

ATRU-NET : ATTENTION TRANSFORMER U-Net UNTUK SEGMENTASI TUMOR OTAK BERDASARKAN CITRA MRI

Krisna Bayu Dharma Putra¹, Rabbani Nur Kumoro², Vincent Yeezekiel³, Dzikri Rahadian Fudholi⁴

¹Fakultas Matematika dan Ilmu Pengetahuan Alam dan Universitas Gadjah Mada, Daerah Istimewa Yogyakarta, 55281, email: krisnabayu01@mail.ugm.ac.id

²Fakultas Matematika dan Ilmu Pengetahuan Alam dan Universitas Gadjah Mada, Daerah Istimewa Yogyakarta, 55281, email: rabbani.nur.kumoro@mail.ugm.ac.id

³Fakultas Matematika dan Ilmu Pengetahuan Alam dan Universitas Gadjah Mada, Daerah Istimewa Yogyakarta, 55281, email: vincent.yeezekiel@mail.ugm.ac.id

⁴Fakultas Matematika dan Ilmu Pengetahuan Alam dan Universitas Gadjah Mada, Daerah Istimewa Yogyakarta, 55281, email: dzikri.r.f@ugm.ac.id

Corresponding Author: Krisna Bayu Dharma Putra

INTISARI — Otak merupakan organ yang penting pada tubuh manusia. Salah satu fungsi penting otak adalah sebagai pengatur sistem saraf. Kemampuan otak dapat menurun karena adanya kelainan, seperti tumor otak, sehingga diperlukan diagnosis otak untuk membantu penanganan. Diagnosis penyakit tumor menjadi tantangan dalam dunia medis. Namun, saat ini terdapat alat untuk membantu deteksi tumor, yaitu *Magnetic Resonance Imaging* (MRI). Analisis citra hasil MRI merupakan pekerjaan yang rumit dan menyita waktu sehingga diperlukan sebuah program komputer yang dapat menganalisis citra tersebut untuk mendeteksi lokasi tumor otak secara otomatis. Metode *Convolutional Neural Network* (CNN) dapat digunakan untuk menganalisis citra MRI. Salah satu arsitektur CNN yang digunakan untuk analisis dan segmentasi citra medis adalah arsitektur U-Net. Meskipun demikian, arsitektur U-Net memiliki kelemahan, yaitu tidak mampu menangkap hubungan antar ciri, terutama pada ciri berjarak jauh, serta hilangnya informasi ciri citra pada tahap enkoder. Penelitian ini mengatasi kekurangan tersebut dengan menambahkan *Feature Pyramid Network* (FPN), *global spatial attention*, dan *attention Transformer*. Metode yang diusulkan memberikan kinerja *Intersection over Union* (IoU) sebesar 0,8292 dan Koefisien *Dice* sebesar 0,8827. Metode ini mampu mengungguli arsitektur U-Net sebesar 13,51% untuk Koefisien *Dice* dan 28,02% pada IoU. Hasil ini juga berhasil mengungguli penelitian sebelumnya, seperti LinkNet, Morphological Geodesic Active Contour, Active Contour dan Fuzzy-C-Means, Convolutional Autoencoder, Multi-Scale CNN, R-CNN, MAG-Net, dan Multimodal Deep Transfer Learning dengan rentang kenaikan skor Koefisien *Dice* sebesar 2,60% hingga 32,73% serta kenaikan IoU sebesar 3,98% hingga 38,20%. Dengan hasil ini, dapat disimpulkan bahwa penambahan FPN, *global spatial attention*, dan *attention Transformers* berhasil meningkatkan performa model dengan cara mengurangi informasi citra yang hilang pada tahap enkoder U-Net serta meningkatkan informasi semantik citra berdependensi jarak jauh.

KATA KUNCI — *Deep Learning, Feature Pyramid Network, Global Spatial Attention, Convolutional Neural Network.*

I. PENDAHULUAN

Tumor otak merupakan kondisi yang dapat memengaruhi fungsi otak secara signifikan dan membutuhkan perhatian serius. Deteksi tumor otak sering kali dilakukan menggunakan *Magnetic Resonance Imaging* (MRI), namun segmentasi tumor secara manual pada hasil MRI adalah proses yang rumit dan memakan waktu [1]. Segmentasi tumor yang akurat sangat penting untuk memberikan diagnosis yang tepat dan mendukung penanganan yang efektif.

Penggunaan MRI dalam segmentasi tumor memberikan banyak manfaat, terutama dengan penerapan metode *Deep Learning* yang dapat meningkatkan akurasi dan efisiensi. Salah satu pendekatan yang efektif adalah penggunaan *Convolutional Neural Networks* (CNN), yang telah terbukti mampu mengidentifikasi dan mengklasifikasikan tumor otak dengan lebih baik dibandingkan metode konvensional [2].

Model U-Net merupakan salah satu model CNN yang telah banyak digunakan dalam segmentasi citra medis, termasuk MRI. U-Net memiliki arsitektur enkoder-dekoder yang dirancang untuk secara akurat mengklasifikasikan piksel gambar, sehingga memungkinkan segmentasi yang lebih baik [3]. Namun, meskipun U-Net efektif, model ini masih memiliki kelemahan dalam menangkap ciri citra dengan dependensi berjarak jauh serta banyaknya informasi yang hilang pada tahap enkoder.

Penelitian ini bertujuan untuk mengintegrasikan mekanisme *Attention Transformer* [4], *Global Spatial Attention* [5], dan

Feature Pyramid Network [6] pada U-Net, agar dapat mengatasi kelemahan tersebut. Model diharapkan dapat menangkap ciri citra dengan lebih baik, termasuk yang memiliki relevansi jarak jauh, serta meminimalisasi informasi ciri yang hilang sehingga segmentasi tumor otak pada MRI dapat dilakukan dengan lebih akurat.

II. METODE PENELITIAN

A. AKUISISI DATA

Dataset yang digunakan adalah *dataset open source* dari Figshare. *Dataset* Figshare merupakan kumpulan gambar MRI otak yang telah diberikan peningkatan kontras berbobot T1 dari Rumah Sakit Nanfang dan Rumah Sakit Umum Tianjin Medical University, dari tahun 2005 hingga 2010, yang terdiri dari berbagai aspek tumor otak dari berbagai sudut pandang, yaitu *sagittal*, *coronal*, dan *axial* [7]. *Dataset* terbagi menjadi tiga buah klasifikasi tumor otak, yaitu Glioma, Meningioma, dan Pituitary dengan jumlah secara berurutan adalah 1435, 708, dan 930.

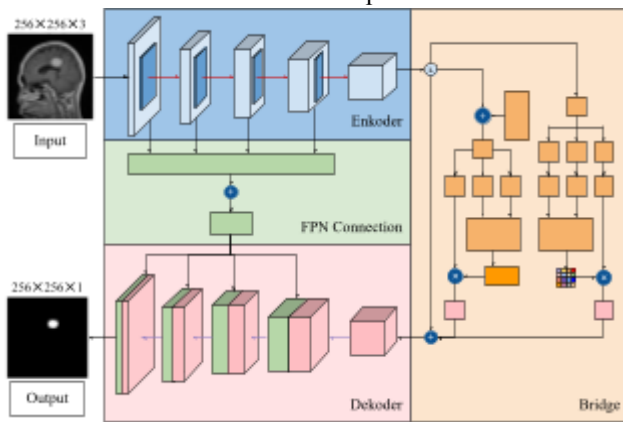
B. PRA-PEMROSESAN DATA

Pra-pemrosesan dilakukan melalui tiga buah tahap. Tahap pertama dilakukan dengan mengubah ukuran citra menjadi 256 × 256. Setelah itu, dilakukan konversi citra menjadi RGB. Pra-pemrosesan pada *masking* dilakukan dengan mengubah

citra menjadi *grayscale* dengan ukuran $256 \times 256 \times 1$. Hal ini dilakukan agar input dan *output* citra memiliki dimensi yang sama dan konsisten. Tahap terakhir adalah mengubah citra menjadi *tensor* agar dapat dimuat ke dalam *data loader*.

C. PEMODELAN ATRU-NET

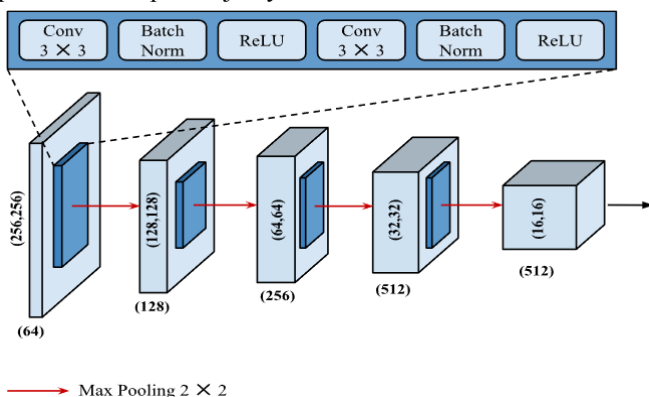
ATRU-Net memiliki tiga buah bagian, yaitu enkoder, dekoder, dan *bridge*. Modifikasi pada U-Net dilakukan dengan menambahkan *attention* ganda menggunakan *multi-head attention* yang berguna untuk menangkap ciri citra global, termasuk yang berjarak jauh, berdasarkan relevansinya. Modifikasi lanjutan dilakukan dengan menambahkan blok *feature pyramid network* (FPN) untuk mengganti *skip-connection* model dasar U-Net. Hal ini bertujuan untuk mendapatkan ekstraksi ciri yang lebih detail serta mengembalikan informasi yang hilang pada tahap enkoder sehingga dapat meningkatkan performa model. Arsitektur utama dari ATRU-Net diilustrasikan pada Gambar 1.



Gambar 1. Arsitektur Attention Transformer U-Net.

1) Enkoder

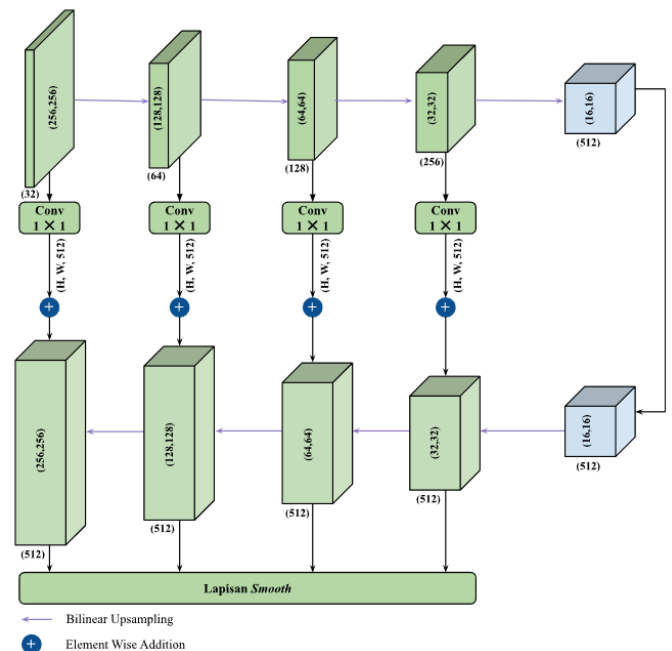
Lapisan Enkoder terdiri dari lima lapisan *downsampling*. Seperti yang digambarkan pada Gambar 2, setiap lapisan *downsampling* memiliki lapisan konvolusi ganda yang diikuti oleh lapisan *max-pooling*. Lapisan enkoder diawali dengan input citra berukuran $256 \times 256 \times 3$. Input citra ini masuk ke dalam layer konvolusi ganda yang memiliki formasi dua kali konvolusi 3×3 diikuti *batch normalization* dengan fungsi aktivasi ReLU. Selanjutnya, output dari setiap tahap downsampling dimasukkan ke dalam lapisan *max-pooling* 2×2 yang mengurangi dimensi citra menjadi 50% dimensi sebelumnya. *Output* dari tiap-tiap *downsampling* ini menjadi input untuk tahap selanjutnya.



Gambar 2. Lapisan Enkoder

2) Lapisan Feature Pyramid Network connection

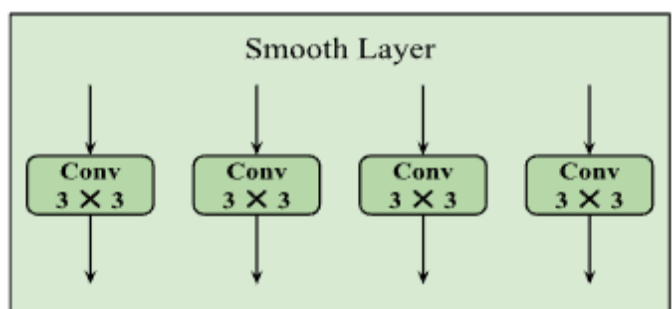
Lapisan *Feature Pyramid Network* (FPN) *connection* terdiri dari beberapa kali *upsampling* dan konvolusi 1×1 . Seperti yang tertera pada Gambar 3, hasil akhir metode *downsampling* telah disalin sehingga mendapatkan lapisan *upsampling*. Lapisan *downsampling* berguna untuk mengekstraksi ciri citra mulai dari ciri tingkat rendah hingga tingkat tinggi. Lapisan *upsampling* berfungsi untuk menangkap informasi dari lapisan *downsampling* melalui metode *element-wise addition* untuk menggabungkan informasi ciri tingkat rendah dan tingkat tinggi. Penggabungan informasi ini ditujukan untuk memperkaya informasi ciri citra sehingga representasi ciri citra akan lebih baik. Hasil akhir dari lapisan *upsampling* kemudian dimasukkan ke dalam lapisan *smooth* untuk dilakukan *smoothing* agar dapat ditambahkan ke dalam lapisan dekoder.



Gambar 3. Lapisan FPN Connection.

3) Lapisan Smooth

Lapisan *smooth* berisi 4 buah lapisan konvolusi 3×3 dengan *output channel* yang berbeda-beda seperti pada Gambar 4. Lapisan ini bertujuan untuk menghaluskan *output* dari lapisan FPN *connection* dan menyamakan ukuran *channel* dari setiap residual agar sesuai dengan lapisan dekodernya. Hal ini bertujuan agar output dari lapisan ini dapat ditambahkan pada lapisan dekoder menggunakan *element-wise addition* sehingga informasi semantik pada lapisan dekoder akan semakin kaya.

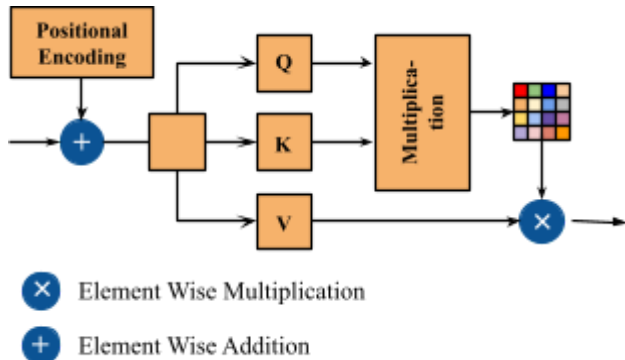


Gambar 4. Lapisan Smooth

4) Self-Attention Transformer

Self-Attention Transformer terdiri dari gabungan matriks Q, K, dan V seperti pada Gambar 5. Lapisan *Self-Attention*

Transformer terdiri dari *multi-head attention* yang membuat model mampu untuk mengenali informasi semantik dari ciri global serta memahami relevansi antar ciri. *Positional encoding* pada lapisan ini memiliki fungsi untuk menangkap informasi posisi absolut dan relatif dari ciri citra.



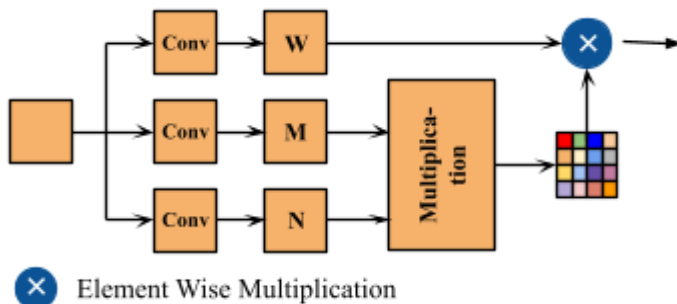
Gambar 5. Lapisan Self-Attention. Transformers

Input ciri pertama kali digabungkan dengan *positional encoding* melalui *element-wise addition*, kemudian matriks *query* Q, matriks *key* K, dan matriks *value* V dibentuk. Matriks Q kemudian dikalikan melalui matriks *multiplication* dengan *transpose* K. Outputnya kemudian dibagi dengan akar dari ukuran *embedding* yang kemudian dimasukkan dalam fungsi aktivasi *softmax*. Selanjutnya, *output* dari matriks ini dikalikan dengan V melalui perkalian matriks untuk mendapatkan output yang terakhir. Secara umum, lapisan *attention* merupakan hasil dari perkalian matriks hasil *softmax* Q dan *transpose* K yang dibagi dengan suatu nilai *temperature* dengan V seperti persamaan 1. Secara keseluruhan, lapisan ini berfungsi untuk peningkatan representasi ciri berjarak jauh.

$$Attention(Q, K, V) = softmax(\frac{QK^T}{temperature})V \quad (1)$$

5) Lapisan Global Spatial Attention

Lapisan *Global Spatial Attention* (GSA) berfungsi untuk mengagregasi ciri global ke dalam ciri yang sudah dipelajari dan mengencode informasi posisi ke dalam ciri lokal. Hal ini berguna untuk meningkatkan kepadatan antarkelas dan mengoptimisasi representasi ciri. Secara umum, GSA terdiri dari gabungan beberapa lapisan konvolusi seperti pada Gambar 6

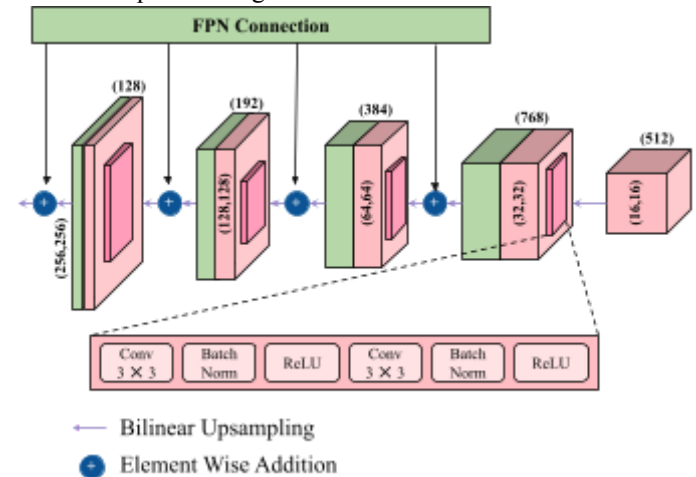


Gambar 6. Lapisan Global Spatial Attention

6) Dekoder

Lapisan dekoder terdiri dari beberapa lapisan *upsampling* menggunakan *bilinier upsampling*. Input ciri pada setiap lapisan diproses menggunakan konvolusi ganda yang terdiri dari kombinasi dua kali konvolusi 3×3 , *batch normalization*, dan fungsi aktivasi ReLU. *Output* setiap formasi ini

di-*upsampling* dan diikuti lapisan formasi konvolusi ganda sampai ekstraksi ciri memiliki dimensi sama dengan dimensi awalnya. Setiap lapisan *upsampling* kemudian ditambahkan melalui metode *concatenate* dan *element-wise addition* dari *smooth layer* dan koneksi FPN seperti yang divisualisasikan pada Gambar 7. Hal ini bertujuan untuk memperkaya informasi ciri dari ekstraksi ciri serta mengembalikan informasi ciri yang hilang saat proses encoding. Lapisan terakhir, lapisan konvolusi 1×1 , digunakan untuk mendapatkan *feature map* terakhir. *Feature map* inilah yang kemudian digunakan untuk melakukan prediksi segmentasi citra.



Gambar 7. Lapisan Dekoder

D. METRIK PENGUJIAN

Hasil segmentasi model kemudian dievaluasi tingkat keakuratan dan keefektifan model dalam mensegmentasi tumor otak. Metrik yang digunakan adalah *Intersection over Union* (IoU) dan Koefisien *Dice*. Kedua metrik tersebut memiliki skala antara 0 hingga 1. Nilai yang tinggi berarti model semakin akurat dan tepat memprediksi wilayah. IoU mengukur kemiripan antara *mask* yang diprediksi dan *ground truth* dengan menghitung rasio perpotongan dengan gabungannya. Disisi lain, metrik Koefisien *Dice* dihitung sebagai dua kali luas area tumpang tindih dibagi dengan jumlah luas area yang diprediksi dan area yang sebenarnya.

A. DATASET

Dataset yang digunakan berjumlah 3064 yang terbagi menjadi data latih, data validasi, dan data uji. *Dataset* dibagi dengan rasio 80:10:10 dengan pembagian 80% untuk data latih, 10% data validasi, dan 10% data uji. Pembagian ini memungkinkan penilaian kinerja model yang dapat diandalkan di berbagai kategori tumor, memungkinkan deteksi *overfitting* dan penyesuaian yang diperlukan untuk meningkatkan generalisasi.

B. SITUASI PERCOBAAN

TABEL I
PARAMETER YANG DIUJI

Hyperparameter	Nilai
<i>Epochs</i>	20 (Diambil hasil yang terbaik)
<i>Batch Size</i>	8
<i>Optimizer</i>	Adam, AdamW, RMSprop, AdaGrad
<i>Learning Rate</i>	0,001, 0,0001
Fungsi Aktivasi	ReLU, GELU

Fungsi Loss	Dice Loss
-------------	-----------

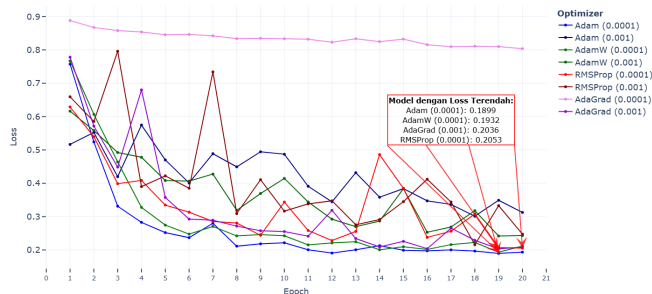
Percobaan dilakukan dengan menggunakan *website* Kaggle, GPU-P100, dan bahasa pemrograman Python. *Framework* yang digunakan pada penelitian ini adalah PyTorch. Selanjutnya, percobaan dilakukan dengan menggunakan 20 epochs, 8 *batch size*, beberapa *optimizer*, *learning rate* 0,001 dan 0,0001, fungsi aktivasi ReLU dan GELU [8], serta fungsi *loss* yaitu Dice *loss* seperti pada Tabel I.

C. PERCOBAAN

1) Learning Rate

Percobaan pertama dilakukan dengan membandingkan *learning rate* pada *optimizer* Adam, AdamW, AdaGrad, dan RMSprop. Percobaan dilakukan dengan menggunakan 20 epoch, fungsi aktivasi ReLU, *batch size* 8, serta *learning rate* 0,001 dan 0,0001.

Percobaan *learning rate* 0,001 mengalami fluktuasi nilai *loss* untuk ke-empat *optimizer* seperti pada Gambar 8. Hasil percobaan menunjukkan bahwa *learning rate* 0,001 menghasilkan grafik yang sangat fluktuatif, terutama untuk *optimizer* Adam, AdamW, dan RMSprop. Pelatihan pada *optimizer* RMSprop menunjukkan fluktuasi yang sangat tinggi dari *epoch* kelima menuju *epoch* ke-6. Hal ini menunjukkan bahwa model membutuhkan waktu yang lama untuk mencapai tingkat konvergensi dengan *learning rate* sebesar 0,001. Di lain sisi, pada *optimizer* AdaGrad dapat dilihat bahwa *validation loss* mengalami fluktuasi yang lebih sedikit dibanding tiga *optimizer* lainnya, namun fluktuasi yang terjadi tetaplah cukup besar. AdaGrad juga berhasil mencapai nilai *validation loss* paling kecil jika dibandingkan dengan tiga *optimizer* lain.

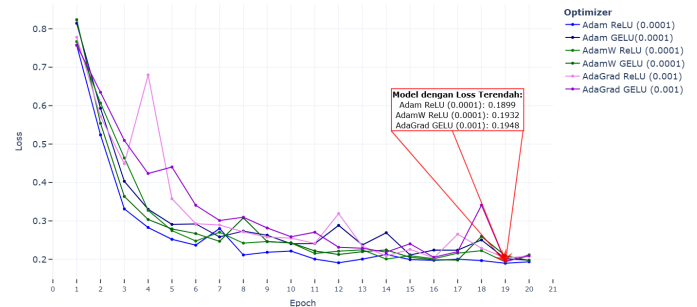


Gambar 8. Pengujian Loss terhadap semua Learning Rate.

Analisis lanjutan dilakukan pada keempat *optimizer* dengan *learning rate* 0,0001. Hasil percobaan menunjukkan bahwa AdaGrad memiliki nilai *validation loss* yang sangat besar pada *learning rate* ini. Hal ini menunjukkan bahwa AdaGrad kurang optimal pada *learning rate* sebesar 0,0001 dan lebih optimal pada *learning rate* 0,001. Perbandingan signifikan justru terlihat di ketiga *optimizer* lain. Dibandingkan dengan *learning rate* 0,001, *learning rate* 0,0001 menunjukkan grafik *validation loss* yang lebih landai dan tidak terlalu fluktuatif, terutama untuk *optimizer* Adam dan AdamW. Kedua *optimizer* ini terbukti memiliki grafik *validation loss* yang cenderung terus turun dengan fluktuasi yang tidak besar yang menunjukkan tingkat pembelajaran model yang cukup baik pada nilai *learning rate* 0,0001. Evaluasi lebih lanjut menunjukkan bahwa *optimizer* dengan performa terbaik adalah Adam dengan nilai *validation loss* terkecil sebesar 0,1899, disusul oleh AdamW dengan nilai *validation loss* terkecil sebesar 0,1932. Oleh karena itu, untuk percobaan selanjutnya, *learning rate* 0,0001 dengan *optimizer* Adam dan AdamW serta AdaGrad dengan *learning rate* 0,001.

2) Fungsi Aktivasi

Setelah didapatkan nilai *learning rate* paling optimal 0,0001 dengan *optimizer* Adam dan AdamW, serta AdaGrad dengan *learning rate* 0,001. Percobaan selanjutnya mencoba membandingkan pengaruh fungsi aktivasi pada ketiga jenis *optimizer*. Penelitian ini mencoba membandingkan pengaruh dua buah jenis fungsi aktivasi, ReLU dan GELU terhadap performa akhir model.



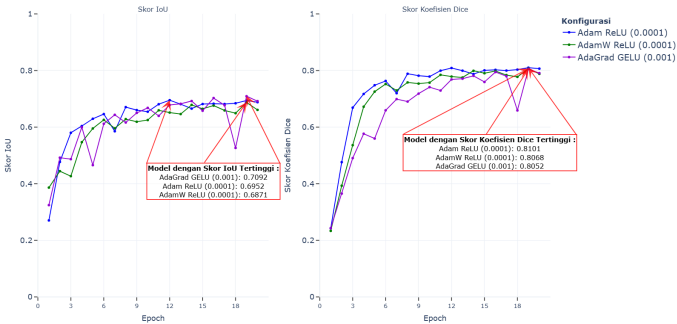
Gambar 9. Pengujian Loss terhadap Fungsi Aktivasi.

Pengujian *learning rate* dengan nilai 0.0001 pada fungsi aktivasi ReLU dan GELU mengalami fluktuasi yang cukup seragam seperti pada Gambar 9. Hasil pengujian menunjukkan bahwa *optimizer* Adam dan AdamW tidak terlalu optimal pada fungsi aktivasi GELU. Hal ini ditunjukkan grafik *loss* pada fungsi ini yang justru lebih tinggi dibandingkan saat menggunakan fungsi aktivasi ReLU. Hal sebaliknya justru ditunjukkan oleh *optimizer* AdaGrad yang menunjukkan fungsi *loss* lebih rendah saat menggunakan fungsi aktivasi GELU. Fungsi aktivasi GELU pada *optimizer* AdaGrad juga terbukti berhasil mengurangi fluktuasi *loss*. Hal ini dapat dilihat pada grafik yang menunjukkan fluktuasi AdaGrad GELU yang lebih rendah dibandingkan AdaGrad ReLU. Hal inilah yang mendasari peneliti untuk menggunakan *optimizer* Adam dan AdamW dengan fungsi ReLU dan AdaGrad dengan fungsi GELU untuk eksperimen selanjutnya.

3) Perbandingan skor IoU dan Koefisien Dice

Percobaan selanjutnya adalah mencoba membandingkan nilai IoU dan Koefisien Dice pada ketiga jenis *optimizer*. Hal ini bertujuan untuk memastikan bahwa model yang telah dilatih mampu mendapatkan skor tinggi tidak hanya pada satu buah metrik, namun juga memiliki nilai yang baik pada metrik lain, seperti IoU.

Hasil eksperimen menunjukkan bahwa nilai IoU untuk ketiga jenis *optimizer* mengalami fluktuasi naik turun selama proses pelatihan sesuai Gambar 10. Fluktuasi terburuk dialami oleh AdaGrad pada sekitaran epoch ke-16, namun segera naik kembali di *epoch* selanjutnya. Grafik *optimizer* Adam dan AdamW terlihat lebih stabil jika dibandingkan dengan AdaGrad. Namun, pada akhir pelatihan AdaGrad berhasil mendapatkan skor IoU tertinggi dibandingkan dua *optimizer* lain. Walaupun begitu, skor IoU akhir AdaGrad dan Adam tidak terlampau jauh, AdaGrad mendapatkan skor IoU tertinggi sebesar 0,7092, sedangkan Adam mendapatkan skor IoU tertinggi sebesar 0,6952. Di lain sisi AdamW mengalami sedikit ketertinggalan dengan hanya skor IoU sebesar 0,6871 pada skor evaluasi IoU.



Gambar 10. Perbandingan Skor IoU dan Koefisien Dice pada Data Validasi.

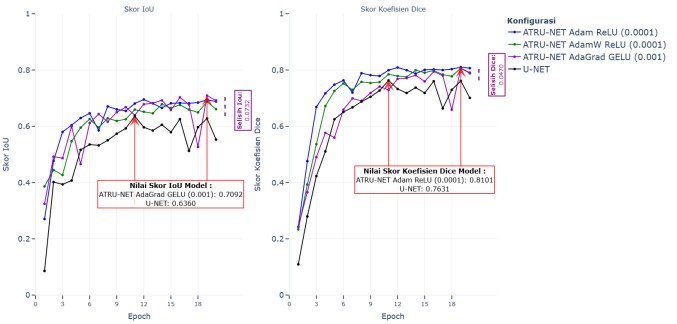
Selanjutnya, evaluasi juga dilakukan pada Koefisien Dice. Hasil pengujian pada data validasi dengan 20 epochs. Pengujian pada data validasi menunjukkan bahwa ketiga *optimizer* mendapatkan skor Koefisien Dice yang cukup mirip seperti pada Gambar 10. *Optimizer* AdaGrad menunjukkan skor yang cukup fluktuatif dan memiliki nilai Koefisien Dice terendah pada epoch satu sampai ke-15, lalu mengalami fluktuasi pada epochs ke-16 dan ke-17 yang selanjutnya mengalami kenaikan kembali hingga cukup menyamai skor *optimizer* AdamW pada epoch ke-18 sampai ke-20. Skor Koefisien Dice tertinggi AdaGrad ada pada epoch ke-19 dengan nilai 0,8052.

Optimizer AdamW memiliki peningkatan tertinggi kedua selama pelatihan pada epoch pertama sampai epoch ke-15. *Optimizer* ini juga memiliki fluktuasi skor yang cukup landai dibandingkan dengan AdaGrad yang menunjukkan stabilitas yang cukup tinggi pada *optimizer* ini. Skor Koefisien Dice tertinggi diraih pada epochs ke-19 dengan skor 0,8068.

Optimizer yang memiliki sejarah pelatihan paling baik adalah ATRU-Net dengan *optimizer* berupa Adam. *Optimizer* ini berhasil mengungguli kedua *optimizer* lain selama pelatihan, meskipun sempat mengalami fluktuasi pada epoch ke-7 dan ke-12. Hasil akhir *optimizer* ini berhasil mendapatkan skor Koefisien Dice tertinggi sebesar 0,8101 pada epoch ke-19. Skor ini berhasil mengungguli ATRU-Net dengan *optimizer* AdamW dan AdaGrad pada skor Koefisien Dice.

4) Perbandingan Model ATRU-Net dengan U-Net

Pengujian terakhir dilakukan dengan melakukan perbandingan metrik evaluasi dengan arsitektur dasar U-Net. Arsitektur dasar U-Net dilatih dengan kondisi *learning rate* sebesar 0,0001, fungsi aktivasi ReLU, *batch size* 8, serta total epochs 20. Hasil percobaan menunjukkan bahwa arsitektur usulan, yaitu ATRU-Net berhasil mengungguli arsitektur dasar U-Net untuk semua variasi di kedua metrik evaluasi seperti pada Gambar 11. Arsitektur ATRU-Net memiliki nilai validation loss yang lebih kecil jika dibandingkan arsitektur U-Net, terutama untuk variasi *optimizer* Adam dan AdaGrad yang menunjukkan performa yang sangat baik. Evaluasi pada skor IoU menunjukkan ATRU-Net dengan variasi Adam 0,0001 menghasilkan skor tertinggi sebesar 0,7092 dan mengungguli U-Net pada skor evaluasi yang sama dengan selisih skor sebesar 0,0732 atau sekitar 7,3%. Evaluasi pada skor Koefisien Dice menunjukkan AdaGrad 0,001 adalah konfigurasi terbaik dengan skor Koefisien Dice sebesar 0,8101. Skor ini mengungguli skor Koefisien Dice U-Net dengan selisih sebesar 0,0470 atau sekitar 4,7%.



Gambar 11. Perbandingan Skor IoU dan Koefisien Dice Loss.

5) Perbandingan Ekstraksi ATRU-Net dengan U-Net

Salah satu keunggulan ATRU-Net dibandingkan U-Net adalah kemampuannya dalam melakukan ekstraksi ciri yang lebih baik seperti pada Gambar 12. Berbeda dengan U-Net yang hanya memiliki empat buah tahap, ATRU-Net memiliki sekitar delapan buah tahap ekstraksi yang masing-masing tahap memiliki fokus yang berbeda. Sebagai contoh, FPN mengekstraksi ciri citra secara global, sedangkan GSA dan Transformer mengekstraksi ciri lokal. Selanjutnya, hasil kedua ekstraksi ciri digabungkan sehingga menghasilkan ciri citra akhir yang kaya. ATRU-Net berhasil mendapatkan skor IoU 0,7776 dan Koefisien Dice 0,8363. Dengan hasil ini, ekstraksi ciri yang lebih detail dari ATRU-Net berhasil mengungguli model U-Net sebesar 0,3348 untuk skor IoU dan 0,2220 untuk skor Koefisien Dice.

6) Perbandingan ATRU-Net dengan U-Net pada Data Uji

Evaluasi terakhir dilakukan dengan menggunakan data uji sebesar 307 citra. Data uji merupakan suatu data yang belum pernah terlihat dan belum pernah digunakan saat pelatihan. Evaluasi dilakukan pada jenis arsitektur ATRU-Net Adam, AdamW, AdaGrad, dan U-Net. Hasil pengujian menunjukkan bahwa ATRU-Net berhasil mengungguli U-Net untuk semua varian pada data uji untuk semua metrik evaluasi dengan arsitektur ATRU-Net. Varian ATRU-Net terbaik diraih oleh ATRU-Net dengan konfigurasi *optimizer* AdaGrad, fungsi aktivasi GELU, dan *learning rate* 0.001 yang berhasil meningkatkan skor IoU sebesar 28,02% serta Koefisien Dice sebesar 13,52% seperti yang terlihat pada Tabel II.

TABEL II
PERBANDINGAN HASIL ANTARA MODEL

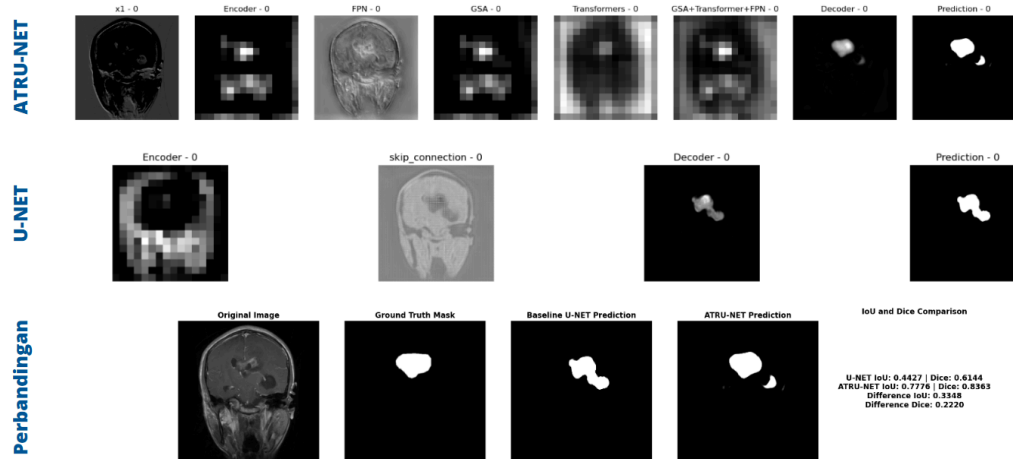
Arsitektur	Nilai Loss	Koefisien Dice	Nilai IoU	Total Parameter
U-Net	0,2223	0,7777	0,6477	31.037.633
ATRU-Net Adam 0,0001 ReLU	0,1524	0,8476	0,7463	30.883.978
ATRU-Net AdamW 0,0001 ReLU	0,1384	0,8616	0,7660	30.883.978
ATRU-Net AdaGrad 0,001 GELU	0,1173	0,8827	0,8292	30.883.978

7) Perbandingan ATRU-Net dengan Penelitian Sebelumnya

Perbandingan antara ATRU-Net dengan penelitian-penelitian sebelumnya dapat dilihat pada Tabel III. Tabel ini menunjukkan bahwa ATRU-Net berhasil mengungguli 8 penelitian lainnya dengan rentang 2,60%

hingga 32,73% pada skor Koefisien *Dice* serta 3,98% hingga 38,20% untuk skor IoU. Hasil ini menunjukkan bahwa penambahan lapisan FPN, *global spatial attention*, dan

attention Transformers pada lapisan U-Net berhasil meningkatkan performa model dalam kasus segmentasi tumor otak.



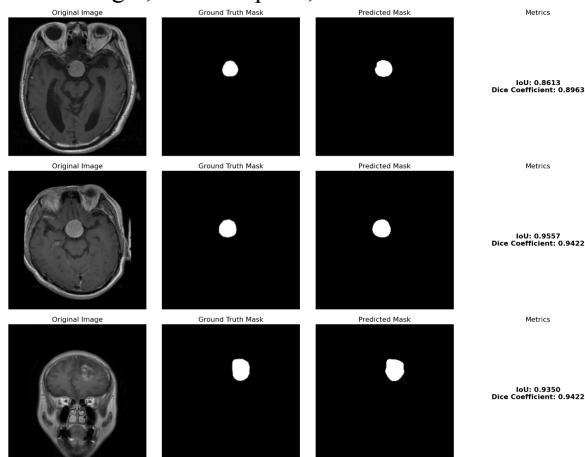
Gambar 12. Perbedaan Ekstraksi Ciri ATRU-Net dengan U-Net.

TABEL III
PERBANDINGAN HASIL ANTARA MODEL

Arsitektur	Koefisien <i>Dice</i>	Nilai IoU
Separate LinkNet [9]	0,7600	-
Morphological Geodesic Active Contour [10]	-	0,7974
Active Contour dan Fuzzy C-Means [11]	0,6650	-
Convolutional Autoencoder [12]	0,7287	-
Multi-Scale CNN [13]	0,8280	-
R-CNN [1]	0,8700	-
MAG-Net [14]	0,7400	0,600
Multimodal Deep Transfer Learning [15]	0,8603	-
U-Net [3]	0,7777	0,6477
ATRU-Net	0,8827	0,8292

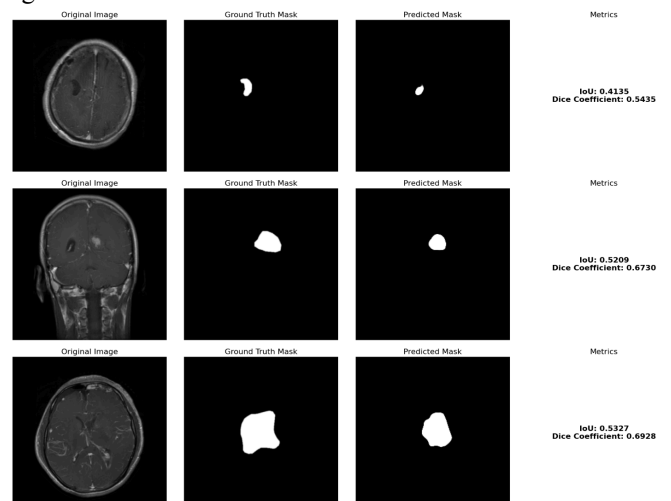
8) Pengujian ATRU-Net secara Visual

Pengujian visualisasi ATRU-Net dilakukan dengan melakukan prediksi segmentasi pada tiga buah citra dari dataset uji seperti pada Gambar 13. Hasil analisis menunjukkan bahwa ATRU-Net berhasil melakukan segmentasi tumor otak berukuran kecil dengan baik. Skor IoU yang dicapai berada di rentang 0,8613 sampai 0,9557 dengan skor Koefisien *Dice* berada di rentang 0,8963 sampai 0,9422.



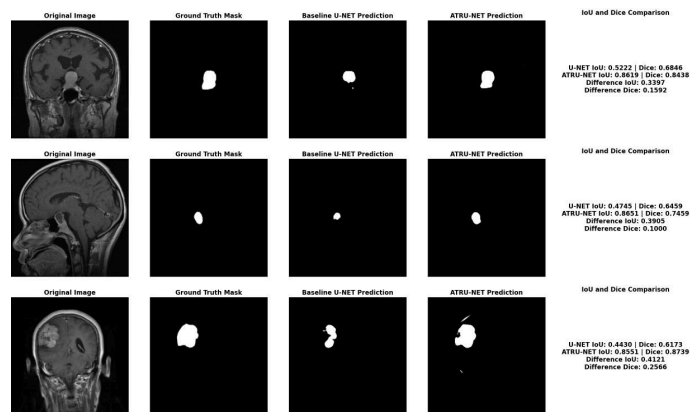
Gambar 13. Pengujian ATRU-Net Pertama.

Pengujian selanjutnya dilakukan dengan menggunakan citra tumor otak berukuran tidak beraturan seperti pada Gambar 14. Hasil pengujian menunjukkan bahwa ATRU-Net mendapatkan skor IoU di rentang 0,4135 sampai 0,5227 serta skor Koefisien *Dice* di rentang 0,5435 sampai 0,6928. Hasil ini mengindikasikan bahwa ATRU-Net memiliki kelemahan dalam segmentasi citra tumor otak berukuran tidak teratur.



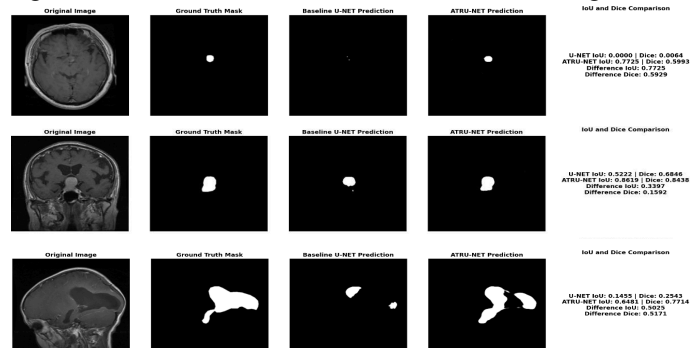
Gambar 14. Pengujian ATRU-Net Kedua

Pengujian lanjutan dilakukan dengan membandingkan performa ATRU-Net dengan U-Net. Pengujian dilakukan dengan menggunakan tiga buah citra data uji. Hasil pengujian menunjukkan bahwa ATRU-Net berhasil mengungguli performa U-Net untuk kedua metrik evaluasi seperti pada Gambar 15. Sebagai contoh, pada pengujian pertama memperlihatkan kelebihan ATRU-Net dalam mensegmentasi tumor otak yang berukuran sedang. ATRU-Net berhasil mendapatkan skor IoU dan Koefisien *Dice* secara berurut sebesar 0,8619 dan 0,8438. Hasil ini berhasil mengungguli performa U-Net sebesar 0,3397 pada skor IoU dan 0,1592 untuk skor Koefisien *Dice*.



Gambar 15. Perbandingan Hasil Uji ATRU-Net dengan U-Net.

Pengujian terakhir dilakukan dengan membandingkan ATRU-Net dengan U-Net pada kasus khusus, yaitu citra berukuran kecil, sedang, dan sangat besar namun tidak beraturan seperti pada Gambar 16. Pengujian pertama dan kedua dilakukan pada sebuah citra dari data uji dengan ukuran kecil dan sedang. Hasil pengujian menunjukkan bahwa U-Net sama sekali tidak mampu mensegmentasi citra berukuran kecil dengan skor IoU dan Koefisien *Dice* mendekati 0 untuk kedua buah citra uji. Di lain sisi, ATRU-Net berhasil mendapatkan skor IoU sebesar 0,7723 dan Koefisien *Dice* sebesar 0,5993. Selanjutnya, pada pengujian kedua, ATRU-Net mendapatkan skor IoU sebesar 0,8619 dan skor Koefisien *Dice* sebesar 0,8438. Kedua hasil ini menunjukkan bahwa ATRU-Net, jika dibandingkan dengan U-Net, memiliki keunggulan pada segmentasi citra tumor otak berukuran kecil dan sedang.



Gambar 16. Perbandingan Hasil Uji Tumor Berukuran Kecil, Sedang, dan Tidak Beraturan ATRU-Net dengan U-Net.

Pada pengujian ketiga, dilakukan uji coba terhadap citra uji berukuran besar dan tidak teratur. Hasil pengujian menunjukkan ATRU-Net berhasil mendapatkan skor IoU sebesar 0,6481 dan Koefisien *Dice* sebesar 0,7714. Hasil ini berhasil mengungguli arsitektur U-Net yang hanya mendapatkan skor IoU 0,1455 dan Koefisien *Dice* 0,2543 dengan peningkatan IoU sebesar 0,5025 serta Koefisien *Dice* sebesar 0,5171.

Hasil observasi secara visual menunjukkan hasil yang menarik. ATRU-Net terbukti berhasil menyempurnakan segmentasi tumor otak U-Net. Penyempurnaan ini dilakukan dengan menambahkan bagian-bagian segmentasi yang sebelumnya tidak mampu dideteksi oleh U-Net. Meskipun hasilnya belum sempurna, ATRU-Net terbukti memiliki potensi untuk dikembangkan lebih lanjut sehingga mampu mendeteksi citra berukuran tidak teratur dengan lebih sempurna.

III. KESIMPULAN

Hasil penelitian menunjukkan adanya peningkatan performa dibandingkan dengan model sebelumnya. Penambahan *attention* Transformers serta *global spatial attention* berhasil meningkatkan representasi citra. Hal ini dibuktikan pada proses ekstraksi fitur ATRU-Net yang berfokus pada ciri citra lokal dan global pada kedua tahap ini. Selain itu, Penambahan *feature pyramid network* menggantikan *skip-connection* U-Net berhasil mengurangi informasi yang hilang pada tahap enkoder dengan menggabungkan representasi ciri tingkat tinggi dengan tingkat rendah.

Keberhasilan ini dibuktikan dengan diraihnya skor IoU sebesar 0,8292 dan Koefisien *Dice* sebesar 0,8292. Hasil ini menunjukkan peningkatan sebesar 28,02% pada skor IoU serta 13,52% pada skor Koefisien *Dice*.

Perbandingan lebih lanjut dilakukan dengan membandingkan ATRU-Net dengan model segmentasi lainnya, seperti LinkNet, *Morphological Geodesic Active Contour*, dan *Multi-Scale CNN*, ATRU-Net menunjukkan peningkatan yang signifikan dalam kinerja, dengan peningkatan skor Koefisien *Dice* berkisar antara 2,60% hingga 32,73% dan peningkatan IoU sebesar 3,98% hingga 38,20%. Hal ini menegaskan keunggulan ATRU-Net sebagai metode yang lebih efektif untuk segmentasi citra medis. Peningkatan yang signifikan ini menunjukkan bahwa metode yang diusulkan berhasil menyelesaikan permasalahan informasi hilang serta tidak adanya representasi citra berdependensi jarak jauh pada arsitektur U-Net.

REFERENSI

- [1] S. Maqsood, R. Damaševičius, dan R. Maskeliūnas, "Multi-Modal Brain Tumor Detection Using Deep Neural Network and Multiclass SVM," *Medicina (Kaunas)*, vol. 58, no. 8, hlm. 1090, Agu 2022, doi: 10.3390/medicina58081090.
- [2] J. Kang, Z. Ullah, dan J. Gwak, "MRI-Based Brain Tumor Classification Using Ensemble of Deep Features and Machine Learning Classifiers," *Sensors*, vol. 21, no. 6, Art. no. 6, Jan 2021, doi: 10.3390/s21062222.
- [3] O. Ronneberger, P. Fischer, dan T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," dalam *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, N. Navab, J. Hornegger, W. M. Wells, dan A. F. Frangi, Ed., Cham: Springer International Publishing, 2015, hlm. 234–241. doi: 10.1007/978-3-319-24574-4_28.
- [4] A. Vaswani *dkk.*, "Attention Is All You Need," 1 Agustus 2023, *arXiv*: arXiv:1706.03762. doi: 10.48550/arXiv.1706.03762.
- [5] L. Xu, J. Huang, A. Nitanda, R. Asaoka, dan K. Yamanishi, "A Novel Global Spatial Attention Mechanism in Convolutional Neural Network for Medical Image Classification," *arXiv.org*. Diakses: 6 September 2024. [Daring]. Tersedia pada: <https://arxiv.org/abs/2007.15897v1>
- [6] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, dan S. Belongie, "Feature Pyramid Networks for Object Detection," 19 April 2017, *arXiv*: arXiv:1612.03144. doi: 10.48550/arXiv.1612.03144.
- [7] J. Cheng, "brain tumor dataset," figshare, 2 April 2017. doi: 10.6084/m9.figshare.1512427.v5.
- [8] D. Hendrycks dan K. Gimpel, "Gaussian Error Linear Units (GELUs)," *arXiv.org*. Diakses: 6 September 2024. [Daring]. Tersedia pada: <https://arxiv.org/abs/1606.08415v5>
- [9] Z. Sobhaninia *dkk.*, "Brain Tumor Segmentation Using Deep Learning by Type Specific Sorting of Images," 20 September 2018, *arXiv*: arXiv:1809.07786. doi: 10.48550/arXiv.1809.07786.
- [10] A. H. Thias, A. F. Al Mubarak, A. Handayani, D. Danurdirdjo, dan T. E. Rajab, "Brain Tumor Semi-automatic Segmentation on MRI T1-weighted Images using Active Contour Models," dalam *2019 International Conference on Mechatronics, Robotics and Systems Engineering (MoRSE)*, Des 2019, hlm. 217–221. doi: 10.1109/MoRSE48060.2019.8998651.
- [11] C. J. J. Sheela dan G. Suganthi, "Brain tumor segmentation with radius contraction and expansion based initial contour detection for active contour model," *Multimedia Tools Appl.*, vol. 79, no. 33–34, hlm. 23793–23819, Sep 2020, doi: 10.1007/s11042-020-09006-1.
- [12] M. M. Badža dan M. Č. Barjaktarović, "Segmentation of Brain Tumors from MRI Images Using Convolutional Autoencoder," *Applied Sciences*, vol. 11, no. 9, Art. no. 9, Jan 2021, doi: 10.3390/app11094317.
- [13] F. J. Díaz-Pernas, M. Martínez-Zarzuela, M. Antón-Rodríguez, dan D. González-Ortega, "A Deep Learning Approach for Brain Tumor

- Classification and Segmentation Using a Multiscale Convolutional Neural Network,” *Healthcare*, vol. 9, no. 2, Art. no. 2, Feb 2021, doi: 10.3390/healthcare9020153.
- [14] S. Gupta, N. S. Pun, S. K. Sonbhadra, dan S. Agarwal, “MAG-Net: Multi-task attention guided network for brain tumor segmentation and classification,” vol. 13147, 2021, hlm. 3–15. doi: 10.1007/978-3-030-93620-4_1.
- [15] P. Razzaghi, K. Abbasi, M. Shirazi, dan S. Rashidi, “Multimodal brain tumor detection using multimodal deep transfer learning,” *Applied Soft Computing*, vol. 129, hlm. 109631, Nov 2022, doi: 10.1016/j.asoc.2022.109631.