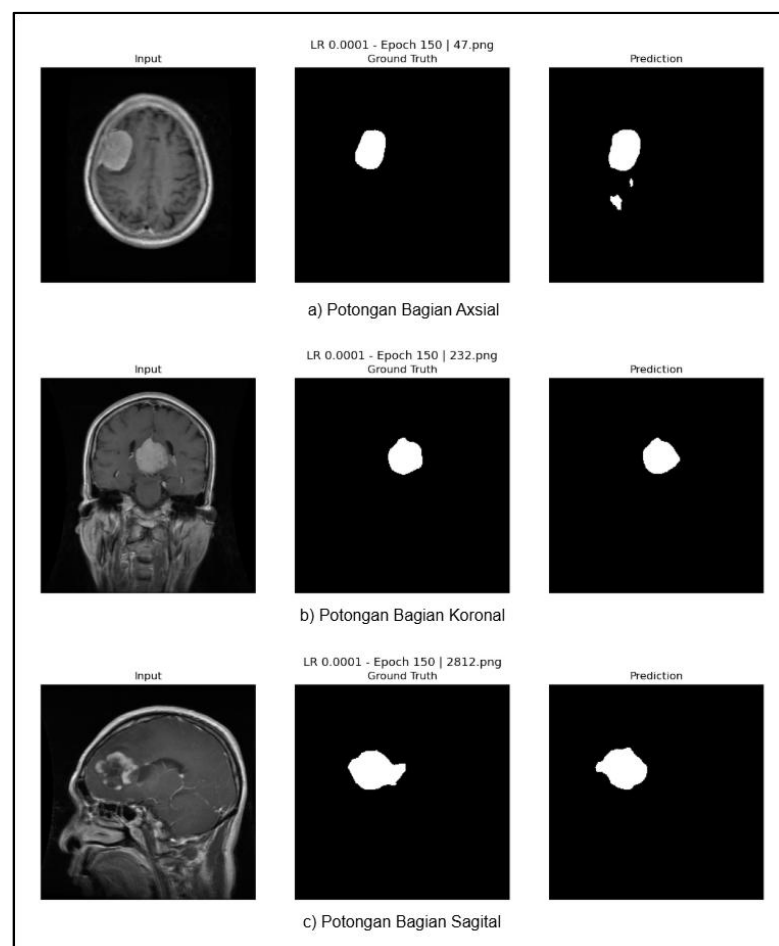
		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	63044	334
	Tumor	216	1942

Secara keseluruhan, nilai *true positive* yang tinggi pada ketiga pengujian menunjukkan bahwa model memiliki kemampuan yang baik dalam mendeteksi area tumor. Nilai *false negative* yang relatif kecil mengindikasikan bahwa hanya sebagian kecil area tumor yang tidak terdeteksi, sedangkan nilai *false positive* yang rendah menunjukkan bahwa model jarang mengklasifikasikan area non-tumor sebagai tumor. Dengan demikian, baik dari analisis kualitatif maupun kuantitatif, model *Lightweight* U-Net pada skenario pertama menunjukkan performa segmentasi yang akurat dan stabil, sehingga mampu merepresentasikan area tumor dengan cukup baik pada data pengujian.

4.2.2 Hasil *Testing* Skenario 2

Pada skenario kedua, pengujian model dilakukan menggunakan *learning rate* sebesar 0,0001. Hasil pengujian ditampilkan pada Gambar 4.5 yang terdiri dari tiga bagian, yaitu (a) potongan aksial, (b) potongan koronal, dan (c) potongan sagital. Pada Gambar 4.5(a), model mampu mengidentifikasi area tumor dengan cukup baik, dimana lokasi dan bentuk segmentasi sudah mendekati *ground truth*, meskipun masih terdapat sedikit area tambahan di luar tumor (*over-segmentation*).

Pada Gambar 4.5(b), hasil prediksi menunjukkan kesesuaian yang lebih baik dengan *ground truth*, dengan batas tumor yang relatif rapi dan minim kesalahan. Sementara itu, pada Gambar 4.5(c), model tetap mampu mengenali area tumor utama, namun terdapat sedikit perbedaan pada bagian tepi sehingga bentuk segmentasi tidak sepenuhnya identik. Secara keseluruhan, hasil visual menunjukkan bahwa model memiliki kemampuan segmentasi yang baik dan cukup konsisten pada berbagai sudut pandang citra MRI.



Gambar 4. 5 Hasil Segmentasi Skenario

Analisis kuantitatif dilakukan berdasarkan *confusion matrix* untuk masing-masing citra uji. Pada data (a) dalam Tabel 4. 4, jumlah *Actual Pixel* Tumor adalah

sebesar 1327 piksel (1324 *true positive* + 3 *false negative*), sedangkan *Predicted Pixel* Tumor sebesar 1672 piksel (1324 *true positive* + 348 *false positive*). Pada data (b) dalam Tabel 4. 5, jumlah *Actual Pixel* Tumor sebesar 1342 piksel (1286 + 56), dan *Predicted Pixel* Tumor sebesar 1333 piksel (1286 + 47). Pada data (c) dalam Tabel 4. 6, jumlah *Actual Pixel* Tumor sebesar 2158 piksel (1760 + 398), dan *Predicted Pixel* Tumor sebesar 2017 piksel (1760 + 257). Hasil ini menunjukkan bahwa model mampu mendeteksi sebagian besar area tumor dengan baik, meskipun pada beberapa kasus masih terjadi kelebihan prediksi maupun kekurangan segmentasi.

Tabel 4. 4 *Confusion Matrix* Skenario 2 Gambar 47.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	63861	348
	Tumor	3	1324

Tabel 4. 5 *Confusion Matrix* Skenario 2 Gambar 232.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	64147	47
	Tumor	56	1286

Tabel 4. 6 Confusion Matrix Skenario 2 Gambar 2812.png

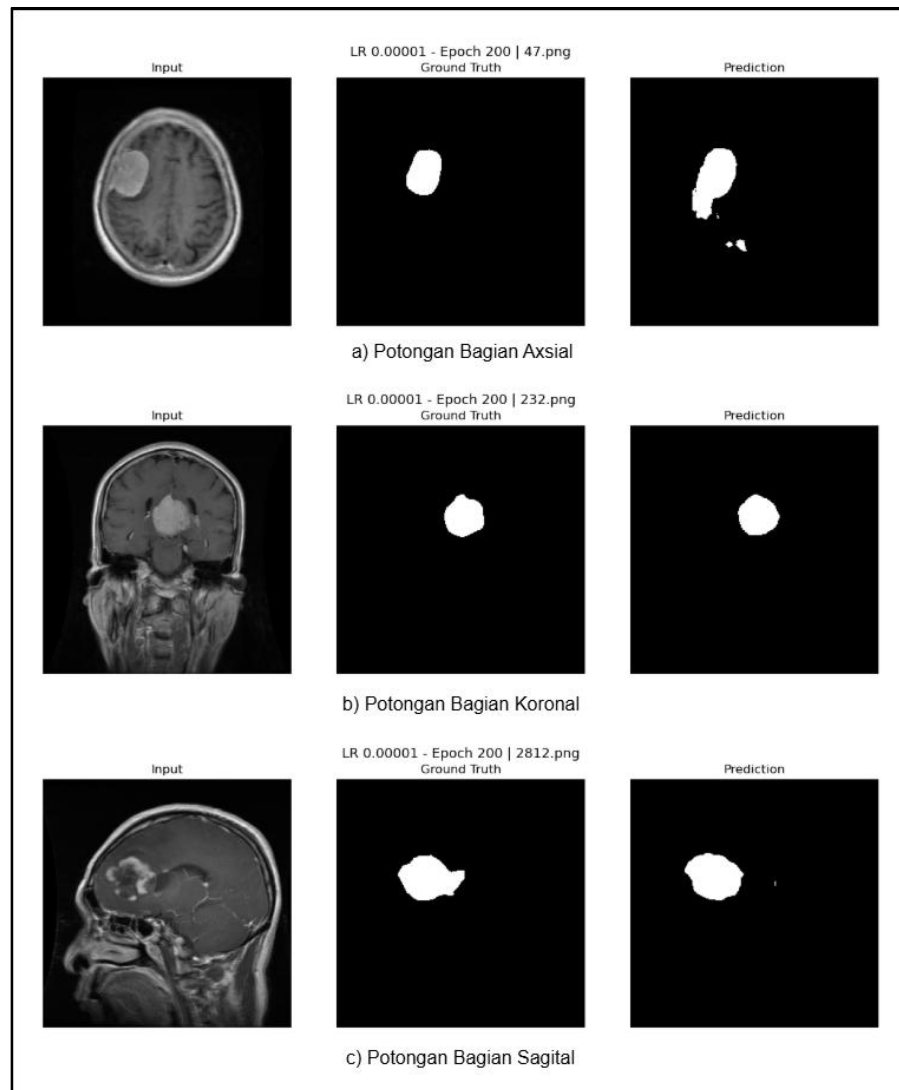
		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	63121	257
	Tumor	398	1760

Secara keseluruhan, nilai *true positive* yang tinggi pada ketiga citra menunjukkan bahwa model memiliki kemampuan deteksi tumor yang baik. Nilai *false negative* yang relatif kecil mengindikasikan bahwa sebagian besar area tumor berhasil dikenali, sedangkan nilai *false positive* yang masih muncul pada beberapa citra menunjukkan adanya kecenderungan model dalam mendeteksi area non-tumor sebagai tumor. Dengan demikian, berdasarkan analisis kualitatif dan kuantitatif, performa model pada skenario kedua tergolong baik dan stabil, serta menunjukkan kemampuan generalisasi yang cukup tinggi pada data pengujian.

4.2.3 Hasil *Testing* Skenario 3

Pada skenario ketiga, pengujian model dilakukan menggunakan *learning rate* sebesar 0,00001. Hasil pengujian ditampilkan pada Gambar 4.6 yang terdiri dari (a) potongan aksial, (b) potongan koronal, dan (c) potongan sagital. Pada Gambar 4.6(a), model mampu mendeteksi area tumor utama, namun terlihat adanya prediksi tambahan di luar area tumor yang menunjukkan *over-segmentation*. Pada Gambar 4.6(b), hasil segmentasi terlihat cukup baik dengan bentuk dan lokasi yang mendekati *ground truth*, meskipun masih terdapat sedikit perbedaan pada batas tepi. Sementara itu, pada Gambar 4.6(c), model tetap dapat mengenali area tumor utama,

namun bentuk segmentasi tampak kurang presisi dibandingkan skenario sebelumnya. Secara keseluruhan, hasil visual menunjukkan bahwa model masih mampu melakukan segmentasi, tetapi tingkat ketepatan dan konsistensinya lebih rendah dibandingkan skenario dengan *learning rate* yang lebih besar.



Gambar 4. 6 Hasil Segmentasi Skenario 3

Analisis kuantitatif dilakukan berdasarkan *confusion matrix* pada masing-masing data uji. Pada data (a) dalam Tabel 4. 7, jumlah *Actual Pixel* Tumor adalah sebesar 1327 piksel (1327 *true positive* + 0 *false negative*), sedangkan *Predicted*

Pixel Tumor sebesar 2193 piksel (1327 *true positive* + 866 *false positive*). Pada data (b) dalam Tabel 4. 8, jumlah *Actual Pixel* Tumor sebesar 1342 piksel (1282 + 60), dan *Predicted Pixel* Tumor sebesar 1335 piksel (1282 + 53). Pada data (c) dalam Tabel 4. 9, jumlah *Actual Pixel* Tumor sebesar 2156 piksel (1749 + 407), dan *Predicted Pixel* Tumor sebesar 2114 piksel (1749 + 365). Hasil ini menunjukkan bahwa meskipun model mampu mendeteksi sebagian besar area tumor, masih terdapat jumlah *false positive* dan *false negative* yang cukup signifikan, terutama pada citra (a) dan (c).

Tabel 4. 7 *Confusion Matrix* Skenario 3 Gambar 47.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	63343	866
	Tumor	0	1327

Tabel 4. 8 *Confusion Matrix* Skenario 3 Gambar 232.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	64141	53
	Tumor	60	1282

Tabel 4. 9 *Confusion Matrix* Skenario 3 Gambar 2812.png

		Prediksi	
		No Tumor	Tumor
Aktual	No Tumor	63013	365
	Tumor	407	1749

Secara keseluruhan, nilai *true positive* yang tinggi menunjukkan bahwa model tetap mampu mengenali area tumor, namun tingginya nilai *false positive* pada beberapa citra mengindikasikan bahwa model cenderung mendeteksi area non-tumor sebagai tumor. Selain itu, keberadaan *false negative* pada beberapa data menunjukkan bahwa masih terdapat bagian tumor yang tidak tersegmentasi dengan baik. Dengan demikian, baik dari analisis kualitatif maupun kuantitatif, performa model pada skenario ketiga dinilai kurang optimal dibandingkan skenario sebelumnya. Hal ini menunjukkan bahwa penggunaan *learning rate* yang terlalu kecil dapat menghambat proses pembelajaran model dan berdampak pada penurunan kualitas segmentasi yang dihasilkan.

4.2.4 Hasil Perbandingan *Testing* Setiap Skenario

Pada subbab ini dilakukan evaluasi kinerja model segmentasi tumor otak berbasis *Lightweight* U-Net untuk menilai kemampuan model dalam memetakan area tumor pada citra MRI secara akurat. Evaluasi kinerja dilakukan menggunakan dua metrik utama, yaitu *Dice Coefficient* dan *Pixel Accuracy*. *Dice Coefficient* digunakan untuk mengukur tingkat kesesuaian antara hasil segmentasi model dan *mask ground truth*, khususnya pada area tumor, sedangkan *Pixel Accuracy*

digunakan untuk menilai ketepatan klasifikasi piksel secara keseluruhan. Selain kedua metrik utama tersebut, *Sensitivity* dan *Precision* digunakan sebagai metrik pendukung untuk analisis lebih lanjut terhadap kemampuan model dalam mendeteksi area tumor dan membedakannya dari latar belakang. Pemilihan keempat metrik ini didasarkan pada penggunaannya yang luas dalam penelitian segmentasi citra medis serta kemampuannya dalam memberikan gambaran performa model yang lebih komprehensif.

Tabel 4. 10 menyajikan hasil evaluasi performa model *Lightweight* U-Net pada setiap skenario pengujian menggunakan metrik *Pixel Accuracy*, *Dice Coefficient*, *Sensitivity*, dan *Precision*. Secara umum, seluruh skenario menunjukkan nilai *Pixel Accuracy* yang sangat tinggi. Skenario 1 dan Skenario 2 memiliki nilai yang hampir identik, masing-masing sebesar 0,9928 dan 0,9925, sedangkan Skenario 3 menunjukkan nilai yang sedikit lebih rendah yaitu 0,9909. Hasil ini menunjukkan bahwa model mampu melakukan klasifikasi piksel secara keseluruhan dengan tingkat ketepatan yang sangat baik pada seluruh konfigurasi pengujian.

Perbedaan performa yang lebih jelas terlihat pada metrik *Dice Coefficient*, *Sensitivity*, dan *Precision* yang secara spesifik merepresentasikan kualitas segmentasi area tumor. Skenario 1 menghasilkan nilai *Dice Coefficient* tertinggi sebesar 0,7483, diikuti oleh Skenario 2 sebesar 0,7268, sedangkan Skenario 3 menunjukkan nilai yang lebih rendah yaitu 0,6615. Pola yang serupa juga terlihat pada metrik *Sensitivity*, di mana Skenario 1 memperoleh nilai 0,7821, Skenario 2 sebesar 0,7582, dan Skenario 3 sebesar 0,6954, yang menunjukkan bahwa

kemampuan deteksi area tumor pada Skenario 1 lebih baik dibandingkan skenario lainnya. Selain itu, nilai *Precision* pada Skenario 1 dan Skenario 2 masing-masing sebesar 0,7769 dan 0,7713, lebih tinggi dibandingkan Skenario 3 yang memperoleh nilai 0,7195, yang menunjukkan bahwa kesalahan klasifikasi area non-tumor sebagai tumor dapat diminimalkan dengan lebih baik pada kedua skenario tersebut.

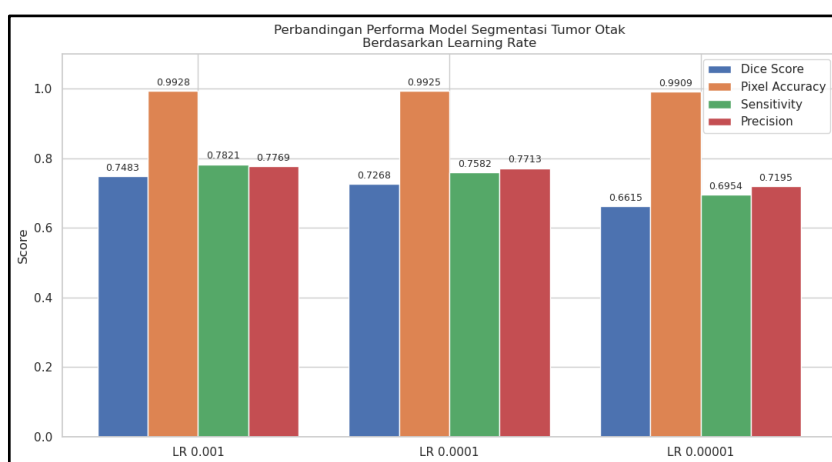
Pixel Accuracy cenderung kurang sensitif dalam merepresentasikan kualitas segmentasi tumor secara spesifik karena dipengaruhi oleh dominasi piksel non-tumor. Sebaliknya, *Dice Coefficient*, *Sensitivity*, dan *Precision* memberikan gambaran yang lebih representatif terhadap kemampuan model dalam mendeteksi dan memetakan area tumor secara akurat. Secara keseluruhan, Skenario 1 menunjukkan performa terbaik dengan keseimbangan yang baik antara ketepatan deteksi tumor dan minimnya kesalahan klasifikasi. Skenario 2 memiliki performa yang relatif mendekati Skenario 1, sedangkan Skenario 3 menunjukkan performa yang lebih rendah terutama dalam mempertahankan ketepatan segmentasi area tumor.

Tabel 4. 10 Hasil Performa Setiap Skenario

No.	Skenario	<i>Pixel Accuracy</i>	<i>Dice Coefficient</i>	<i>Sensitivity</i>	<i>Precision</i>
1.	Skenario 1	0.9928	0.7483	0.7821	0.7769
2.	Skenario 2	0.9925	0.7268	0.7582	0.7713
3.	Skenario 3	0.9909	0.6615	0.6954	0.7195

Gambar 4. 7 menyajikan visualisasi perbandingan performa model Lightweight U-Net pada tahap pengujian dalam bentuk diagram batang berdasarkan tiga variasi *learning rate*, yaitu 0,001; 0,0001; dan 0,00001. Perbandingan dilakukan menggunakan empat metrik evaluasi, yaitu *Dice Coefficient*, *Pixel Accuracy*, *Sensitivity*, dan *Precision*. Secara umum, seluruh skenario menunjukkan

nilai *Pixel Accuracy* yang sangat tinggi dan relatif seragam, yaitu berada di atas 0,99. Hal ini mengindikasikan bahwa model mampu mengklasifikasikan piksel citra MRI dengan tingkat ketepatan yang sangat baik secara keseluruhan. Namun demikian, karena dominasi piksel non-tumor dalam citra, metrik *Pixel Accuracy* menjadi kurang sensitif dalam merepresentasikan kualitas segmentasi area tumor secara spesifik.



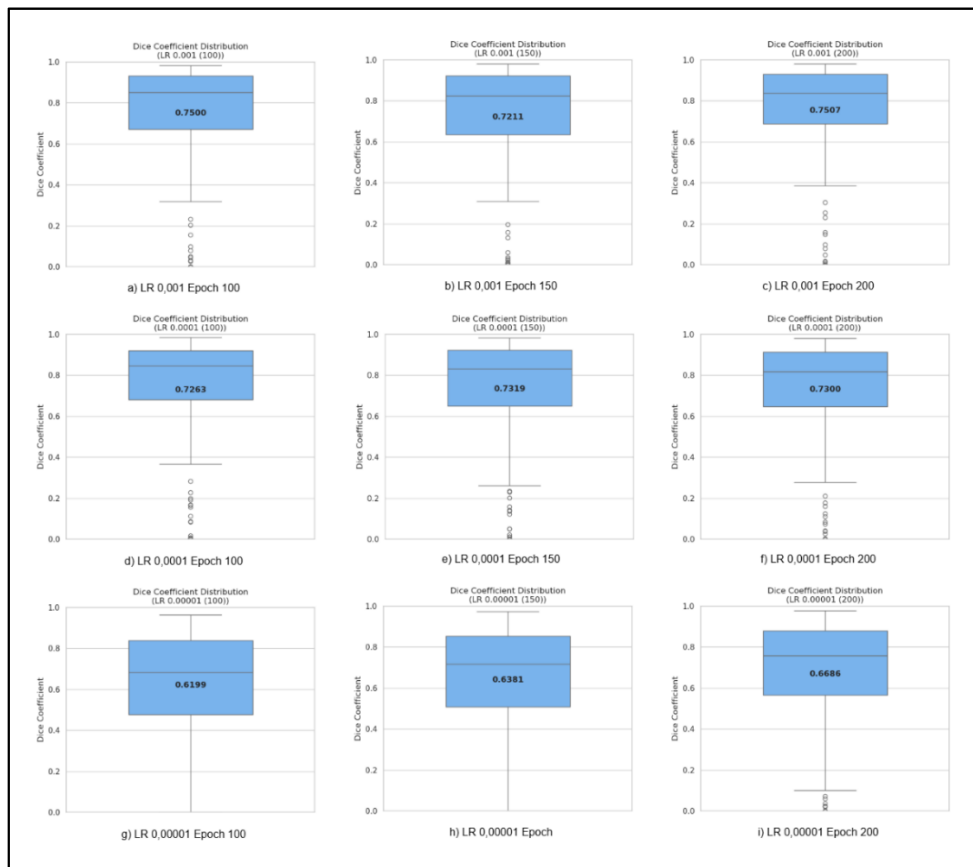
Gambar 4. 7 Hasil Perbandingan Setiap Skenario

Perbedaan performa antar skenario terlihat lebih jelas pada metrik *Dice Coefficient*, *Sensitivity*, dan *Precision* yang divisualisasikan melalui tinggi batang pada masing-masing kategori. *Learning rate* 0,001 menunjukkan nilai *Dice Coefficient* dengan nilai 0,7483 dan *Sensitivity* dengan nilai 0,7821 tertinggi, yang menandakan kemampuan model dalam mendeteksi area tumor secara lebih optimal. Pada *learning rate* 0,0001, nilai *Precision* 0,7713 menjadi yang tertinggi, yang menunjukkan kemampuan model dalam meminimalkan kesalahan prediksi pada area non-tumor. Sementara itu, *learning rate* 0,00001 menghasilkan nilai terendah pada sebagian besar metrik, khususnya *Dice Coefficient* dengan nilai 0,6615 dan *Sensitivity* dengan nilai 0,6954, yang mengindikasikan penurunan kemampuan

model dalam mengidentifikasi area tumor. Berdasarkan hasil tersebut, dapat disimpulkan bahwa *learning rate* 0,001 memberikan performa segmentasi yang paling optimal dibandingkan skenario lainnya.

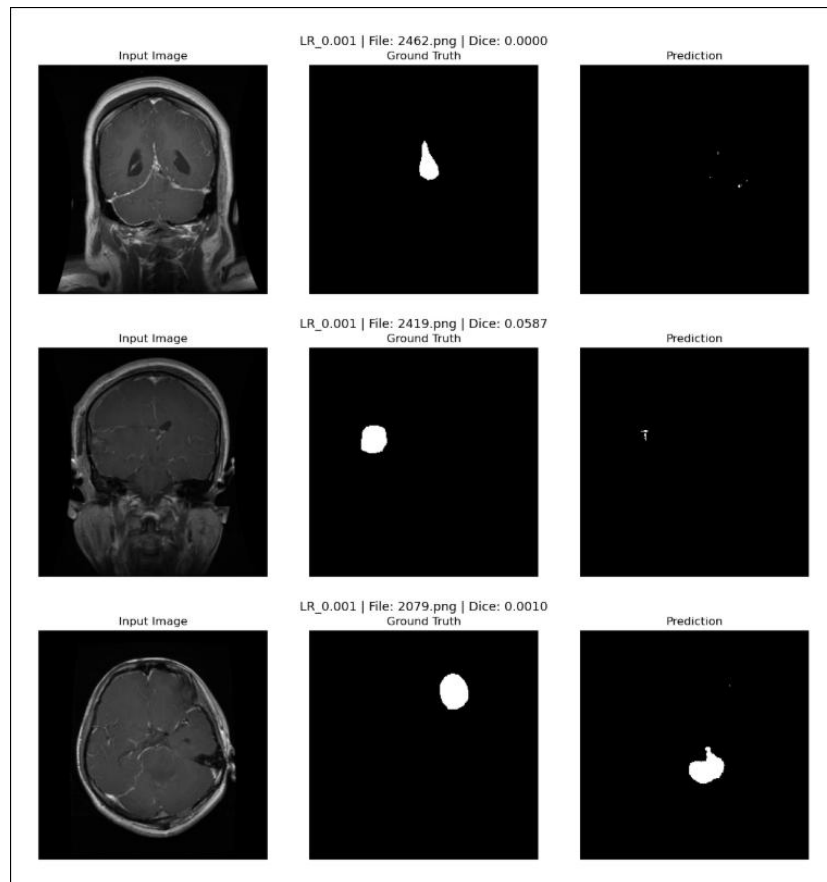
Gambar 4. 8, ditunjukkan visualisasi boxplot nilai *Dice Coefficient* dari hasil pengujian model Lightweight U-Net pada tiga pengaturan *learning rate*, yaitu 0,001; 0,0001; dan 0,00001, dengan variasi jumlah *epoch*. Visualisasi ini digunakan untuk menganalisis sebaran nilai *Dice Coefficient* serta kestabilan performa model pada seluruh data uji. Pada masing-masing boxplot, garis tengah menunjukkan nilai median, sedangkan posisi kotak merepresentasikan rentang interkuartil yang menggambarkan distribusi utama data. Selain itu, titik-titik di luar *whisker* menunjukkan keberadaan *outlier* yang mencerminkan sampel dengan performa segmentasi yang menyimpang dari mayoritas data.

Gambar 4.8(a)–(c) menunjukkan hasil untuk *learning rate* 0,001 pada *epoch* 100, 150, dan 200, yang memiliki distribusi nilai *Dice* yang relatif tinggi dan stabil. Gambar 4.8(d)–(f) merepresentasikan *learning rate* 0,0001, yang juga menunjukkan median tinggi dengan sebaran yang cukup konsisten di setiap variasi *epoch*. Gambar 4.8(g)–(i) memperlihatkan hasil pada *learning rate* 0,00001, dimana nilai median cenderung lebih rendah dengan rentang distribusi yang lebih lebar. Selain itu, jumlah *outlier* dengan nilai *Dice* rendah terlihat lebih banyak pada skenario ini, yang mengindikasikan bahwa model kurang mampu melakukan segmentasi secara optimal pada beberapa citra tertentu. Secara keseluruhan, distribusi pada *learning rate* 0,001 dan 0,0001 cenderung lebih rapat dan stabil dibandingkan *learning rate* 0,00001.



Gambar 4. 8 Hasil Perbandingan Nilai *Dice Coefficient*

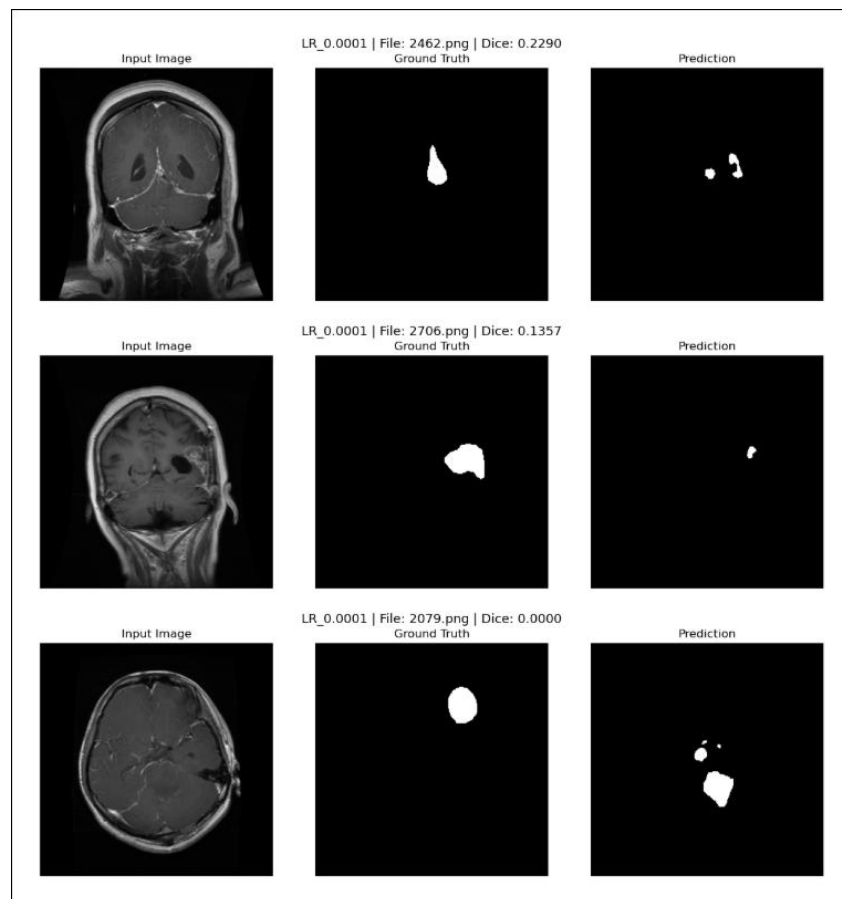
Hasil evaluasi menunjukkan bahwa model Lightweight U-Net mampu mencapai performa segmentasi tumor otak yang baik pada seluruh skenario pengujian, meskipun terdapat variasi performa pada beberapa data uji. Keberadaan nilai *Dice Coefficient* yang berada jauh di bawah median pada setiap skenario mengindikasikan adanya outlier, yaitu sampel citra yang menghasilkan segmentasi kurang optimal. Outlier ini berkontribusi terhadap penyebaran nilai dan mencerminkan adanya kasus-kasus tertentu yang lebih sulit untuk diproses oleh model. Analisis selanjutnya difokuskan pada pembahasan contoh data outlier tersebut untuk mengidentifikasi karakteristik citra yang memengaruhi kegagalan segmentasi serta memahami keterbatasan model dalam menangani variasi data uji.



Gambar 4. 9 Sample Outlier Skenario 1

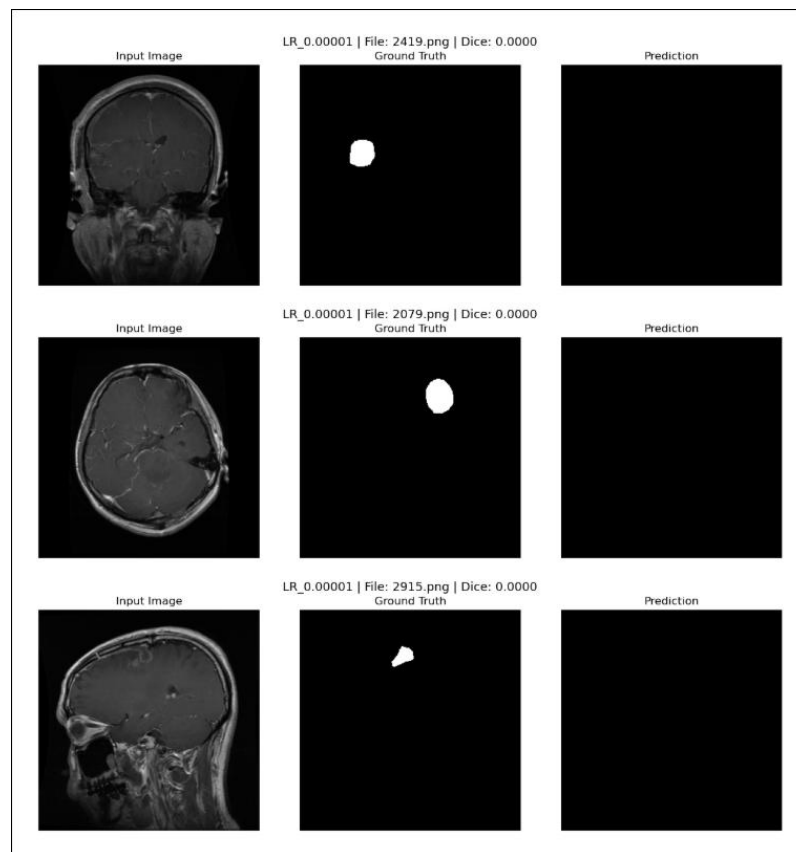
Gambar 4.9 menunjukkan beberapa contoh outlier pada skenario *learning rate* 0,001. Pada File 2462.png, model gagal mendeteksi area tumor yang terlihat jelas pada *ground truth*, menghasilkan nilai *Dice Coefficient* 0,0000. Pada File 2419.png, model hanya mendeteksi sebagian kecil area tumor sehingga menghasilkan nilai *Dice* yang sangat rendah. Pada File 2079.png, model berhasil mendeteksi area tumor namun terdapat ketidaksesuaian signifikan pada bentuk dan lokasi prediksi. Ketiga kasus ini mengindikasikan dominasi *false negative* dan ketidaktepatan representasi area tumor, yang menunjukkan bahwa model masih memiliki keterbatasan dalam menangani variasi ukuran, bentuk, dan lokasi tumor pada kasus-kasus tertentu.

Pada Gambar 4. 10, ditunjukkan beberapa contoh outlier pada skenario *learning rate* 0,0001 yang diidentifikasi berdasarkan nama file pada masing-masing citra. Pada File: 2462.png, model mampu mendeteksi sebagian area tumor, namun hasil prediksi masih terfragmentasi dan tidak sepenuhnya mengikuti bentuk ground truth, sehingga menghasilkan nilai *Dice Coefficient* yang relatif rendah. Pada File: 2706.png, model hanya mendeteksi sebagian kecil area tumor dengan ukuran yang jauh lebih kecil dibandingkan ground truth, yang menunjukkan adanya *under-segmentation*.



Gambar 4. 10 Sample Outlier Skenario 2

Pada File: 2079.png, model gagal merepresentasikan area tumor dengan baik karena lokasi prediksi tidak sesuai dengan ground truth, sehingga menghasilkan nilai *Dice Coefficient* sebesar 0,0000. Secara keseluruhan, contoh-contoh tersebut menunjukkan bahwa meskipun *learning rate* 0,0001 memberikan hasil yang relatif lebih stabil, masih terdapat kesalahan segmentasi berupa false negative serta ketidaksesuaian bentuk dan lokasi tumor yang memengaruhi variasi nilai *Dice Coefficient*.



Gambar 4. 11 Sample Outlier Skenario 3

Gambar 4.11 menunjukkan beberapa contoh outlier pada skenario learning rate 0,00001. Pada File 2419.png dan File 2079.png, model gagal mendeteksi area tumor sama sekali meskipun ground truth menunjukkan keberadaan tumor yang

kelas, menghasilkan nilai Dice Coefficient 0,0000. Pada File 2915.png, model juga tidak menghasilkan prediksi terhadap area tumor yang ada. Ketiga kasus ini menunjukkan dominasi false negative akibat learning rate yang terlalu kecil, sehingga model tidak mampu menangkap pola penting dari data dan mengalami kegagalan segmentasi.

Adanya variasi nilai performa serta ditemukannya beberapa data *outlier* pada hasil pengujian sebelumnya menunjukkan bahwa pemilihan parameter pada setiap skenario memberikan dampak yang berbeda terhadap kemampuan segmentasi model. Untuk membuktikan apakah perbedaan performa tersebut bersifat signifikan secara statistik dan bukan terjadi karena faktor kebetulan, maka dilakukan analisis inferensial menggunakan uji signifikansi.

Metode yang digunakan dalam penelitian ini adalah *Paired Sample T-Test*. Uji ini dipilih untuk memvalidasi perbandingan nilai rata-rata dari dua metrik evaluasi utama, yaitu *Dice Coefficient* dan *Pixel Accuracy*, yang dihasilkan oleh model pada skenario yang berbeda. Secara matematis, prosedur perhitungan nilai t dilakukan dengan menggunakan rumus sebagai berikut:

$$t = \frac{\bar{d}}{\frac{s_d}{\sqrt{n}}} \quad (4.1)$$

Keterangan:

t = nilai statistic uji (t-statistic)

\bar{d} = rata-rata selisih antara dua kelompok data

s_d = standar deviasi dari selisih

\sqrt{n} = akar dari jumlah sample

Hasil dari perhitungan melalui Rumus 4.1 kemudian dianalisis menggunakan nilai signifikansi (p -value). Dalam penelitian ini, digunakan taraf

signifikansi sebesar $\alpha = 0,05$ sebagai batas pengambilan keputusan. Suatu hasil uji dikatakan signifikan apabila nilai *p-value* lebih kecil dari 0,05 ($p\text{-value} < 0,05$), sedangkan hasil uji dinyatakan tidak signifikan apabila nilai *p-value* lebih besar atau sama dengan 0,05 ($p\text{-value} \geq 0,05$). Dengan demikian, jika nilai *p-value* berada di bawah batas tersebut, maka dapat disimpulkan bahwa terdapat perbedaan performa yang signifikan secara statistik antara skenario yang diuji. Sebaliknya, jika nilai *p-value* berada di atas atau sama dengan batas tersebut, maka perbedaan yang terjadi tidak bersifat signifikan secara statistik.

Hasil uji statistik menggunakan *T-Test* terhadap nilai *Dice Coefficient* ditunjukkan pada Tabel 4. 11 dalam bentuk matriks *p-value* untuk setiap pasangan skenario *learning rate*. Perbandingan antara LR 0,001 dan LR 0,0001 menghasilkan nilai *p-value* sebesar 1.233955×10^{-1} , yang lebih besar dari 0,05 sehingga menunjukkan bahwa perbedaan performa kedua model tidak signifikan secara statistik. Sebaliknya, perbandingan antara LR 0,001 dengan LR 0,00001 menghasilkan *p-value* sebesar 2.509778×10^{-25} , sedangkan perbandingan antara LR 0,0001 dengan LR 0,00001 menghasilkan *p-value* sebesar 1.037852×10^{-25} . Nilai *p-value* yang sangat kecil pada kedua perbandingan tersebut mengindikasikan adanya perbedaan performa yang sangat signifikan antara model dengan *learning rate* 0,00001 terhadap dua skenario lainnya.

Tabel 4. 11 Hasil *p-value* Uji *T-Test Dice Coefficient* Antar Skenario *Learning Rate*

	LR 0,001	LR 0,0001	0,00001
LR 0,001	-	1.233955×10^{-1}	2.509778×10^{-25}
LR 0,0001	1.233955×10^{-1}	-	1.037852×10^{-25}
LR 0,00001	2.509778×10^{-25}	1.037852×10^{-25}	-

Interpretasi hasil uji pada Tabel 4.12 terhadap nilai *Dice Coefficient* ditunjukkan melalui klasifikasi signifikansi, dimana simbol S (Signifikan) menunjukkan *p-value* < 0,05 dan TS (Tidak Signifikan) menunjukkan *p-value* ≥ 0,05. Hasil menunjukkan bahwa perbandingan antara LR 0,001 dan LR 0,0001 termasuk kategori TS, sehingga tidak terdapat perbedaan performa yang signifikan. Sebaliknya, seluruh perbandingan yang melibatkan LR 0,00001 termasuk kategori S, yang menandakan adanya perbedaan performa yang signifikan. Dengan demikian, penggunaan *learning rate* 0,00001 memberikan pengaruh yang signifikan terhadap penurunan performa segmentasi berdasarkan nilai *Dice Coefficient*, sedangkan perbedaan antara LR 0,001 dan 0,0001 tidak signifikan secara statistik.

Tabel 4. 12 Keputusan Uji *T-Test Dice Coefficient* Antar *Learning Rate*

	LR 0,001	LR 0,0001	0,00001
LR 0,001	-	TS	S
LR 0,0001	TS	-	S
LR 0,00001	S	S	-

Hasil uji statistik menggunakan *T-Test* terhadap nilai *Pixel Accuracy* ditunjukkan pada Tabel 4. 13 dalam bentuk matriks *p-value* untuk setiap pasangan skenario *learning rate*. Perbandingan antara LR 0,001 dan LR 0,0001 menghasilkan nilai *p-value* sebesar 3.358635×10^{-1} , yang lebih besar dari 0,05 sehingga menunjukkan bahwa perbedaan performa kedua model tidak signifikan secara statistik. Sementara itu, perbandingan antara LR 0,001 dengan LR 0,00001 menghasilkan *p-value* sebesar 5.145594×10^{-24} , dan perbandingan antara LR 0,0001 dengan LR 0,00001 menghasilkan *p-value* sebesar 7.725202×10^{-27} . Nilai *p-value* yang sangat kecil pada kedua perbandingan tersebut menunjukkan adanya perbedaan performa yang sangat signifikan, khususnya pada model dengan *learning rate* 0,00001 dibandingkan dengan dua skenario lainnya.

Tabel 4. 13 Hasil *p-value* Uji *T-Test Pixel Accuracy* Antar Skenario *Learning Rate*

	LR 0,001	LR 0,0001	0,00001
LR 0,001	-	3.358635×10^{-1}	5.145594×10^{-24}
LR 0,0001	3.358635×10^{-1}	-	7.725202×10^{-27}
LR 0,00001	5.145594×10^{-24}	7.725202×10^{-27}	-

Interpretasi hasil uji T-Test pada Tabel 4.14 terhadap nilai *Pixel Accuracy* ditunjukkan melalui klasifikasi signifikansi, dimana simbol S (Signifikan) menunjukkan *p-value* < 0,05 dan TS (Tidak Signifikan) menunjukkan *p-value* ≥ 0,05. Hasil menunjukkan bahwa perbandingan antara LR 0,001 dan LR 0,0001 termasuk kategori TS, sehingga tidak terdapat perbedaan performa yang signifikan. Sebaliknya, seluruh perbandingan yang melibatkan LR 0,00001 termasuk kategori S, yang menandakan adanya perbedaan performa yang signifikan. Dengan demikian, penggunaan *learning rate* 0,00001 memberikan pengaruh yang signifikan terhadap penurunan performa berdasarkan nilai *Pixel Accuracy*, sedangkan perbedaan antara LR 0,001 dan 0,0001 tidak signifikan secara statistik.

Tabel 4. 14 Keputusan Uji *T-Test Pixel Accuracy* Antar *Learning Rate*

	LR 0,001	LR 0,0001	0,00001
LR 0,001	-	TS	S
LR 0,0001	TS	-	S
LR 0,00001	S	S	-

Berdasarkan hasil pengujian menggunakan uji *paired T-Test* terhadap nilai *Dice Coefficient* pada variasi *epoch*, yaitu Tabel 4. 15, diperoleh bahwa perbandingan antara *epoch* 100 dan 150 menghasilkan nilai *p-value* sebesar 7.449134×10^{-1} , yang lebih besar dari 0,05 sehingga dinyatakan tidak signifikan, sehingga dinyatakan tidak signifikan. Hal ini menunjukkan bahwa tidak terdapat perbedaan yang bermakna antara performa model pada *epoch* 100 dan 150. Sebaliknya, perbandingan antara *epoch* 100 dengan 200 serta *epoch* 150 dengan 200 masing-masing menghasilkan nilai *p-value* sebesar 6.251385×10^{-4} dan 6.833615×10^{-4} , yang lebih kecil dari 0,05, sehingga dinyatakan signifikan. Dengan demikian, terdapat perbedaan performa yang bermakna ketika *epoch* ditingkatkan hingga 200.

Tabel 4. 15 Hasil *p-value* Uji *T-Test Dice Coefficient* Antar *Epoch Learning Rate*

	100	150	200
100	-	7.449134×10^{-1}	6.251385×10^{-4}
150	7.449134×10^{-1}	-	6.833615×10^{-4}
200	6.251385×10^{-4}	6.833615×10^{-4}	-

Hasil pada Tabel 4.16 menunjukkan bahwa peningkatan *epoch* hingga 200 memberikan pengaruh yang signifikan terhadap performa model berdasarkan nilai *Dice Coefficient*, sedangkan peningkatan dari *epoch* 100 ke 150 tidak menunjukkan perbedaan yang signifikan. Hal ini mengindikasikan bahwa model mulai mencapai kestabilan performa pada rentang tersebut. *Epoch* 200 dapat dianggap sebagai

konfigurasi yang lebih optimal karena memberikan peningkatan performa yang signifikan secara statistik.

Tabel 4. 16 Keputusan Uji *T-Test Dice Coefficient* Antar *Epoch Learning Rate*

	100	150	200
100	-	TS	S
150	TS	-	S
200	S	S	-

Berdasarkan hasil pengujian menggunakan uji *paired t-test* terhadap nilai *Pixel Accuracy* pada variasi *epoch*, yaitu Tabel 4. 17, diperoleh bahwa seluruh pasangan perbandingan menunjukkan nilai *p-value* yang lebih kecil dari tingkat signifikansi ($\alpha = 0,05$). Perbandingan antara *epoch* 100 dan 150 menghasilkan *p-value* sebesar 1.477949×10^{-2} , sedangkan perbandingan antara *epoch* 100 dengan 200 dan *epoch* 150 dengan 200 masing-masing menghasilkan *p-value* sebesar 3.237414×10^{-7} dan 7.066466×10^{-3} . Karena seluruh nilai *p-value* tersebut berada di bawah ambang batas signifikansi, maka seluruh perbandingan dinyatakan signifikan. Hal ini menunjukkan bahwa terdapat perbedaan performa yang bermakna pada setiap peningkatan jumlah *epoch* terhadap metrik *Pixel Accuracy*.

Tabel 4. 17 Hasil *p-value* Uji *T-Test Pixel Accuracy* Antar *Epoch Learning Rate*

	100	150	200
100	-	1.477949×10^{-2}	3.237414×10^{-7}
150	1.477949×10^{-2}	-	7.066466×10^{-3}
200	3.237414×10^{-7}	7.066466×10^{-3}	-

Hasil pada Tabel 4.18 menunjukkan bahwa peningkatan epoch memberikan pengaruh signifikan terhadap performa model berdasarkan nilai *Pixel Accuracy*. Berbeda dengan *Dice Coefficient*, metrik ini menunjukkan sensitivitas yang lebih tinggi terhadap perubahan epoch. Peningkatan jumlah epoch cenderung meningkatkan akurasi piksel secara signifikan, sehingga epoch 200 dapat dianggap sebagai konfigurasi yang paling optimal.

Tabel 4. 18 Keputusan Uji *T-Test Pixel Accuracy* Antar *Epoch Learning Rate*

	100	150	200
100	-	S	S
150	S	-	S
200	S	S	-

4.3 Integrasi Islam

Berdasarkan hasil pengujian yang telah dilakukan pada penelitian ini, metode segmentasi tumor otak menggunakan *Lightweight U-Net* berbasis citra MRI otak menunjukkan kinerja yang baik dalam mengidentifikasi area tumor. Hasil segmentasi yang diperoleh mampu membedakan jaringan tumor dan jaringan

normal dengan cukup akurat, sehingga sistem yang dikembangkan berpotensi membantu tenaga medis dalam proses diagnosis dan perencanaan penanganan lebih lanjut. Keberhasilan ini menunjukkan bahwa tujuan penelitian dalam mendukung peningkatan kualitas diagnosis penyakit tumor otak telah tercapai.

Dalam perspektif Islam, hasil penelitian yang memberikan manfaat nyata bagi kesehatan manusia memiliki nilai kemaslahatan yang tinggi. Segmentasi tumor otak yang akurat dapat membantu proses deteksi dini serta mengurangi risiko kesalahan dalam diagnosis, sehingga peluang penanganan yang tepat dan cepat menjadi lebih besar. Upaya ini sejalan dengan ajaran Islam yang menempatkan kesehatan sebagai amanah yang harus dijaga, serta mendorong umat manusia untuk menjaga keselamatan dan kualitas hidupnya.

Hasil penelitian ini juga dapat dikaitkan dengan konsep *Maqashid Syariah*, khususnya *Hifzh an-Nafs* (menjaga jiwa), karena penelitian di bidang kesehatan secara langsung bertujuan untuk melindungi dan menyelamatkan kehidupan manusia. Selain itu, pemanfaatan model kecerdasan buatan seperti *Lightweight U-Net* dalam segmentasi tumor otak mencerminkan *Hifzh al-'Aql*, yaitu penggunaan akal dan ilmu pengetahuan secara optimal demi kemaslahatan. Pengembangan dan penerapan teknologi ini menunjukkan bahwa ilmu pengetahuan dapat menjadi sarana untuk mencapai tujuan-tujuan syariat Islam.

Keberhasilan sistem segmentasi tumor otak yang dihasilkan dalam penelitian ini juga mencerminkan prinsip ikhtiar dalam Islam. Islam mengajarkan bahwa manusia tidak hanya diperintahkan untuk bersabar dalam menghadapi ujian berupa penyakit, tetapi juga diwajibkan untuk berusaha secara maksimal dalam

mencari solusi dan pengobatan yang tepat. Hal ini sebagaimana firman Allah SWT dalam Al-Qur'an Surah Al-Baqarah ayat 195:

وَأَنْفِقُوا فِي سَبِيلِ اللَّهِ وَلَا تُلْقُوا بِأَيْدِيكُمْ إِلَى التَّهْلُكَةِ وَأَحْسِنُوا إِنَّ اللَّهَ يُحِبُّ الْمُحْسِنِينَ

“Dan belanjakanlah (hartamu) di jalan Allah, dan janganlah kamu menjatuhkan dirimu sendiri ke dalam kebinasaan, dan berbuat baiklah. Sesungguhnya Allah menyukai orang-orang yang berbuat baik.” (QS. Al-Baqarah: 195)

Ayat tersebut menegaskan larangan bagi manusia untuk membiarkan dirinya berada dalam kondisi yang dapat membawa kepada kebinasaan, termasuk akibat keterlambatan dalam mendeteksi dan menangani penyakit serius seperti tumor otak. Pemanfaatan teknologi pengolahan citra medis dan kecerdasan buatan dalam penelitian ini dapat dipandang sebagai bentuk ikhtiar manusia untuk mencegah dampak yang lebih buruk melalui proses diagnosis yang lebih cepat dan akurat.

Menurut tafsir yang bersumber dari Imam As-Suyuthi dalam *Tafsirul Jalalain*, QS. Al-Baqarah ayat 195 mengandung perintah untuk berinfak di jalan Allah, yaitu menggunakan harta, tenaga, dan kemampuan dalam berbagai bentuk ketaatan serta kemaslahatan. Ayat ini juga mengandung larangan menjerumuskan diri ke dalam kebinasaan, yang salah satunya dapat terjadi akibat meninggalkan kewajiban berinfak atau tidak memanfaatkan sumber daya yang dimiliki secara optimal. Ayat tersebut menekankan pentingnya berbuat ihsan sebagai bentuk ketaatan yang bernilai tinggi di sisi Allah (Ubab, 2021). Selain ikhtiar, Islam juga menekankan pentingnya kesabaran dalam menghadapi ujian, sebagaimana firman Allah SWT dalam Surah Al-Baqarah ayat 153:

يَا أَيُّهَا الَّذِينَ آمَنُوا اسْتَعِينُوا بِالصَّبْرِ وَالصَّلَاةِ إِنَّ اللَّهَ مَعَ الصَّابِرِينَ

“Hai orang-orang yang beriman, jadikanlah sabar dan salat sebagai penolongmu, sesungguhnya Allah bersama orang-orang yang sabar.” (QS. Al-Baqarah: 153).

Kesabaran dalam menghadapi penyakit tidak meniadakan kewajiban untuk berusaha, melainkan berjalan seiring dengan ikhtiar melalui pemanfaatan ilmu pengetahuan dan teknologi. Keberhasilan segmentasi tumor otak dalam penelitian ini mencerminkan keseimbangan antara usaha manusia dan nilai-nilai spiritual yang diajarkan dalam Islam.

Hasil penelitian ini tidak hanya memberikan kontribusi dalam pengembangan teknologi segmentasi citra medis, tetapi juga selaras dengan nilai-nilai Islam yang menekankan pentingnya menjaga kehidupan, memanfaatkan akal dan ilmu pengetahuan, serta berikhtiar dalam menghadapi penyakit. Integrasi antara hasil penelitian dan nilai Islam ini menunjukkan bahwa pengembangan teknologi di bidang kesehatan dapat menjadi bagian dari upaya mewujudkan kemaslahatan umat dan pengabdian kepada kemanusiaan.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Hasil pengujian menggunakan beberapa skenario *Learning Rate* dan *Epoch* menunjukkan bahwa model *Lightweight* U-Net mampu menghasilkan segmentasi tumor otak pada citra MRI dengan performa yang baik. Evaluasi menggunakan metrik *Dice Coefficient* dan *Pixel Accuracy* menunjukkan seluruh skenario memiliki nilai *Pixel Accuracy* di atas 0,99, yang menandakan kemampuan model dalam mengklasifikasikan piksel secara akurat. Nilai *Dice Coefficient* terbaik mencapai 0,7483, yang menunjukkan tingkat kesesuaian yang baik antara hasil segmentasi dan *ground truth*.

Berdasarkan hasil uji *paired sample T-Test*, *Learning Rate* 0,001 dan 0,0001 tidak memiliki perbedaan performa yang signifikan terhadap *Dice Coefficient* maupun *Pixel Accuracy*, sehingga keduanya menunjukkan kestabilan pembelajaran yang baik. *Learning Rate* 0,00001 menghasilkan performa yang secara signifikan lebih rendah, yang menunjukkan proses pelatihan belum mencapai kondisi optimal. Berdasarkan hasil tersebut, kombinasi *Learning Rate* 0,001 dengan 200 *epoch* dapat dianggap sebagai konfigurasi yang paling optimal atau paling mendekati kondisi konvergen dalam penelitian ini, karena menghasilkan nilai *Dice Coefficient* tertinggi dan *Pixel Accuracy* yang konsisten tinggi. Hal ini menunjukkan bahwa pemilihan *Learning Rate* yang tidak terlalu kecil serta jumlah *epoch* yang cukup berperan penting dalam mencapai performa segmentasi yang optimal dan stabil.

5.2 Saran

Berdasarkan hasil penelitian yang telah dilakukan, terdapat beberapa saran yang dapat dijadikan acuan untuk penelitian selanjutnya, antara lain:

1. Melakukan eksplorasi arsitektur segmentasi lain atau melakukan modifikasi lebih lanjut pada *Lightweight* U-Net guna meningkatkan akurasi segmentasi tanpa menambah kompleksitas komputasi secara signifikan.
2. Melakukan eksperimen dengan jumlah data yang lebih besar atau berasal dari berbagai sumber untuk menguji ketahanan dan konsistensi performa model *Lightweight* U-Net pada kondisi data yang berbeda.
3. Melakukan eksperimen terhadap penggunaan metrik evaluasi selain *Dice Coefficient*, *Pixel Accuracy*, Sensitifitas (*Recall*), dan *Precision*. Penelitian lanjutan disarankan untuk menambahkan metrik evaluasi lain seperti *Area Under the Curve* (AUC), *F1-Score*, dan *Intersection over Union* (IoU) agar analisis performa segmentasi menjadi lebih komprehensif.

DAFTAR PUSTAKA

- Abbas, M., Khalaf, A. A. M., Mogahed, H., Hussein, A. I., Gaber, L., & Mabrook, M. (2026). An integrated hybrid U-Net and EfficientNetV2-S approach for brain tumor segmentation and classification. *Bulletin of Electrical Engineering and Informatics*, 15(1), 702–711. <https://doi.org/10.11591/eei.v15i1.9818>
- Alali, A. Z., & Hussein Ali, K. (2022). Segmentation of Human Brain Gliomas Tumour Images using U-Net Architecture with Transfer Learning. *Diyala Journal of Engineering Sciences*, 8716(1), 17–29. <https://doi.org/10.24237/djes.2022.15102>
- Ardan, I. S., & Indraswari, R. (2024). Sistem Berbasis Deep Learning untuk Segmentasi dan Klasifikasi Tingkat Keganasan Tumor Otak Menggunakan Citra MRI 3D. *Journal of Computer Science and Applied Informatics*, 6(2), 117–126.
- Arora, A., Jayal, A., Gupta, M., Mittal, P., & Satapathy, S. C. (2021). Brain tumor segmentation of MRI images using processed image driven u-net architecture. *Computers*, 10(11). <https://doi.org/10.3390/computers10110139>
- Cheng, J. (2015). *brain tumor dataset*. Figshare. https://figshare.com/articles/dataset/brain_tumor_dataset/1512427/1
- Diana-Albelda, C., García-Martín, Á., & Bescos, J. (2025). A Review on Deep Learning Methods for Glioma Segmentation, Limitations, and Future Perspectives. *Journal of Imaging*, 11(8), 1–24. <https://doi.org/10.3390/jimaging11080269>
- Dolz1, J. I. M. · S. B. · D. Q. · M. F. (2025). Deep learning inference optimisation for IoT: Conv2D-ReLU-BN layer fusion and quantisation. *The Journal of Supercomputing*, The deployment of deep learning models on resource.
- Haq, A. F. Q. (2022). Deteksi Tumor Otak Menggunakan Modification LU-Net Berbasis Citra MRI. In *Universitas Islam Negeri Maulana Malik Ibrahim Malang* (Vol. 5, Issue 8.5.2017).
- Hussain, S. S., Wani, N. A., Kaur, J., Ahmad, N., & Ahmad, S. (2025). Next-Generation Automation in Neuro-Oncology: Advanced Neural Networks for MRI-Based Brain Tumor Segmentation and Classification. *IEEE Access*, 13(March), 41141–41158. <https://doi.org/10.1109/ACCESS.2025.3547796>
- Lin, S. Y., & Lin, C. L. (2024). Brain tumor segmentation using U-Net in conjunction with EfficientNet. *PeerJ Computer Science*, 10. <https://doi.org/10.7717/peerj-cs.1754>
- Matuszewski, D. J., & Sintorn, I. (2019). Reducing the U-Net size for practical

- scenarios : Virus recognition in electron microscopy images. *Computer Methods and Programs in Biomedicine*, 178, 31–39. <https://doi.org/10.1016/j.cmpb.2019.05.026>
- McFaline-Figueroa, J. R., & Lee, E. Q. (2018). Brain Tumors. *American Journal of Medicine*, 131(8), 874–882. <https://doi.org/10.1016/j.amjmed.2017.12.039>
- Misbullah, A., Mursyida, W., Farsiah, L., & Sukiakhy, K. M. (2024). Analisis Performa Segmentasi Citra MRI Tumor Otak dengan Arsitektur U-Net. *Journal of Informatics, Information System, and Artificial Intelligence*, 2(2), 83–95.
- Pourmahboubi, A., Arsalani Saeed, N., & Tabrizchi, H. (2025). A brain tumor segmentation enhancement in MRI images using U-Net and transfer learning. *BMC Medical Imaging*, 25(1). <https://doi.org/10.1186/s12880-025-01837-4>
- Preetha, R., M, J. P. P., & Nisha, J. S. (2025). Brain tumor segmentation using multi-scale attention U-Net with EfficientNetB4 encoder for enhanced MRI analysis. *Scientific Reports*, 1–20.
- Ramadhan, F., & Hernadi, J. (2025). Evaluasi Optimizer Adam dan RMSPROP pada Arsitektur VGG-19 Klasifikasi Ekspresi Wajah Manusia. *Jurnal Ilmiah Penelitian Dan Pembelajaran Informatikaa*, 10(2), 1414–1426. <https://doi.org/10.29100/jipi.v10i2.6197>
- Ronneberger, O., Fischer, P., & Brox, T. (2015). U-Net: Convolutional Networks for Biomedical Image Segmentation. *University of Freiburg*, 1–8.
- Sangui, S., Iqbal, T., Chandra, P. C., Ghosh, S. K., & Ghosh, A. (2022). 3D MRI Segmentation using U-Net Architecture for the detection of Brain Tumor. *Procedia Computer Science*, 218, 542–553. <https://doi.org/10.1016/j.procs.2023.01.036>
- Santoso, I. B. (2024). Multi-Model of Convolutional Neural Networks for Brain Tumor Classification in Magnetic Resonance Imaging Images. *International Journal of Intelligent Engineering & Systems*, 17(5), 741–758. <https://doi.org/10.22266/ijies2024.1031.56>
- Sasongko, A., Maulana, M. S., Mustopa, A., & Nugraha, W. (2024). Automatic Wound Image Segmentation with U-Net Model for Smartphone Application. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 10(2), 267. <https://doi.org/10.26418/jp.v10i2.78548>
- Sharma, S., Sharma, S., & Athaiya, A. (2020). Activation Functions in Neural Networks. *International Journal of Engineering Applied Sciences and Technology*, 04(12), 310–316. <https://doi.org/10.33564/ijeast.2020.v04i12.054>
- Soydaner, D. (2020). A Comparison of Optimization Algorithms for Deep Learning. *International Journal of Pattern Recognition and Artificial Intelligence*, 34(13), 1–26. <https://doi.org/10.1142/S0218001420520138>

- Sung, H., Ferlay, J., Siegel, R. L., Laversanne, M., Soerjomataram, I., Jemal, A., & Bray, F. (2021). Global Cancer Statistics 2020: GLOBOCAN Estimates of Incidence and Mortality Worldwide for 36 Cancers in 185 Countries. *CA: A Cancer Journal for Clinicians*, 71(3), 209–249. <https://doi.org/10.3322/caac.21660>
- Suta, I. B. L. M., Sudarma, M., & Satya Kumara, I. N. (2020). Segmentasi Tumor Otak Berdasarkan Citra Magnetic Resonance Imaging Dengan Menggunakan Metode U-NET. *Majalah Ilmiah Teknologi Elektro*, 19(2), 151. <https://doi.org/10.24843/mite.2020.v19i02.p05>
- Ubab, A. J. (2021). *Tafsir Surat Al-Baqarah ayat 195*. NuOnline. <https://nu.or.id/tafsir/tafsir-surat-al-baqarah-ayat-195-maksud-menginfakkan-harta-di-jalan-allah-Fr9Ss>
- Ubab, A. J. (2022). *Tafsir Surat Al-Baqarah ayat 153*. NuOnline. <https://nu.or.id/tafsir/tafsir-surat-al-baqarah-ayat-153-kesabaran-paling-utama-adalah-kesabaran-meninggalkan-maksiat-3FSYI>
- Walsh, J., Othmani, A., Jain, M., & Dev, S. (2022). Using U-Net network for efficient brain tumor segmentation in MRI images. *Healthcare Analytics*, 2(June), 100098. <https://doi.org/10.1016/j.health.2022.100098>
- Wardhani, R., Nafi'iyah, N., & M, K. F. (2023). U-Net Analysis Architecture For MRI Brain Tumor Segmentation. *Jurnal Teknologi Informasi Dan Pendidikan*, 16(September), 127–138.
- Zhang, Q., Yu, H., Zhong, J., Cheng, W., & Qi, Y. (2025). Global, regional, and national burden of brain and central nervous system cancer: a systematic analysis of incidence, deaths, and DALYS with predictions to 2040. *International Journal of Surgery (London, England)*, 111(6), 4033–4038. <https://doi.org/10.1097/JS9.0000000000002359>