



TUGAS AKHIR - EB184803

**SEGMENTASI LESI HYPODENSE PADA CITRA CT SCAN OTAK
MENGUNAKAN CNN ARSITEKTUR U-NET UNTUK DETEKSI
AWAL STROKE ISKEMIA**

Bhetri Sonia Yolandari
0731174000029

DOSEN PEMBIMBING:

Dr. Tri Arief Sardjono, S.T., M.T.
Nada Fitriyatul Hikmah, S.T., M. T.

PROGRAM SARJANA
DEPARTEMEN TEKNIK BIOMEDIK
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2021



TUGAS AKHIR - EB184803

**SEGMENTASI LESI HYPODENSE PADA CITRA CT SCAN OTAK
MENGUNAKAN CNN ARSITEKTUR U-NET UNTUK DETEKSI AWAL
STROKE ISKEMIA**

**Bhetri Sonia Yolandari
0731174000029**

**DOSEN PEMBIMBING:
Dr. Tri Arief Sardjono, S.T., M.T.
Nada Fitriyatul Hikmah, S.T., M.T.**

**PROGRAM SARJANA
DEPARTEMEN TEKNIK BIOMEDIK
FAKULTAS TEKNOLOGI ELEKTRO DAN INFORMATIKA CERDAS
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2021**



FINAL PROJECT - EB184803

**SEGMENTATION OF HYPODENSE LESION IN BRAIN CT SCAN IMAGE
USING CNN U-NET ARCHITECTURE FOR EARLY DETECTION OF
ISCHEMIC STROKE**

**Bhetri Sonia Yolandari
0731174000029**

**SUPERVISOR:
Dr. Tri Arief Sardjono, S.T., M.T.
Nada Fitriyatul Hikmah, S.T., M.T.**

**UNDERGRADUATE PROGRAM
BIOMEDICAL ENGINEERING DEPARTMENT
FACULTY OF INTELLIGENT ELECTRICAL AND INFORMATICS
TECHNOLOGY
SEPULUH NOPEMBER INSTITUTE OF TECHNOLOGY
SURABAYA
2021**

PERNYATAAN KEASLIAN TUGAS AKHIR

Dengan ini saya menyatakan bahwa sebagian maupun keseluruhan isi tugas akhir saya yang berjudul “**Segmentasi Lesi *Hypodense* Pada Citra CT Scan Otak Menggunakan CNN Arsitektur U-Net Untuk Deteksi Awal Stroke Iskemia**” merupakan hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya mandiri.

Semua referensi yang dikutip maupun dirujuk telah ditulis secara lengkap pada daftar pustaka. Apabila ternyata pernyataan ini tidak benar, saya bersedia menerima sanksi sesuai peraturan yang berlaku.

Surabaya, Juli 2021

Bhetri Sonia Yolandari
NRP. 0731174000029

**Tugas Akhir disusun untuk memenuhi salah satu syarat memperoleh gelar
Sarjana Teknik (S.T.)
di
Institut Teknologi Sepuluh Nopember**

**Oleh
Bhetri Sonia Yolandari
NRP. 0731174000029**

**Tanggal Ujian : 07 Juli 2021
Periode Wisuda : September 2021**

Disetujui Oleh :

Dosen Penguji :



**1. Dr. Rachmad Setiawan, S.T., M.T.
NIP. 196905291995121001**



**2. Atar Fuady Babgei, S.T., M.Sc.
NIP. 198911112018121001**



**3. Eko Agus Suprayitno, S.Si., M.T.
NIP. 1987202011026**

Dosen Pembimbing :



**1. Dr. Tri Arief Sardjono, S.T., M.T.
NIP. 197002121995121001**



**2. Nada Fitriyatul Hikmah, S.T., M.T.
NIP. 199001072018032001**

Kepala Departemen Teknik Biomedik,



**Dr. Achmad Arifin, S.T., M.Eng.
NIP. 197103141997021001**

SEGMENTASI LESI HYPODENSE PADA CITRA CT SCAN OTAK MENGUNAKAN CNN ARSITEKTUR U-NET UNTUK DETEKSI AWAL STROKE ISKEMIA

Nama : Bhetri Sonia Yolandari
NRP : 0731174000029
Dosen Pembimbing : 1. Dr. Tri Arief Sardjono, S.T., M.T.
2. Nada Fitriyatul Hikmah, S.T., M.T.

ABSTRAK

Stroke atau cedera *cerebrovascular* merupakan penyakit yang terjadi ketika pembuluh darah yang membawa oksigen dan nutrien ke otak mengalami penyumbatan atau pecah sehingga menyebabkan sel otak mati. Sebanyak 87% dari keseluruhan kejadian stroke merupakan stroke iskemia. Salah satu komponen diagnosis stroke dilakukan dengan menganalisis citra CT scan pasien. Dalam pencitraan CT, stroke iskemia ditandai dengan adanya daerah yang gelap (*hypodense*). Namun, pada pendeteksian awal stroke iskemia menggunakan CT scan, daerah *hypodense* pada otak pasien tidak bisa terlihat secara jelas. Oleh sebab itu, dibutuhkan suatu metode yang mampu mengenali daerah *hypodense* dari citra CT scan otak. Pada penelitian ini dilakukan segmentasi otomatis pada lesi *hypodense* dengan menggunakan CNN. Kerangka kerja dari penelitian ini dapat dibagi menjadi 2 bagian utama yaitu: (1) pemrosesan awal pada *dataset* citra CT scan otak yang akan digunakan sebagai masukan model CNN serta bagian (2) perancangan CNN dengan arsitektur U-Net untuk mendeteksi *lesi hypodense*. Pemrosesan awal menghasilkan citra yang sudah ditingkatkan dengan metode DWT-CLAHE. Berdasarkan perhitungan kuantitatif terhadap beberapa metrik diperoleh citra dengan penggunaan *mother wavelet* jenis Biorthogonal 5.5 serta dekomposisi level 1 pada DWT menghasilkan citra yang paling baik. Peningkatan citra ini juga berpengaruh terhadap proses *learning* CNN. Berdasarkan percobaan yang dilakukan, CNN dengan *input* berupa citra yang telah ditingkatkan memiliki nilai DSC yang lebih besar daripada CNN dengan *input* berupa citra tanpa peningkatan. Percobaan dilakukan terhadap *hyperparameter epoch* untuk menentukan jumlah *epoch* yang tepat untuk proses *learning* model. Berdasarkan nilai *Dice Similarity Coefficient* (DSC) yang diperoleh pada masing-masing *epoch*, dapat disimpulkan bahwa *epoch* 100 memberikan hasil segmentasi yang paling baik. Hasil pengujian terhadap hasil akhir sistem menunjukkan nilai DSC sebanyak 0.932 yang menunjukkan bahwa citra hasil prediksi dan *ground truth* 93.2% sama. Pengembangan lebih lanjut dapat dilakukan untuk klasifikasi stroke iskemia dan stroke hemoragik.

Kata kunci : *Computed Tomography*, stroke iskemia, segmentasi citra, *Convolutional Neural Network*

SEGMENTATION OF HYPODENSE LESION IN BRAIN CT SCAN IMAGE USING CNN U-NET ARCHITECTURE FOR EARLY DETECTION OF ISCHEMIC STROKE

Name : Bhetri Sonia Yolandari
Student Identity Number : 07311740000029
Supervisors : 1. Dr. Tri Arief Sardjono, S.T., M.T.
2. Nada Fitriyatul Hikmah, S.T., M.T.

ABSTRACT

Stroke or cerebrovascular accident is a condition that occurs when blood vessels that carry oxygen and nutrients to the brain are blocked or ruptured, that lead brain cells to die. 87% of all strokes are ischemic stroke. Strokes are usually diagnosed by doing physical tests and studying images of the brain produced during a scan by using CT scan. In CT imaging, ischemic stroke indicated by the presence of dark areas (hypodense). However, in the early detection of ischemic stroke using a CT scan, the hypodense areas of the patient's brain cannot be clearly seen. Therefore, a method that can recognize hypodense areas from CT scan images of the brain is needed. In this study, automatic segmentation of hypodense lesions was performed using CNN. The framework of this research can be divided into 2 main parts: (1) preprocessing of the CT scan image dataset of the brain which will be used as input for the CNN model and (2) the design of CNN with U-Net architecture to detect hypodense lesions. Preprocessing produces an image that has been enhanced using DWT-CLAHE method. Based on quantitative calculations on several metrics, mother wavelet Biorthogonal 5.5 and level 1 decomposition on DWT produces the best enhanced image. This image enhancement also affects the CNN learning process. Based on experiments, CNN with input enhanced images has a higher DSC value than CNN with input images without enhancement. Experiments were carried out on the epoch hyperparameter to determine the right number of epochs for the learning process. Based on Dice Similarity Coefficient (DSC) values obtained for each epoch, it can be concluded that epoch 100 provides the best segmentation results. Result resting of the system obtained a DSC value of 0.932 which indicates that the predicted image and ground truth are 93.2% identical. Further research can be done to classify ischemic stroke and hemoragic stroke.

Keywords: Computed Tomography, ischemic stroke, image segmentation, Convolutional Neural Network

KATA PENGANTAR

Segala puji bagi Allah SWT atas segala rahmat-Nya, karena-Nya Tugas Akhir ini dapat berjalan lancar dan selesai tepat pada waktunya. Tujuan dari penulisan Tugas Akhir ini adalah untuk memenuhi persyaratan kelulusan program Studi Strata I pada Departemen Teknik Biomedik di Institut Teknologi Sepuluh Nopember. Penulis menyadari bahwa penyusunan laporan ini tidak lepas dari bantuan berbagai pihak, maka pada kesempatan ini penulis menyampaikan ucapan terima kasih kepada:

1. Kedua orang tua yang selalu mendukung, memberi nasihat, serta memberikan fasilitas untuk membantu penyelesaian Tugas Akhir ini.
2. Bapak Dr. Tri Arief Sardjono, S.T., M.T. dan Ibu Nada Fitriyatul Hikmah, S.T., M.T., selaku dosen pembimbing yang senantiasa membimbing, mengarahkan, serta menginspirasi penulis selama pengerjaan penelitian ini.
3. Dr. Achmad Arifin, S.T., M.Eng. selaku kepala departemen yang telah memberi inspirasi, motivasi, serta bimbingan selama perkuliahan di ITS.
4. Bapak ibu dosen dan tenaga pendidik Departemen Teknik Biomedik yang telah membimbing dan mengajarkan banyak hal kepada penulis.
5. Keluarga besar 7B yang amat penulis sayangi.
6. Teman-teman BME-3 yang telah membantu penulis selama masa perkuliahan di Teknik Biomedik.
7. Serta seluruh pihak yang tidak dapat disebutkan satu persatu yang telah membantu penulis hingga Tugas Akhir ini ini bisa diselesaikan dengan semaksimal mungkin.

Terlepas dari semua itu, penulis sadar bahwa masih terdapat kekurangan pada berbagai aspek dan masih banyak hal yang perlu diperbaiki. Oleh karena itu, penulis dengan terbuka menerima kritik dan saran dari pembaca agar dapat memperbaiki Tugas Akhir ini. Semoga penelitian ini dapat memberikan manfaat untuk banyak orang dan dapat dikembangkan lebih baik lagi pada penelitian selanjutnya.

Surabaya, Juli 2021

Bhetri Sonia Yolandari
NRP. 0731174000029

DAFTAR ISI

PERNYATAAN KEASLIAN TUGAS AKHIR	v
ABSTRAK.....	vii
ABSTRACT.....	viii
KATA PENGANTAR	ix
DAFTAR ISI.....	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL.....	xv
BAB 1 PENDAHULUAN	16
1.1. Latar Belakang	16
1.2. Perumusan Masalah	17
1.3. Batasan Masalah	17
1.4. Tujuan dan Manfaat	18
1.5. Kontribusi	18
BAB 2 TINJAUAN PUSTAKA	19
2.1. Anatomi Otak	19
2.2. Stroke Iskemia	20
2.3. Pencitraan pada Stroke Iskemia	22
2.4. Median filter	24
2.5. Laplacian Filter	25
2.6. Discrete Wavelet Transform (DWT)	26
2.7. Contrast Limited Adaptive Histogram Equalization (CLAHE)	30
2.8. Convolutional Neural Network (CNN)	31
2.8.1. <i>Convolutional Layer</i>	33
2.8.2. <i>Pooling Layer</i>	34
2.8.3. <i>Fully-Connected Layer</i>	34
2.9. Arsitektur U-Net	34
2.10. Rasional	36
BAB 3 METODE PENELITIAN	39

3.1. Diagram Blok Sistem	39
3.2. Data.....	40
3.3. <i>Preprocessing</i>	40
3.3.1 <i>Grayscale</i>	41
3.3.2 Filter median dan morfologi	41
3.3.3 <i>Autocropping</i>	41
3.3.4 Segmentasi Otak.....	41
3.3.5 Peningkatan Citra	42
3.4. Perancangan CNN.....	42
3.5. Pelatihan CNN.....	45
3.6. Pengujian Performansi	45
3.6.1. Peningkatan Citra	45
BAB 4 PENGUJIAN SISTEM	49
4.1 <i>Preprocessing</i>	49
4.1.1 <i>Grayscale</i>	49
4.1.2 Filter median dan morfologi	50
4.1.3 <i>Autocropping</i>	58
4.1.4 Segmentasi Otak.....	58
4.1.5 Peningkatan Citra.....	58
4.2 Pelatihan CNN.....	70
BAB 5 PEMBAHASAN.....	82
5.1. <i>Preprocessing</i>	82
5.1.1. <i>Grayscale</i>	82
5.1.2. Filter median dan morfologi	82
5.1.3. <i>Autocropping</i>	83
5.1.4. Segmentasi Otak.....	83
5.1.5. Peningkatan Citra.....	83
5.2. Pelatihan CNN.....	88
5.3. Hasil Performansi.....	91
BAB 6 KESIMPULAN DAN SARAN	94
6.1. Kesimpulan	94

6.2. Saran.....	95
6.3. Ucapan Terima Kasih	95
DAFTAR PUSTAKA	96
LAMPIRAN.....	98
A. Daftar sumber data	98
B. Hasil peningkatan citra dengan mother wavelet Biorthogonal 5.5 level 1.....	99
C. Hasil segmentasi	102
BIODATA PENULIS	105

DAFTAR GAMBAR

Gambar 2.1 Struktur dan Fungsi Otak [11]	19
Gambar 2.2 Bagian vaskular serebral otak [12].....	20
Gambar 2.3 Perbedaan thrombosis, emboli dan hipoperfusi sistemik [12]	21
Gambar 2.4 (A) pompa normal dan sirkulasi arteri; (B) tekanan pompa rendah dan iskemia pada borderzone, (C) pembuluh tersumbat dan infark pada arteri cerebral tengah [12]	22
Gambar 2.5 (A) infarct pada tahap awal(panah berwarna putih) yang melibatkan daerah MCA kanan dari cerebral cortex; (B) infarct dengan batas jelas yang diambil 36 jam setelah serangan [12].....	24
Gambar 2.6 Median filter.....	24
Gambar 2.7 Hasil penerapan Median filter pada citra [6]	25
Gambar 2.8 Laplacian Filter	26
Gambar 2.9 Deteksi tepi menggunakan Laplacian Filter [6].....	26
Gambar 2.10 Sub-band LL, LH, HL dan HH dari sebuah gambar	27
Gambar 2.11 Ilustrasi DWT.....	28
Gambar 2.12 Haar wavelet [16].....	28
Gambar 2.13 Coiflet wavelet [16]	29
Gambar 2.14 Daubechies wavelet [16].....	29
Gambar 2.15 Symlet wavelet [16]	29
Gambar 2.16 Discrete Meyer wavelet	30
Gambar 2.17 Wavelet biorthogonal.....	30
Gambar 2.18 Konsep dari Contrast Limited Adaptive Histogram Equalization [6].....	30
Gambar 2.19 Struktur dasar ANN	31
Gambar 2.20 Perbandingan antara ANN dan CNN [22]	32
Gambar 2.21 Ilustrasi konvolusi pada citra [23].....	33
Gambar 2.22 Ilustrasi max pooling.....	34
Gambar 2.23 Arsitektur U-Net	35
Gambar 2.24 Fish bone diagram	38
Gambar 3.1 Diagram blok sistem	40
Gambar 3.2 ReLU activation function.....	43
Gambar 3.3 Arsitektur U-Net dengan encoder Resnet	44
Gambar 4.1 Tahap preprocessing	49
Gambar 4.2 Hasil pengecekan data citra.....	51
Gambar 4.3 Hasil grayscaling.....	53
Gambar 4.4 Hasil median filter.....	55
Gambar 4.5 Hasil morfologi filter	57
Gambar 4.6 Hasil autocropping	60
Gambar 4.7 Hasil segmentasi otak dan tengkorak.....	62
Gambar 4.8 Biorthogonal 1.1.....	63
Gambar 4.9 Biorthogonal 2.2.....	63
Gambar 4.10 Biorthogonal 3.3.....	63
Gambar 4.11 Biorthogonal 4.4.....	63
Gambar 4.12 Biorthogonal 5.5.....	64
Gambar 4.13 Biorthogonal 6.8.....	64
Gambar 4.14 Coiflet 1.....	64
Gambar 4.15 Coiflet 2.....	64
Gambar 4.16 Coiflet 3.....	65
Gambar 4.17 Coiflet 4.....	65
Gambar 4.18 Coiflet 5.....	65

Gambar 4.19 Daubechies 1	65
Gambar 4.20 Daubechies 5	66
Gambar 4.21 Daubechies 10	66
Gambar 4.22 Daubechies 15	66
Gambar 4.23 Daubechies 20	66
Gambar 4.24 Discrete meyer	67
Gambar 4.25 Haar	67
Gambar 4.26 Reverse biorthogonal 1.1	67
Gambar 4.27 Reverse biorthogonal 3.3	67
Gambar 4.28 Reverse biorthogonal 4.4	68
Gambar 4.29 Reverse biorhogonal 5.5	68
Gambar 4.30 Reverse biorhogonal 6.8	68
Gambar 4.31 Symlet 2	68
Gambar 4.32 Symlet 5	69
Gambar 4.33 Symlet 10	69
Gambar 4.34 Symlet 15	69
Gambar 4.35 Symlet 20	69
Gambar 5.1 Daerah diluar ROI (ditandai dengan marker merah)	83
Gambar 5.2 Deteksi tepi dengan Laplacian filter	84
Gambar 5.3 Perbandingan mother wavelet Biorthogonal	84
Gambar 5.4 Perbandingan mother wavelet Coiflet	85
Gambar 5.5 Perbandingan mother wavelet reverse biorthogonal	86
Gambar 5.6 Perbandingan mother wavelet Daubechies	86
Gambar 5.7 Perbandingan mother wavelet Symlet	87
Gambar 5.8 Perbandingan antar mother wavelet	87
Gambar 5.9 Perbandingan antar level DWT	88
Gambar 5.10 Learning curve CNN dengan epoch 50	90
Gambar 5.11 Learning curve CNN dengan epoch 10	90
Gambar 5.12 Learning curve CNN dengan epoch 100	91
Gambar 5.13 Hasil peningkatan citra pada kasus posterior cerebral artery territory infarct ...	91
Gambar 5.14 Hasil segmentasi yang tidak mengandung ROI	93

DAFTAR TABEL

Tabel 2.1 Kelebihan dan kekurangan CT scan pada pemindaian pasien stroke	23
Tabel 2.2 Abnormalitas pada citra CT scan pasien stroke iskemia	23
Tabel 2.3 Perbedaan ANN dan CNN	32
Tabel 4.1 Model summary	70
Tabel 4.2 Hasil segmentasi CNN dengan epoch 10.....	76
Tabel 4.3 Hasil segmentasi CNN dengan epoch 50.....	77
Tabel 4.4 Hasil segmentasi CNN dengan epoch 100.....	78
Tabel 4.5 Hasil segmentasi CNN epoch 10 dengan input citra tanpa peningkatan	79
Tabel 4.6 Hasil segmentasi CNN epoch 50 dengan input citra tanpa peningkatan	80
Tabel 4.7 Hasil segmentasi CNN epoch 100 dengan input citra tanpa peningkatan	81

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Stroke atau cedera *cerebrovascular* merupakan penyakit yang mempengaruhi pembuluh darah menuju atau di dalam otak. Stroke terjadi ketika pembuluh darah yang membawa oksigen dan nutrisi ke otak mengalami penyumbatan atau pecah sehingga asupan oksigen menuju otak menjadi berkurang dan menyebabkan sel otak yang kekurangan oksigen tersebut mati. Berdasarkan data dari *World Health Organization* (WHO), stroke merupakan penyebab kematian terbanyak kedua dan bertanggung jawab terhadap 11% dari total kematian di dunia [1]. Angka kejadian stroke di Indonesia juga mengalami peningkatan prevalensi. Riset Kesehatan Dasar (Riskesdas) 2018 menunjukkan bahwa prevalensi stroke di Indonesia meningkat dari 7% menjadi 10,9% jika dibandingkan dengan Riskesdas pada tahun 2013 [2].

Berdasarkan patologi anatomi dan penyebabnya, stroke dapat dibagi menjadi 2 jenis yaitu stroke iskemik dan stroke hemoragik. Stroke iskemik adalah stroke yang terjadi akibat tersumbatnya pembuluh darah yang menyebabkan aliran darah ke sebagian atau keseluruhan otak terhenti. Penyumbatan bisa terjadi di sepanjang jalur pembuluh darah arteri yang menuju ke otak. Stroke hemoragik adalah stroke yang terjadi akibat pecahnya salah satu arteri di dalam otak yang memicu pendarahan sehingga aliran darah pada sebagian otak berkurang atau terputus. Sebanyak 87% dari keseluruhan kejadian stroke merupakan stroke iskemik [3]. Pengenalan gejala-gejala stroke secara cepat oleh dokter gawat darurat medis (*Emergency Medicine*) merupakan kunci dari penanganan stroke, karena kecepatan dari penanganan dapat mengubah gejala sisa neurologis dari AIS (*Acute Ischemic Stroke*) secara drastis [4].

Diagnosis stroke dilakukan dengan menganalisis hasil dari berbagai pengujian diagnostik. Salah satunya adalah dengan melakukan analisis terhadap citra medis pasien. Ada berbagai modalitas yang tersedia untuk memperoleh citra medis pasien, diantaranya CT scan, MRI, PET, dan lain sebagainya. Diantara berbagai modalitas tersebut, CT (*Computed Tomography Scan*) adalah prosedur yang lebih sering digunakan dalam diagnosis awal stroke, karena menawarkan keunggulan dalam aspek kecepatan, biaya, dan kriteria lainnya jika dibandingkan dengan pencitraan MRI. Analisis terhadap citra CT pasien dilakukan untuk mendiferensiasi gangguan vaskular dan non-vaskular, menentukan jenis stroke yang terjadi, dan lain sebagainya [5].

Dalam pencitraan CT, stroke iskemik ditandai dengan adanya daerah yang gelap (*hypodense*) dengan kontras yang relatif terhadap daerah disekelilingnya, tergantung pada waktu sejak stroke terjadi. Namun, pendeteksian stroke iskemik menggunakan CT scan memiliki kekurangan yaitu ketika dilakukan *scanning* awal daerah *hypodense* pada otak pasien tidak bisa terlihat secara jelas dan akan menjadi lebih terlihat pada beberapa jam berikutnya [6]. Selain itu, analisis visual yang dilakukan pada citra CT scan bersifat subjektif, sehingga dapat berbeda antara satu radiologis dengan radiologis lainnya. Oleh sebab itu, dibutuhkan suatu metode yang mampu mengenali daerah *hypodense* dari citra CT scan otak dengan bantuan komputer (*Computer Aided Diagnosis*) sehingga dapat menjadi pemberi opini kedua untuk membantu radiologis dalam mendeteksi stroke iskemik.

Pada penelitian sebelumnya, Tan et al. melakukan peningkatan kontras pada citra CT scan otak menggunakan metode *Histogram Equalization* [7], namun proses peningkatan dengan metode ini tidak hanya meningkatkan kontras daerah *hypodense*. Metode ini juga ikut

meningkatkan kontras jaringan normal sehingga antara daerah *hypodense* dan jaringan normal masih belum dapat didiferensiasi. Lalu, Jayakumar melakukan pendeteksian cacat pada citra CT scan dengan menggunakan metode *Contrast Limited Adaptive Histogram Equalization* (CLAHE) [8] dan berhasil memperbaiki kontras citra. Selanjutnya, Boraste et al. [9] melakukan penelitian menggunakan metode *Discrete Wavelet Transform* (DWT) untuk meningkatkan resolusi citra. Kemudian, Lidong et al. [10] mengkombinasikan metode *Discrete Wavelet Transform* (DWT) dengan *Contrast Limited Adaptive Histogram Equalization* (CLAHE) untuk *denoising*, memperjelas detail dan meningkatkan kontras pada citra. Namun pada penelitian tersebut citra yang digunakan adalah citra non medis. Andina dkk. [6] selanjutnya menerapkan metode ini untuk mendeteksi lesi *hypodense* pada citra CT scan otak. Metode ini terbukti mampu meningkatkan kontras lesi *hypodense* secara tepat dan tidak berlebihan. Namun pada penelitian ini proses segmentasi lesi *hypodense* masih dilakukan secara manual. Pada penelitian ini dilakukan *thresholding-based segmentation* untuk mendeteksi daerah *hypodense*. Namun, belum mampu mendeteksi daerah *hypodense* yang ukurannya sangat kecil, dan belum mampu mensegmentasi secara jelas antara daerah *hypodense* dengan jaringan lunak normal dan ventrikel otak secara otomatis.

Oleh sebab itu, pada Tugas Akhir ini dilakukan segmentasi otomatis pada lesi *hypodense* dengan menggunakan *Convolutional Neural Network*. *Convolutional Neural Network* (CNN) adalah salah satu jenis *neural network* yang digunakan untuk mengolah data dalam bentuk citra yang tersusun atas piksel. Melalui proses *learning* yang dilakukan oleh CNN, proses segmentasi dapat dilakukan secara otomatis. Dengan begitu diharapkan dapat membantu deteksi dini pada pasien stroke iskemia sehingga pasien dapat sesegera mungkin diberikan pengobatan yang tepat.

1.2. Perumusan Masalah

Salah satu permasalahan yang sering terjadi pada citra CT scan otak pasien stroke iskemia adalah tidak jelasnya lesi *hypodense* pada citra. Pada penelitian-penelitian sebelumnya, telah diperoleh suatu metode peningkatan citra yang mampu memberikan hasil yang baik sehingga proses peningkatan citra tidak dilakukan secara berlebihan namun efektif dalam memperjelas lesi *hypodense* pada citra stroke iskemia. Namun, pada penelitian tersebut proses pendeteksian lesi *hypodense* masih dilakukan dengan cara segmentasi manual oleh radiologis. Hal ini akan dapat menimbulkan kesalahan dalam melakukan interpretasi karena analisis visual yang dilakukan bersifat subjektif dan dengan keterbatasan dari jumlah radiologis akan membutuhkan waktu yang lama untuk mendapatkan hasil diagnosis. Oleh sebab itu, dibutuhkan sebuah metode untuk memperoleh bagian dari lesi *hypodense* secara otomatis sehingga dapat membantu mengurangi waktu yang dibutuhkan dalam diagnosis stroke iskemia dan dapat menjadi opini kedua bagi dokter dan radiologis dalam analisis citra CT scan pasien stroke iskemia.

1.3. Batasan Masalah

Batasan masalah pada penelitian ini yang pertama adalah data yang digunakan merupakan data citra CT scan otak pasien stroke iskemia yang diperoleh dari platform *open source* dan disertai anotasi deskriptif yang menunjukkan letak daerah *hypodense* pada citra CT scan otak. Batasan masalah kedua adalah stroke yang dideteksi merupakan jenis stroke iskemia. Batasan

masalah yang terakhir adalah metode yang diterapkan merupakan metode dari penelitian-penelitian yang sudah ada sebelumnya.

1.4. Tujuan dan Manfaat

Tujuan dari penelitian ini adalah untuk memperoleh suatu metode yang dapat meningkatkan kualitas citra CT scan otak pasien stroke iskemia dan metode untuk segmentasi terhadap lesi *hypodense* pada citra otak sehingga dapat menjadi opini kedua bagi radiologis dalam diagnosis stroke iskemia. Diharapkan dengan adanya metode ini, segmentasi lesi *hypodense* dapat dilakukan secara otomatis dengan menggunakan CNN sehingga dapat mempersingkat waktu yang merupakan suatu hal krusial dalam penanganan awal stroke.

Manfaat yang ingin dicapai dalam penelitian ini meliputi aspek teoritis dan praktis. Manfaat dari aspek teoritis yang diharapkan ialah dapat menambah wawasan yang berkaitan dengan pengolahan citra dari CT scan untuk penderita stroke iskemia dan menambah wawasan terkait *Convolutional Neural Network*. Manfaat dari aspek praktis yang diharapkan adalah dapat membantu dan mempermudah dalam penentuan daerah *hypodense* pada citra CT scan otak untuk deteksi stroke iskemia.

1.5. Kontribusi

Kontribusi dari penelitian terbagi menjadi dua aspek utama yaitu aspek ilmiah dan aspek praktis. Kontribusi aspek ilmiah dalam penelitian ini adalah untuk mengembangkan metode yang dapat digunakan untuk mendeteksi stroke iskemia melalui segmentasi lesi *hypodense* dari citra CT scan otak. Sehingga pada penelitian ini ditawarkan suatu aspek kebaharuan dimana digunakan metode lain yaitu metode *Convolutional Neural Network* untuk melakukan segmentasi terhadap citra otak agar diperoleh hasil yang lebih baik dengan waktu proses yang lebih cepat.

Kontribusi aspek praktis dalam penelitian ini adalah untuk memberikan pilihan metode segmentasi kepada radiologis dan dokter untuk mendeteksi lesi *hypodense* pada citra stroke iskemia. Hal ini didasari pada fakta bahwa selama ini deteksi lesi *hypodense* dilakukan secara manual sehingga membutuhkan banyak waktu dan bersifat subjektif. Oleh sebab itu, melalui metode ini diharapkan proses segmentasi dari lesi *hypodense* pada citra otak dapat dilakukan secara otomatis dan dapat membantu radiologis dan dokter dalam pengambilan keputusan.

BAB 2

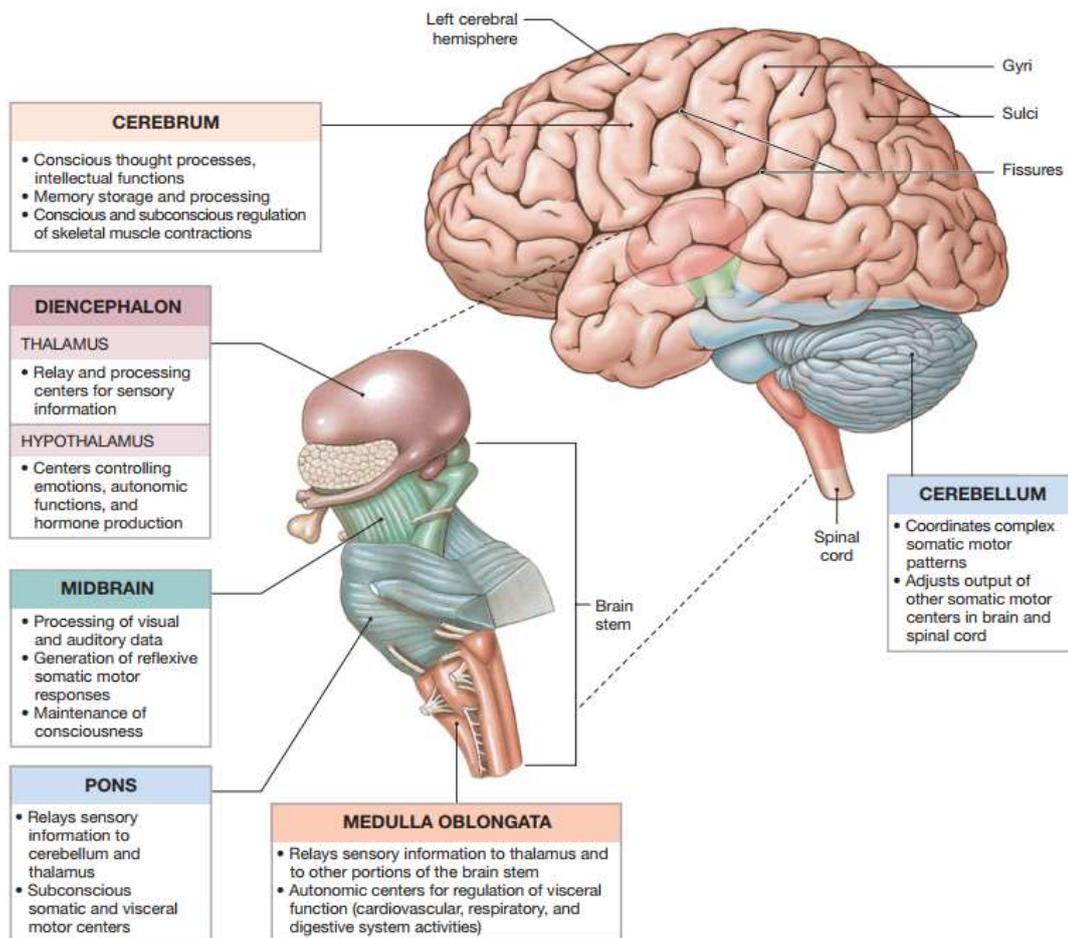
TINJAUAN PUSTAKA

2.1. Anatomi Otak

Otak merupakan organ tubuh yang berfungsi sebagai pusat sistem saraf pada manusia dan merupakan organ paling kompleks dengan jumlah neuron mencapai 14-16 milyar. Otak dibagi menjadi beberapa bagian besar seperti yang ditunjukkan pada Gambar 2.1 [11].

Bagian terbesar pada otak manusia adalah *cerebrum* yang terbagi menjadi sepasang *cerebral hemisphere*. Permukaan *cerebral hemisphere* ditutupi oleh *cerebral cortex*, yaitu lapisan superfisial berupa materi abu-abu (*gray matter*). *Cerebral cortex* ini membentuk lipatan-lipatan yang disebut dengan *gyri*(*gyrus*) untuk memperlebar permukaannya. *Gyri* ini dipisahkan oleh celah dengan kedalaman yang berbeda-beda. Celah yang dangkal disebut dengan *sulcus* dan celah yang dalam disebut dengan *fissures*. *Cerebrum* merupakan pusat kendali pikiran, kecerdasan, memori, dan gerakan-gerakan kompleks [11].

Didalam *cerebrum* terdapat bagian otak yang disebut dengan *diencephalon*. *Diencephalon* terdiri dari *left thalamus* dan *right thalamus* yang merupakan pusat pemrosesan informasi sensorik. Bagian bawah dari *diencephalon* disebut dengan *hypothalamus* yang mengatur emosi, fungsi autonomik, dan produksi hormon [11].



Gambar 2.1 Struktur dan Fungsi Otak [11]

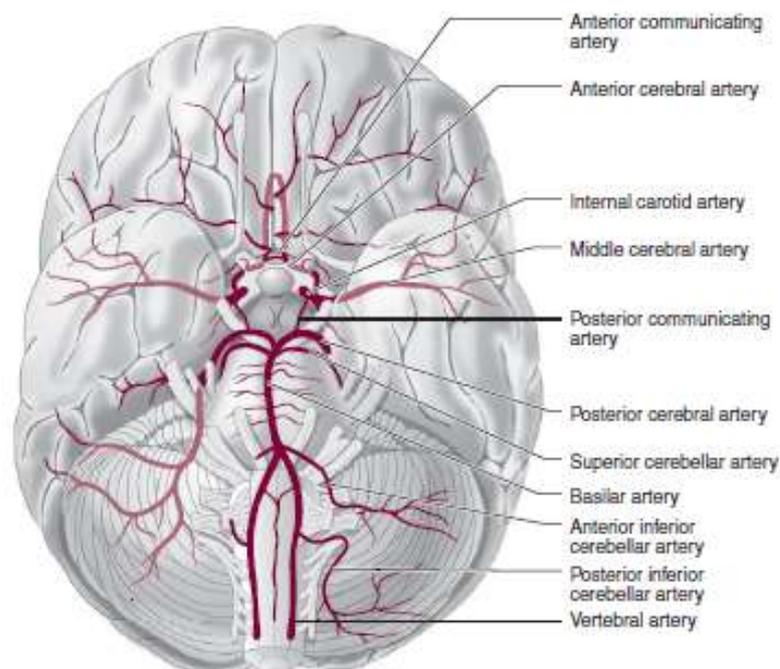
Cerebellum merupakan bagian terbesar kedua dari otak. Sama seperti *cerebrum*, *cerebellum* juga dilapisi oleh lapisan *gray matter* yang disebut *cerebellar cortex*. *Cerebellum* mengendalikan gerakan, menjaga keseimbangan, dan merupakan pusat koordinasi tubuh [11].

Batang otak atau *brain stem* terdiri dari otak tengah, *pons*, dan *medulla oblongata*. Otak tengah atau *mesencephalon* mengandung *nuclei* yang memproses informasi visual dan audio serta mengontrol gerak refleks. *Pons* berfungsi untuk menghubungkan *cerebellum* dengan batang otak. *Pons* juga memiliki *nuclei* yang terlibat dalam pengaturan motor *somatic* dan *visceral*. *Medulla oblongata* berfungsi untuk menghubungkan otak dengan tulang belakang, menyampaikan informasi sensorik ke *thalamus* dan batang otak, dan mengatur beberapa fungsi autonomik seperti detak jantung, tekanan darah, dan pencernaan [11].

Otak dengan jutaan neuronnya adalah organ yang sangat aktif dan membutuhkan asupan oksigen dan nutrient secara kontinu. Asupan ini dipenuhi oleh sirkulasi darah ke otak melalui *internal carotid arteries* dan *vertebral arteries* seperti yang terlihat pada Gambar 2.2 [12]. Ketika suplai darah ke otak ini berkurang, neuron disekitarnya akan mati. Kejadian ini disebut dengan *cerebrovascular accident (CVA)* atau stroke.

2.2. Stroke Iskemia

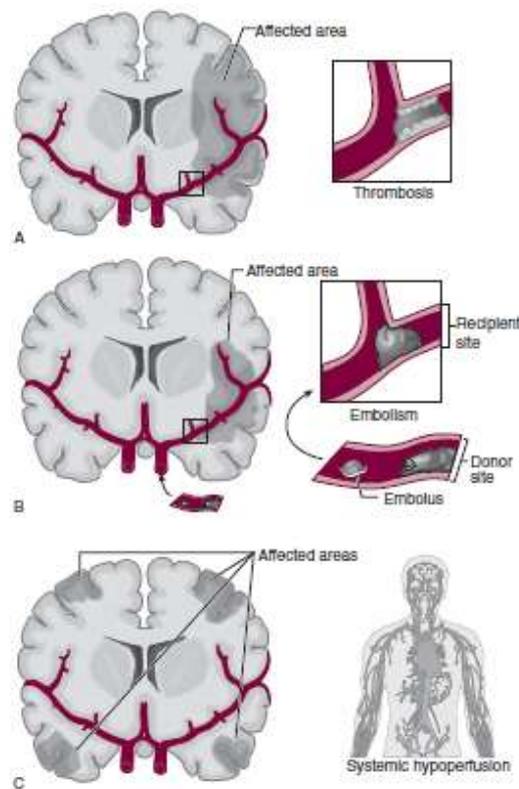
Stroke atau penyakit *cerebrovascular* adalah penyakit yang mempengaruhi pembuluh yang menyuplai darah ke otak. Istilah stroke mengacu pada cedera yang terjadi pada otak atau tulang belakang yang disebabkan oleh ketidaknormalan suplai darah [12]. Stroke dapat terjadi ketika pembuluh yang menyuplai darah pecah atau terjadi penyumbatan pada pembuluh darah. Karena adanya kekurangan oksigen, sel saraf pada daerah otak yang terkena tidak dapat melakukan fungsi dasar yang menyebabkan kematian pada jaringan otak dan dapat mengakibatkan disabilitas jangka panjang bahkan kematian [13].



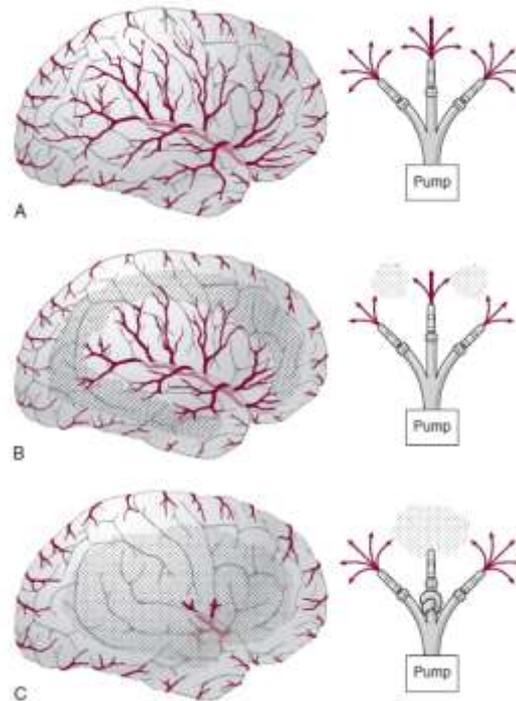
Gambar 2.2 Bagian vaskular serebral otak [12]

Berdasarkan patologi anatomi dan penyebabnya, stroke dibagi menjadi dua yaitu stroke iskemia dan stroke hemoragik. Stroke iskemia adalah stroke yang disebabkan oleh tersumbatnya pembuluh darah yang menyebabkan aliran darah ke sebagian atau keseluruhan otak terhenti. Penyumbatan bisa terjadi di sepanjang jalur pembuluh darah arteri yang menuju ke otak. Stroke hemoragik adalah kondisi pecahnya salah satu arteri dalam otak yang memicu pendarahan sehingga aliran darah pada sebagian otak berkurang atau terputus [12].

Stroke iskemia dapat dibagi menjadi tiga mekanisme yang berbeda yaitu trombotik, emboli, dan hipoperfusi sistemik. Perbedaan antara ketiga mekanisme tersebut dapat dilihat pada Gambar 2.3 [12]. Trombotik mengacu pada penyumbatan aliran darah karena proses oklusif (penyumbatan) lokal dalam satu atau lebih pembuluh darah yang disebabkan oleh adanya plak, aterosklerosis (pengerasan arteri yang disebabkan oleh timbunan plak kolesterol), arteritis (inflamasi pada dinding arteri), dan lain sebagainya. Ketika bahan terbentuk di tempat lain dalam sistem vaskular bersarang di arteri dan menghambat aliran darah, maka disebut dengan emboli. Penyumbatan dapat bersifat sementara atau dapat bertahan selama berjam-jam atau berhari-hari sebelum bergerak ke distal. Hipoperfusi sistemik (penurunan perfusi) terjadi apabila aliran menuju jaringan otak berkurang disebabkan oleh tekanan perfusi sistemik yang rendah. Penyebab paling umum dari rendahnya tekanan perfusi sistemik ini ialah kegagalan dalam pompa jantung seperti *myocardial infarction* atau *arrhythmia* dan hipotensi sistemik karena kehilangan darah atau *hypovolemia*. Perfusi paling kritis terjadi di zona perbatasan atau yang disebut *watershed regions* di periferal wilayah pasokan vaskular utama seperti pada Gambar 2.4 [12]. Ketiga mekanisme stroke iskemik tersebut dapat menyebabkan kerusakan jaringan sementara atau permanen (*infarction*) [12].



Gambar 2.3 Perbedaan thrombosis, emboli dan hipoperfusi sistemik [12]



Gambar 2.4 (A) pompa normal dan sirkulasi arteri; (B) tekanan pompa rendah dan iskemia pada *borderzone*, (C) pembuluh tersumbat dan infark pada arteri *cerebral* tengah [12]

Pembuluh kapiler dan pembuluh lain yang berada dalam jaringan iskemik juga dapat mengalami cedera sehingga reperfusi (restorasi aliran darah setelah terjadinya pendarahan) dapat menyebabkan kebocoran darah yang berakhir pada *hemorrhagic infarction*. Kerusakan otak dan pembuluh darah ini dapat menyebabkan *brain edema* (pembengkakan otak akibat penumpukan cairan) dalam beberapa jam atau hari setelah stroke terjadi [12]. Oleh sebab itu, waktu merupakan faktor yang sangat menentukan untuk mengurangi risiko kematian dan kecacatan permanen. Identifikasi yang benar dan tepat waktu dapat memungkinkan dokter untuk memulai terapi yang tepat, sehingga dapat memberikan hasil fungsional yang lebih baik.

2.3. Pencitraan pada Stroke Iskemia

Diagnosis penyakit stroke dilakukan dengan menggunakan informasi-informasi berikut: (1) analisis gejala neurologi, (2) hasil pemeriksaan neurologi, dan (3) hasil pemeriksaan citra otak dan vaskular [12]. Prosedur pencitraan otak telah menjadi salah satu bagian yang tidak terpisahkan pada evaluasi pasien dengan penyakit *cerebrovascular* [12]. Hal ini karena stroke adalah penyakit dengan potensi merusak yang sangat tinggi sehingga diagnosis yang tepat dari berbagai informasi menjadi sangat penting. Pencitraan otak dapat dilakukan dengan menggunakan berbagai macam modalitas, seperti CT scan, MRI, PET, dan lain sebagainya. Diantara berbagai modalitas tersebut, *Computed Tomography* (CT) adalah prosedur yang lebih sering digunakan dalam diagnosis awal stroke akut, karena menawarkan keunggulan dalam aspek kecepatan, biaya, dan kriteria lainnya jika dibandingkan dengan pencitraan MRI. Tabel 2.1 menunjukkan kelebihan dan kekurangan CT scan pada pemindaian pasien stroke [12].

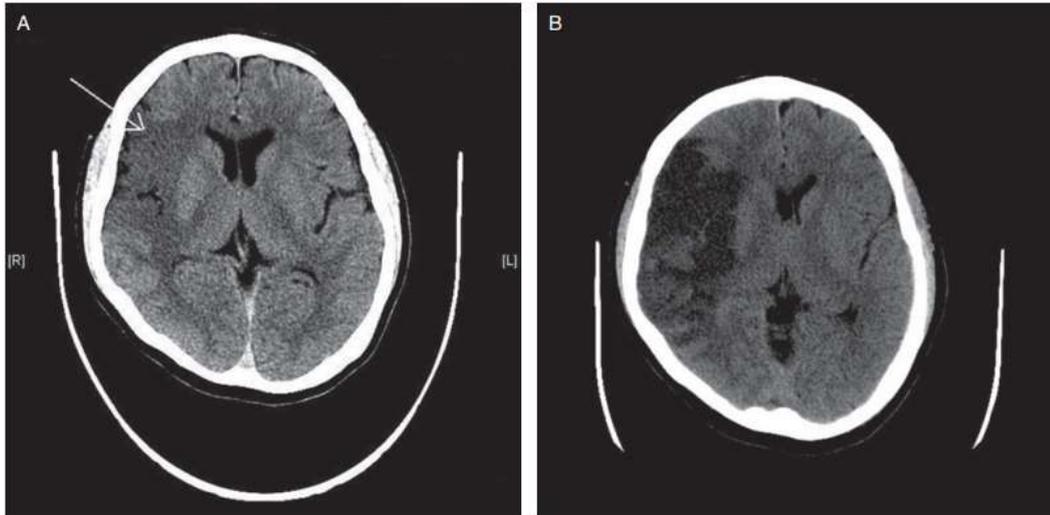
Tabel 2.1 Kelebihan dan kekurangan CT scan pada pemindaian pasien stroke

Kelebihan CT scan	Kekurangan CT scan
Lebih banyak tersedia dan lebih mudah untuk ditemukan dibandingkan MRI.	MRI lebih peka dalam mendeteksi dan melakukan pencitraan infark akut pada otak jika dibandingkan dengan CT.
CT scan dari ICH(<i>intracerebral brain hemorrhage</i>) lebih mudah diinterpretasikan daripada MRI dan pada sebagian besar keadaan, menghasilkan data yang memadai untuk pengambilan keputusan klinis tanpa perlu pemindaian MRI.	CT tidak akurat dalam menggambarkan lesi yang berdekatan dengan permukaan tulang (misalkan di orbital, kutub frontal, dan <i>lobus temporal</i>). MRI lebih baik dalam mencitrakan batang otak dan infark <i>cerebral</i> .
CT scan lebih mudah dibaca oleh sebagian besar dokter yang bukan stroke neurologis atau neuroradiologis.	CT tidak dapat digunakan dalam mendeteksi dan menggambarkan infark pada sumsum tulang belakang
CT scan lebih murah dibandingkan MRI.	
CT scan membutuhkan waktu pemindaian yang lebih singkat.	

Computed Tomography Scan atau CT-scan merupakan alat diagnostik dengan teknik radiografi yang menghasilkan gambar potongan tubuh secara melintang berdasarkan penyerapan sinar-X pada irisan tubuh yang ditampilkan pada layar monitor. *Computed Tomography* (CT) disebut juga dengan *Computed Axial Tomography* (CAT), *Computed-assisted Tomography*, atau *Body Section Rontgenography* yang merupakan suatu proses menggunakan *digital processing* dari satu rangkaian sinar-X yang menghasilkan gambar dua dimensi. Pencitraan CT dari organ dalam misalnya tulang, jaringan lunak, dan pembuluh darah menghasilkan detail gambar yang jauh lebih jelas daripada sinar-X konvensional, terutama gambaran dari jaringan lunak dan pembuluh darah. Tabel 2.2. menunjukkan abnormalitas yang terjadi pada citra CT pasien dengan stroke iskemia [12].

Tabel 2.2 Abnormalitas pada citra CT scan pasien stroke iskemia

No.	Abnormalitas
1.	Hilangnya perbedaan antara materi abu-abu dan putih(<i>gray and white matter</i>) yang mengindikasikan cedera yang ireversibel
2.	Penipisan sulkus yang bersifat reversibel
3.	<i>Hypodensity</i> pada daerah <i>infarct</i>
4.	<i>Hyperdense arteries</i> yang mengindikasikan thrombosis
5.	Kalsifikasi emboli didalam arteri.



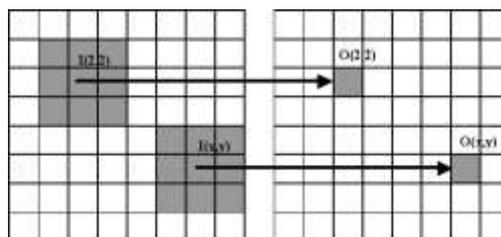
Gambar 2.5 (A) *infarct* pada tahap awal(panah berwarna putih) yang melibatkan daerah MCA kanan dari *cerebral cortex*; (B) *infarct* dengan batas jelas yang diambil 36 jam setelah serangan [12]

Pada stroke iskemia, CT dapat menunjukkan *infarct* sebagai lesi *hypodense* seperti pada Gambar 2.5[12]. Perubahan iskemik awal pada CT scan bisa sangat halus atau tidak terlalu terlihat dalam beberapa jam pertama timbulnya gejala seperti pada Gambar 2.5(A) dan akan terlihat lebih jelas setelah beberapa jam seperti pada Gambar 2.5(B).

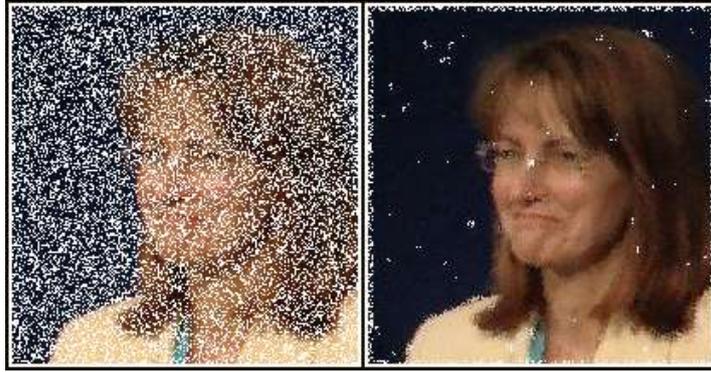
Selain diagnosis awal stroke, citra otak dibutuhkan oleh dokter untuk memperoleh informasi mengenai karakteristik lesi. Informasi seperti letak, ukuran, jangkauan, dan efek dari lesi terhadap struktur intrakranial dapat membantu dokter untuk menentukan mekanisme yang terjadi, pembuluh darah yang terlibat, dan menentukan pengobatan yang optimal [12].

2.4. Median filter

Noise atau derau adalah suatu sinyal gangguan yang terdapat pada citra yang bukan merupakan sinyal yang diinginkan. Untuk memperoleh hasil citra yang lebih baik, *noise* yang terdapat pada citra harus dikurangi atau dihilangkan dengan menggunakan filter. Median filter merupakan sebuah teknik *digital filtering* yang digunakan untuk menghilangkan *noise* dari citra. Median filter adalah salah satu metode filter nonlinear yang sering digunakan dalam pengolahan citra digital karena filter ini dapat mempertahankan karakteristik batas(*edge*) citra dan mereduksi *noise* pada citra tersebut. Nilai *noisy* dari sebuah piksel digantikan oleh nilai median dari piksel-piksel yang ada di sekitarnya seperti diilustrasikan pada Gambar 2.6. Piksel tetangga atau yang biasa disebut *mask* atau kernel ini diberi peringkat dalam urutan *gray level*-nya, dan nilai median grup disimpan untuk menggantikan nilai *noisy*. Keluaran dari median filter ditunjukkan oleh Persamaan 2.1 [14].



Gambar 2.6 Median filter



Gambar 2.7 Hasil penerapan Median filter pada citra [6]

$$g(x, y) = \text{med}\{f(x - i, y - j), i, j \in W\} \quad (2.1)$$

dimana (x, y) , $g(x, y)$ adalah gambar asli dan keluaran gambar secara berurutan. W adalah *mask* dua dimensi: ukuran *mask* adalah $n \times n$ (dimana n biasanya ganjil) seperti 3×3 , 5×5 , dan lain-lain. Bentuk *mask* bisa linier, kotak, lingkaran, *cross* dan lain-lain. Hasil penerapan *median filter* dapat dilihat pada Gambar 2.7.

2.5. Laplacian Filter

Edge detection atau penentuan batas citra merupakan metode matematis yang digunakan untuk menentukan poin dimana intensitas citra berubah secara drastis. *Edge detection* sangat penting dalam proses deteksi dan ekstraksi fitur.

Laplacian filter merupakan salah satu *edge detector* yang digunakan untuk menentukan batas(*edge*) dari citra. Filter ini bekerja dengan menghitung turunan kedua dari citra secara vertikal dan horizontal untuk menentukan gradien terbesar yang dideteksi sebagai batas dari citra seperti yang diilustrasikan pada Gambar 2.8. Dalam pemrosesan citra, *Laplacian filter* seringkali merujuk pada filter FIR sederhana berukuran 3×3 . Respon impuls dari *Laplacian filter* ditunjukkan oleh Persamaan 2.2 [15].

$$h = \frac{4}{\alpha+1} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}; 0 \leq \alpha \leq 1 \quad (2.2)$$

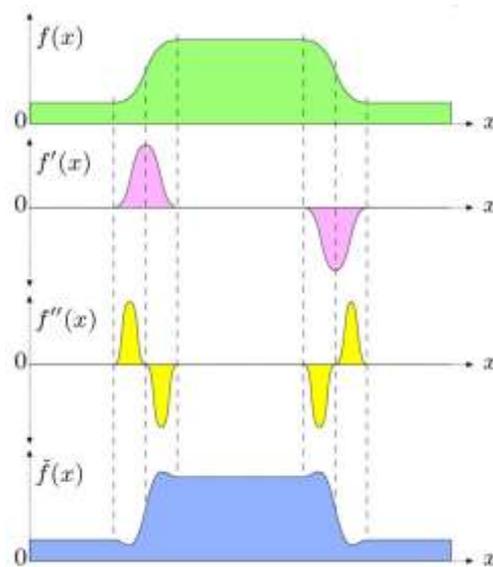
dengan h adalah respon impuls dan α adalah konstanta dengan nilai 0 hingga 1. Bentuk *Laplacian filter* yang sering digunakan ditunjukkan oleh Persamaan 2.3 dan 2.4.

Laplacian 1:

$$h = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad (2.3)$$

Laplacian 2:

$$h = \begin{bmatrix} 1 & 1 & 1 \\ 1 & -8 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad (2.4)$$



Gambar 2.8 Laplacian Filter

Properti *zero-crossing* dari *Laplacian filter* dapat digunakan untuk melokasi tepian pada gambar. *Zero-crossing* merupakan titik pada fungsi matematika yang ditandai dengan adanya perpotongan garis fungsi dengan *axis* (nilai 0). Contoh hasil penerapan *Laplacian filter* pada suatu citra ditunjukkan oleh Gambar 2.9.

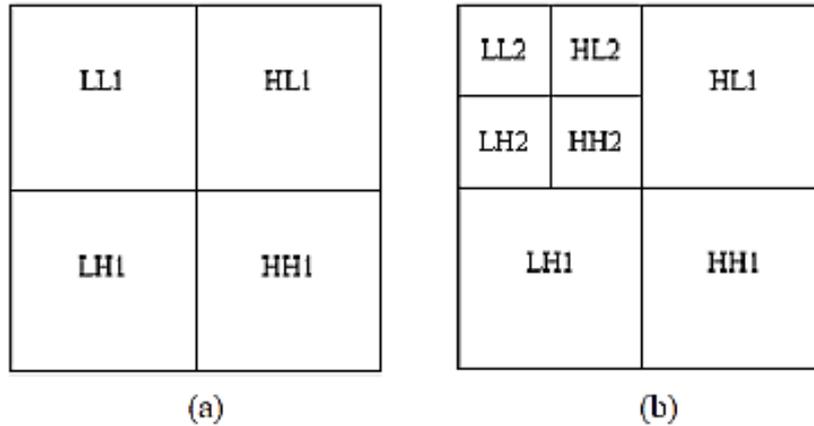
2.6. Discrete Wavelet Transform (DWT)

Transformasi wavelet adalah teknik matematis untuk menyintesis sinyal dengan menggunakan dilasi dan translasi dari fungsi wavelet. Dilasi dan translasi merupakan properti dasar dari wavelet yang menentukan bentuk wavelet yang dihasilkan dari sebuah fungsi wavelet. Fungsi wavelet $\Psi(t)$ adalah gelombang kecil yang beresilasi terhadap frekuensi [16]. Wavelet dapat dianggap sebagai *framework* yang dapat menguraikan sinyal menjadi komponen-komponen penyusunnya dalam berbagai skala. Dalam prosesnya, wavelet akan dikonvolusi dengan sinyal untuk memperoleh hasil dekomposisi dari sinyal. Nilai fungsi waktu kontinu atau kompleks $\Psi(t)$ terintegrasi ke nol. Fungsi ini disebut wavelet karena bentuknya yang beresilasi atau bergelombang [17].

wavelet.



Gambar 2.9 Deteksi tepi menggunakan *Laplacian Filter* [6]



Gambar 2.10 Sub-band LL, LH, HL dan HH dari sebuah gambar

Transformasi wavelet dapat bersifat kontinu atau diskrit sehingga dapat diklasifikasikan sebagai *Continuous Wavelet Transform (CWT)* atau *Discrete Wavelet Transform (DWT)*.

$$f(t) = \sum_{k=-\infty}^{\infty} \sum_{l=-\infty}^{\infty} d(k, l) 2^{-\frac{k}{2}} \psi(2^{-k}t - l) \quad (2.5)$$

Persamaan 2.5 menggunakan nilai diskrit untuk dilatasi dan translasi. Dilatasi mengambil nilai dari bentuk $a = 2^k$, yang mana k adalah bilangan bulat. Pada setiap dilatasi 2^k , parameter translasi mengambil nilai dari bentuk 2^{kl} dimana l adalah bilangan bulat. Urutan dua dimensi $d(k, l)$ umumnya disebut sebagai *Discrete Wavelet Transform (DWT)* [18]. *Discrete Wavelet Transform* adalah teknik terkomputerisasi untuk menghitung *fast wavelet transform* dari sinyal. DWT adalah solusi optimal untuk permasalahan waktu komputasi dan lebih mudah dalam pengoperasian serta implementasi. DWT dapat digunakan untuk pemrosesan citra dengan resolusi citra tinggi yang akan membutuhkan memori yang banyak pada *disk*.

Dalam transformasi wavelet, sinyal dalam domain waktu didekomposisi dengan melewatkannya melalui filter *high pass* dan *low pass*. Untuk menghasilkan dua versi yang berbeda, yaitu versi *low pass* (LL) dan versi *high pass* (HH). Versi *low pass* dapat didekomposisi lebih lanjut dengan mengirimkannya ke grup filter *low pass* (LL) dan *high pass* (LH). Demikian pula versi *high pass* dapat didekomposisi lebih lanjut dengan mengirimkannya ke grup filter *low pass* (HL) dan *high pass* (HL). Proses ini dapat dilanjutkan sampai sinyal yang diberikan didekomposisi ke level referensi yang telah ditentukan. Kemudian dalam dekomposisi mendalam akan menghasilkan sekelompok sinyal yang menentukan *frequency bands* yang berbeda tetapi secara total mewakili sinyal tunggal. Kemudian sinyal yang sesuai dengan *bands* yang berbeda dapat disatukan dengan diplot pada grafik tiga dimensi.

Dengan menerapkan DWT pada citra, citra tersebut diuraikan menjadi empat *sub-band* seperti pada Gambar 2.10 (a). *sub-band* LH1 (*Low-High*), HL1 (*High-Low*) dan HH1 (*High-High*) mewakili koefisien wavelet skala terbaik atau disebut juga dengan citra detail [19]. Sedangkan sub-band LL1 (*Low-Low*) berhubungan dengan koefisien level kasar, atau disebut juga dengan citra aproksimasi. Untuk mendapatkan tingkat koefisien wavelet kasar berikutnya, sub-band LL1 selanjutnya didekomposisi dan disampel. Kemudian ini menghasilkan dekomposisi wavelet level dua seperti yang ditunjukkan pada Gambar 2.10 (b). Untuk mendapatkan dekomposisi selanjutnya, LL2 akan digunakan. Proses ini berlanjut hingga skala akhir tercapai [19]. Gambar 2.11 menampilkan hasil dekomposisi citra level 1.



Gambar 2.11 Ilustrasi DWT

Ada sejumlah fungsi dasar yang dapat digunakan sebagai *mother wavelet* untuk transformasi wavelet. Karena *mother wavelet* menghasilkan semua fungsi wavelet yang digunakan dalam transformasi melalui translasi dan *scaling*, *mother wavelet* menentukan karakteristik dari transformasi wavelet yang dihasilkan. Detail dari pengaplikasian transformasi wavelet harus diperhitungkan saat memilih *mother wavelet* agar transformasi wavelet dapat digunakan secara efektif [18].

a. Haar Wavelet

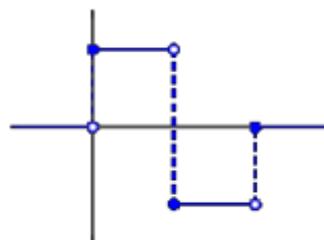
Ditemukan oleh adalah ahli matematika Alfred Haar, merupakan wavelet pertama dan paling sering digunakan. Haar wavelet adalah urutan tertentu dari fungsi berbentuk persegi yang disusun kembali dan membentuk keluarga wavelet atau basis. Kekurangan dari segi teknisnya adalah Haar wavelet tidak dapat dibedakan [19]. Haar wavelet ditunjukkan pada Gambar 2.12 [16].

b. Daubechies Wavelet

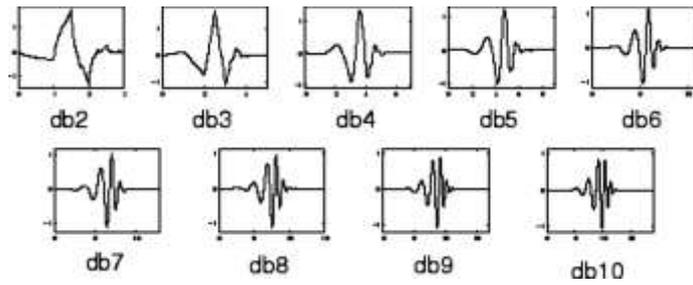
Ingrid Daubechies menemukan *compactly supported orthonormal wavelets*, yang merupakan salah satu penemuan paling penting dalam dunia penelitian wavelet sehingga membuat analisis DWT dapat dilakukan. Wavelet dalam keluarga Daubechies ditulis sebagai dbN , dimana N adalah skala dekomposisinya. Daubechies merupakan wavelet yang bersifat ortogonal[20].

c. Coiflet Wavelet

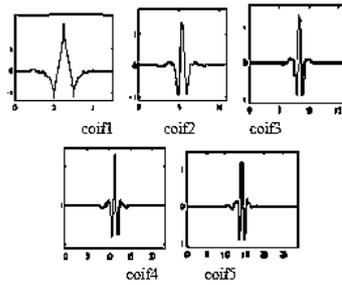
Fungsi wavelet ini memiliki momen $2N$ sama dengan 0 dan fungsi skala memiliki momen $2N-1$ sama dengan 0. Kedua fungsi memiliki dukungan panjang $6N-1$. Coiflet wavelet memiliki grafik yang hampir simetris. Wavelet ini mirip dengan Daubechies wavelet, yaitu memiliki *vanishing moments* yang besar [20]. Coiflet wavelet ditunjukkan pada Gambar 2.13 [16].



Gambar 2.12 Haar wavelet [16]



Gambar 2.14 Daubechies wavelet [16]



Gambar 2.13 Coiflet wavelet [16]

d. Symlet Wavelet

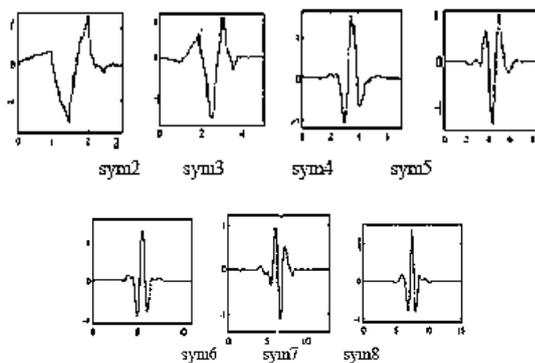
Symlet wavelet adalah “*symmetrical wavelet*”. Wavelet ini dirancang sehingga wavelet ini memiliki sifat asimetris dan jumlah maksimum *vanishing moments* yang paling sedikit [16]. Symlet wavelet ditunjukkan pada Gambar 2.15 [16].

e. Discrete Meyer

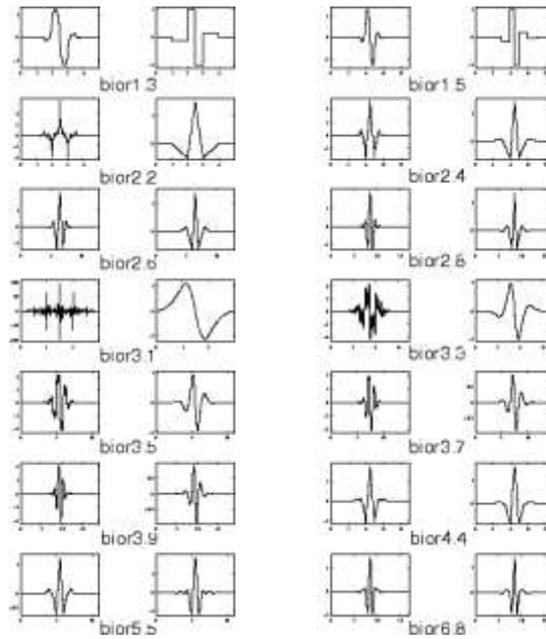
Meyer wavelet ditemukan oleh Yves Meyer. Meyer wavelet adalah jenis wavelet orthogonal dengan *compact support* [16]. Meyer wavelet ditunjukkan pada Gambar 2.16.

f. Biorthogonal

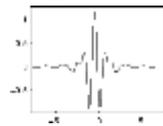
Pada biorthogonal wavelet, terdapat 2 skala yang menunjukkan skala dekomposisi dan rekonstruksi. Penerapan biorthogonal wavelet memberikan kebebasan lebih baik dalam pengaturan skala daripada wavelet dengan 1 skala. Biorthogonal wavelet ditunjukkan pada Gambar 2.17. Pada gambar dapat dilihat bahwa setiap *mother wavelet* memiliki 2 wavelet, yaitu wavelet dekomposisi(kiri) dan wavelet rekonstruksi(kanan).



Gambar 2.15 Symlet wavelet [16]



Gambar 2.17 Wavelet biorthogonal



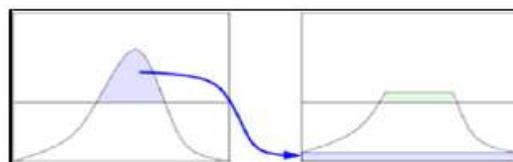
Gambar 2.16 Discrete Meyer wavelet

g. Reverse biorthogonal

Konsep dari wavelet reverse biorthogonal mirip dengan wavelet biorthogonal, namun jika pada biorthogonal wavelet skala dekomposisi ditentukan lebih dahulu dari skala rekonstruksi, pada reverse biorthogonal adalah sebaliknya.

2.7. Contrast Limited Adaptive Histogram Equalization (CLAHE)

Contrast Limited Adaptive Histogram Equalization (CLAHE) berbeda dari *adaptive histogram equalization* umum terutama pada pembatasan kontrasnya. Fitur ini juga dapat diterapkan pada *histogram equalization* global, sehingga menimbulkan *Contrast Limited Histogram Equalization (CLAHE)*, yang mana jarang digunakan pada praktiknya. Dalam metode CLAHE, prosedur pembatasan kontras harus diterapkan untuk setiap lingkungan tempat fungsi transformasi diturunkan. CLAHE dikembangkan untuk mencegah amplifikasi *noise* yang berlebihan yang dapat disebabkan oleh *Adaptive Histogram Equalization (AHE)*. Hal ini dicapai dengan membatasi peningkatan kontras dari AHE. Amplifikasi kontras di sekitar nilai piksel yang diberikan oleh kemiringan *Cumulative Distribution Function (CDF)* dan karenanya untuk nilai histogram pada nilai piksel tersebut.



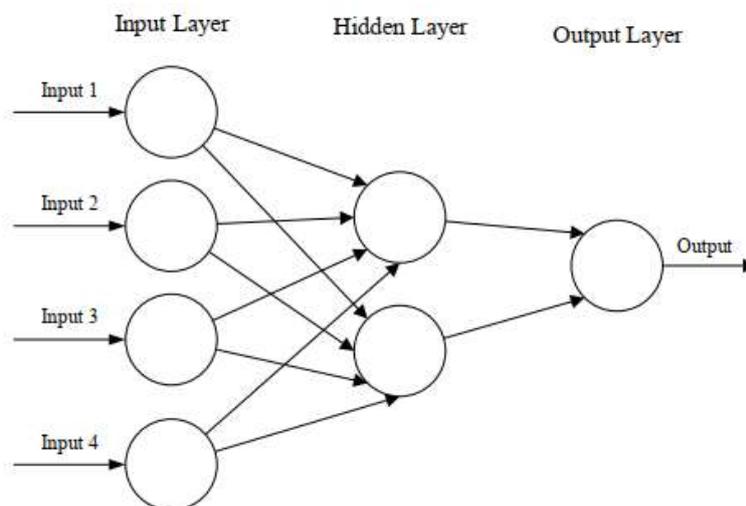
Gambar 2.18 Konsep dari *Contrast Limited Adaptive Histogram Equalization* [6]

CLAHE membatasi amplifikasi dengan memotong histogram pada nilai yang telah ditentukan sebelum menghitung CDF. Nilai dimana histogram dipotong, yang disebut batas klip, tergantung pada normalisasi histogram dan dengan demikian pada ukuran *neighborhood region*. Nilai umum membatasi yang membatasi amplifikasi yang dihasilkan adalah antara 3 dan 4. Kemudian bagian histogram yang melebihi batas klip tidak dibuang, tetapi didistribusikan secara merata di antara semua *histogram bins* (Gambar 2.18). Redistribusi akan mendorong beberapa *bins* melebihi batas klip lagi (bagian yang diarsir hijau pada Gambar 2.18), menghasilkan batas klip efektif yang lebih besar dari batas yang ditentukan dan nilai eksak tergantung pada gambar. Jika masih belum sesuai yang diinginkan, prosedur redistribusi dapat dilakukan secara berulang sampai kelebihanannya dapat diabaikan.

2.8. Convolutional Neural Network (CNN)

Artificial Intelligence (AI) adalah bagian dari ilmu komputer yang mempelajari bagaimana mesin dapat melakukan pekerjaan yang dilakukan oleh manusia. Salah satu metode pendekatan yang digunakan untuk melatih mesin agar dapat melakukan hal tersebut adalah *Machine Learning*. Pada metode *machine learning*, mesin dilatih dengan menggunakan data input melalui proses yang disebut dengan *learning process*. Terdapat berbagai teknik yang mampu menyelesaikan proses *learning* ini, salah satunya adalah dengan *Artificial Neural Network* (ANN). ANN adalah sistem komputasi yang terinspirasi dari sistem saraf biologis dari otak manusia[21]. ANN terbentuk dari banyak *node* yang terkoneksi satu sama lain yang disebut juga dengan neuron. Struktur dasar ANN ditunjukkan pada Gambar 2.19[21].

Convolutional Neural Network (CNN) adalah salah satu metode ANN yang sering digunakan dalam menyelesaikan permasalahan *pattern recognition* termasuk dalam pengolahan citra. Seperti namanya, CNN melakukan komputasi dengan metode konvolusi antara input dengan filter tertentu untuk mendapatkan *feature map* yang akan mendeteksi fitur dari input. Filter atau kernel pada konvolusi CNN terbentuk dari sejumlah *weight* yang tersusun dalam *array* 2 dimensi. Filter ini diaplikasikan pada semua data input dengan melakukan *shifting* di sepanjang data. Perbedaan antara ANN dan CNN dapat dilihat pada Tabel 2.3.

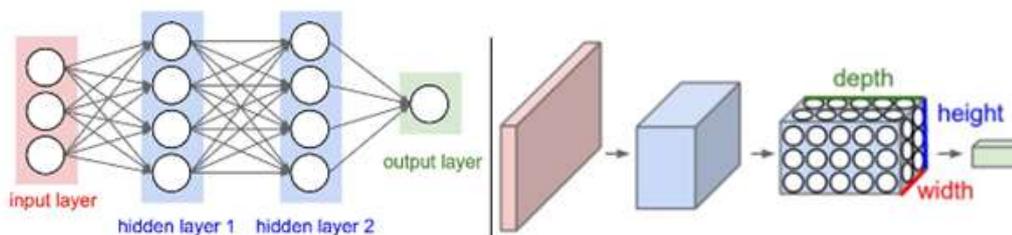


Gambar 2.19 Struktur dasar ANN

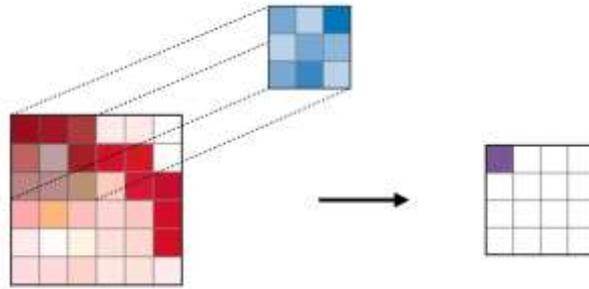
Tabel 2.3 Perbedaan ANN dan CNN

<i>Artificial Neural Network(ANN)</i>	<i>Convolutional Neural Network(CNN)</i>
Menggunakan <i>Fully Connected Layer</i>	Menggunakan <i>partially connected layer</i> dengan ukuran <i>connection</i> ditentukan oleh ukuran filter. <i>Fully connected layer</i> hanya diaplikasikan di <i>layer</i> terakhir
Hanya dapat digunakan untuk citra berukuran kecil	Dapat digunakan pada citra dengan berbagai ukuran
Jumlah parameter sangat banyak, contohnya pada citra dengan ukuran 28x28 dan <i>layer</i> dengan 32 neuron, setiap <i>layer</i> akan memiliki $28*28*32=25.088$ parameter sehingga komputasinya membutuhkan waktu yang lama	Jumlah parameter lebih sedikit dari ANN, contohnya pada citra dengan ukuran 28x28 dan <i>layer</i> dengan 32 filter berukuran 5x5, setiap <i>layer</i> akan memiliki $(5*5*1+1)*32=832$ sehingga komputasinya lebih cepat
Tidak efisien pada data input berupa citra	Efisien pada data input berupa citra
Tidak mempelajari <i>pattern</i> dalam <i>spatial hierarchy</i> (<i>output</i> dari <i>layer</i> sebelumnya tidak dipelajari)	Mempelajari <i>pattern</i> dalam <i>spatial hierarchy</i> sehingga lebih efektif dalam <i>pattern recognition</i>
Tidak <i>translation invariant</i> . Jika ANN sudah mempelajari <i>pattern</i> pada suatu lokasi, <i>recognizing</i> hanya dapat dilakukan pada lokasi tersebut.	<i>Translation invariant</i> . Jika CNN sudah mempelajari <i>pattern</i> pada suatu lokasi, <i>recognizing</i> dapat dilakukan pada lokasi-lokasi lainnya.

Gambar 2.20 memperlihatkan perbandingan arsitektur antara CNN dan *neural network* secara umum. Pada data berupa citra, metode CNN dapat berfungsi untuk melakukan segmentasi atau mendeteksi letak objek dan klasifikasi atau mengenali jenis objek pada gambar. Arsitekturnya dibentuk dengan menggabungkan blok-blok individual dengan tujuan-tujuan sendiri. Blok-blok ini disebut juga dengan *layer*. *Layer* pada CNN disusun dalam bentuk 3 dimensi yaitu *width*, *height*, dan *depth*. 3 jenis layer dasar yang membentuk CNN dapat dijelaskan sebagai berikut:



Gambar 2.20 Perbandingan antara ANN dan CNN [22]



Gambar 2.21 Ilustrasi konvolusi pada citra [23]

2.8.1. Convolutional Layer

Convolutional layer merupakan *building block* utama pada CNN yang melakukan mayoritas komputasi [22]. Konvolusi pada citra merupakan transformasi dari piksel ke piksel. Jenis transformasi yang digunakan didefinisikan sebagai kumpulan *weights* atau disebut juga dengan filter atau kernel. *Convolution layer* akan mengkomputasi *output* dari neuron yang terhubung ke region lokal dari input (*patch*) dengan cara melakukan konvolusi antara *weight* dengan region tersebut seperti ilustrasi pada Gambar 2.21 [23].

Proses dari transformasi piksel dijelaskan pada Persamaan 2.6 dengan w adalah kumpulan *weights* dan x merupakan kumpulan dari *pixel input* dan b merupakan nilai *bias*.

$$y = \sum_{i=0}^n w_i x_i + b \quad (2.6)$$

Filter yang terdapat pada *convolutional layer* ini memiliki ukuran kernel matriks yang kecil secara spasial namun dapat meluas hingga pada keseluruhan kedalaman pada volume *input*. Selain proses konvolusi, pada *convolution layer* juga terdapat penambahan fungsi aktivasi dengan jenis ReLU (*Rectified Linear Unit*) yang akan membuat proses *training* menjadi lebih cepat dan lebih efektif dengan cara memetakan nilai negatif menjadi 0 dan mempertahankan nilai positif. Bagian ini disebut juga dengan aktivasi karena hanya fitur aktif saja yang akan dapat diteruskan ke *layer* selanjutnya.

Terdapat 3 parameter yang mengatur *output volume* pada *layer*, yaitu *depth*, *stride*, dan *zero-padding*. *Depth* merupakan ukuran dari input yang digunakan. *Stride* adalah parameter yang menentukan jumlah pergeseran filter. Misal, jika digunakan nilai *stride* 1 maka *convolution filter* akan bergeser sebanyak 1 piksel secara horizontal dan vertikal. Semakin kecil nilai *stride* yang digunakan akan semakin detail informasi yang didapatkan dari sebuah input namun membutuhkan komputasi yang lebih besar. Namun dengan menggunakan nilai *stride* yang kecil tidak akan selalu mendapatkan performa yang baik. *Zero padding* adalah parameter yang menentukan jumlah piksel berisi 0 yang ditambahkan pada setiap sisi dari *input*. Hal ini bertujuan untuk memanipulasi dimensi *output* dari *convolution layer*. Tujuan dari penggunaan *padding* adalah untuk meningkatkan performa dari model karena *convolution layer* akan terfokus pada informasi yang sebenarnya yaitu yang berada diantara *zero padding*. Selain itu, dimensi *output* dari *convolution layer* selalu lebih kecil dari *input*. *Output* ini akan digunakan kembali sebagai *input layer* selanjutnya sehingga semakin banyak informasi yang terbuang. Dengan menambahkan *padding* maka akan dapat mengatur dimensi *output* agar tidak berkurang secara drastis. Persamaan 2.7 digunakan untuk menentukan dimensi dari *feature map* yang ingin dicapai,

$$\text{Dimensi} = \frac{W - N + 2P}{S} + 1 \quad (2.7)$$

dengan W adalah panjang atau tinggi input, N adalah panjang atau tinggi filter, P adalah *zero padding* dan S adalah *stride*.

2.8.2. Pooling Layer

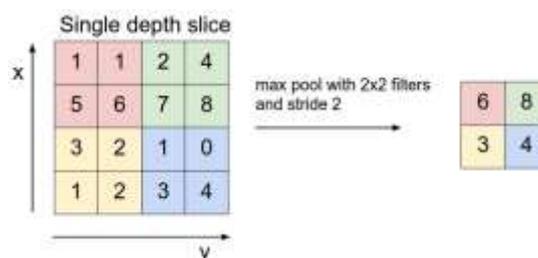
Convolutional layer biasanya diikuti dengan adanya *pooling layer*. *Pooling layer* terdiri dari sebuah *filter* dengan ukuran dan *stride* tertentu yang akan bergeser pada seluruh area *feature map*. Tujuan dari penggunaan *pooling layer* adalah mengurangi dimensi dari *feature map* (*downsampling*) sehingga mempercepat komputasi karena parameter yang perlu diperbaharui akan semakin sedikit dan dapat mengatasi *overfitting* [22]. Jenis *pooling* yang biasa digunakan adalah *max pooling* dan *average pooling*. *Max pooling* akan mengambil nilai paling maksimal sesuai dengan nilai *stride* yang digunakan. Sedangkan, *average pooling* akan mengambil nilai rata-rata. *Max pooling* memiliki fungsi untuk mempertajam bagian *edge* pada gambar, sedangkan *average pooling* berfungsi untuk memperhalus gambar. Sehingga, pada gambar *grayscale* biasanya digunakan *max pooling*. Ilustrasi dari penggunaan *max pooling* dapat dilihat pada Gambar 2.22.

2.8.3. Fully-Connected Layer

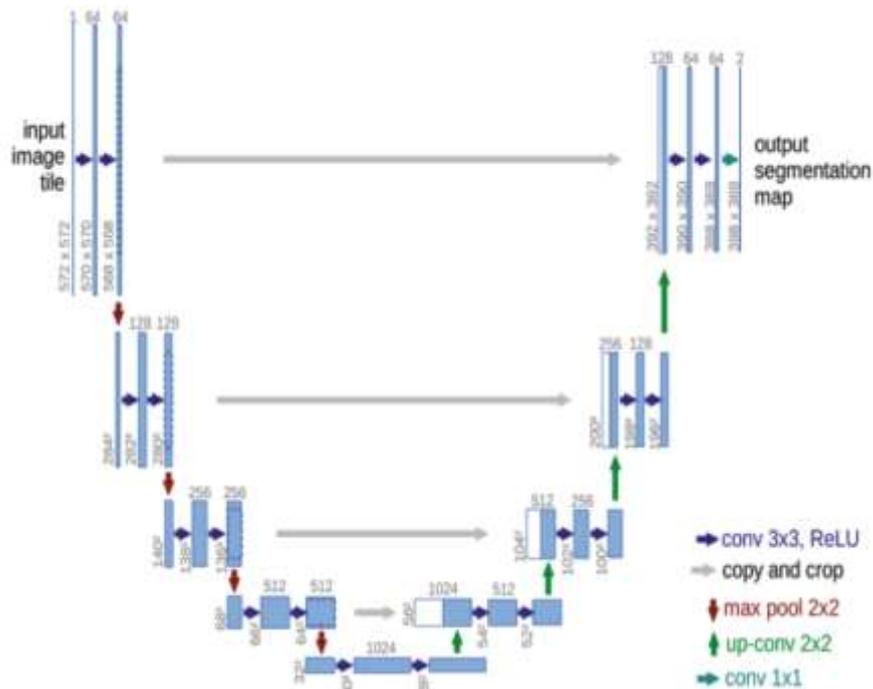
Fully-connected layer merupakan *layer* yang bekerja seperti ANN dan memiliki beberapa *hidden layer*, *activation function*, *output layer* dan *loss function*. *Feature map* terakhir yang dihasilkan sebelum *fully-connected layer* memiliki bentuk *multidimensional array* sehingga perlu dilakukan *flatten* atau *reshape feature map* menjadi sebuah vektor 1 dimensi sehingga dapat digunakan sebagai *input layer* ini. Tujuan dari *fully-connected layer* ini untuk melakukan transformasi pada dimensi data agar dapat diklasifikasikan secara *linear*.

2.9. Arsitektur U-Net

CNN biasanya digunakan untuk melakukan klasifikasi yang outputnya merupakan sebuah label kelas tunggal. Namun, output yang diharapkan dapat berupa lokalisasi, yaitu sebuah label kelas diberikan pada setiap piksel [24]. Salah satu arsitektur CNN yang dapat digunakan untuk proses ini adalah arsitektur U-Net seperti pada Gambar 2.23. Arsitektur ini melengkapi bagian *contracting layer* dengan *expansive layer*, dimana *pooling operator* digantikan oleh *upsampling operator* sehingga *layer* ini meningkatkan resolusi dari *output*. Untuk melakukan lokalisasi, *high resolution features* dari *contracting path* dikombinasikan dengan *output* yang telah mengalami *upsampling*. *Succesive convolution layer* nantinya dapat belajar untuk memberikan *output* yang lebih presisi berdasarkan informasi ini [24].



Gambar 2.22 Ilustrasi max pooling



Gambar 2.23 Arsitektur U-Net

Pada arsitektur ini, *upsampling part* juga memiliki jumlah *feature channel* yang besar, sehingga membuat *network* dapat mempropagasi informasi konteks ke *layer* dengan resolusi yang lebih tinggi. Hal ini membuat *expansive path* lebih kurang simetris dengan *contracting path* sehingga dihasilkan arsitektur dengan bentuk U. Arsitektur ini tidak memiliki *fully connected layers* dan hanya menggunakan *valid part* dari setiap konvolusi sehingga memungkinkan segmentasi yang baik dari citra berukuran besar. Untuk memprediksi batas pada citra, *missing context* diekstrapolasikan dengan melakukan pencerminan dari citra *input*. Strategi ini penting untuk pengaplikasian *network* pada citra yang besar karena limitasi dari memori GPU [24].

Untuk kasus dengan data *training* yang sangat sedikit, dilakukan augmentasi data dengan menggunakan deformasi elastis pada citra. Hal ini memungkinkan *network* untuk mempelajari invariansi pada deformasi tersebut tanpa melihat transformasi ini dalam citra anotasi. Hal ini penting untuk segmentasi citra biomedik karena deformasi merupakan variasi yang paling sering ditemukan pada jaringan. *Network* dengan arsitektur U-Net ini dapat diaplikasikan pada berbagai masalah segmentasi citra biomedis [24].

Arsitektur U-Net seperti yang terlihat pada Gambar 2.23 terdiri dari *contracting path* (bagian kiri) dan *expansive path* (bagian kanan). *Contracting path* terdiri dari 2 konvolusi 3x3 (*unpadded convolutions*) dimana setiap konvolusi diikuti oleh sebuah *Rectified Linear Unit* (ReLU) dan sebuah *max pooling operation* 2x2 dengan 2 *stride* untuk *downsampling*. Pada setiap *downsampling*, jumlah *feature channels* digandakan. Setiap *step* pada *expansive path* terdiri dari *upsampling* dari *feature map* yang diikuti oleh konvolusi 2x2 (*up-convolution*) yang membagi 2 jumlah *feature channel*, *concatenation* dengan *cropped feature channel* terhadap *contracting path*, dan 2 konvolusi 3x3 yang diikuti oleh ReLU. *Cropping* perlu dilakukan karena pada setiap konvolusi terjadi *loss of border*. Pada *layer* terakhir, konvolusi 1x1

digunakan untuk memetakan ke-64 komponen *feature vector* pada jumlah kelas yang diinginkan. Sehingga total *network* pada *convolutional layer* adalah 23 [24].

Citra masukan dan peta segmentasi diperlukan untuk melatih *network*. *Weight map* untuk setiap segmentasi *grond truth* dikomputasi untuk mengimbangi perubahan frekuensi piksel dari kelas tertentu pada *training dataset* dan untuk membuat *network* dapat mempelajari *small separation border*.

Pada *network* dengan banyak *convolutional layer path* yang berbeda, inisialisasi *weight* yang tepat sangat penting untuk menghindari aktivasi berlebihan. Untuk arsitektur ini hal ini dapat dilakukan dengan menggambarkan *initial weight* dari distribusi Gaussian dengan standar deviasi $\sqrt{2/N}$ dengan N merupakan jumlah *node* yang masuk pada setiap neuron [24].

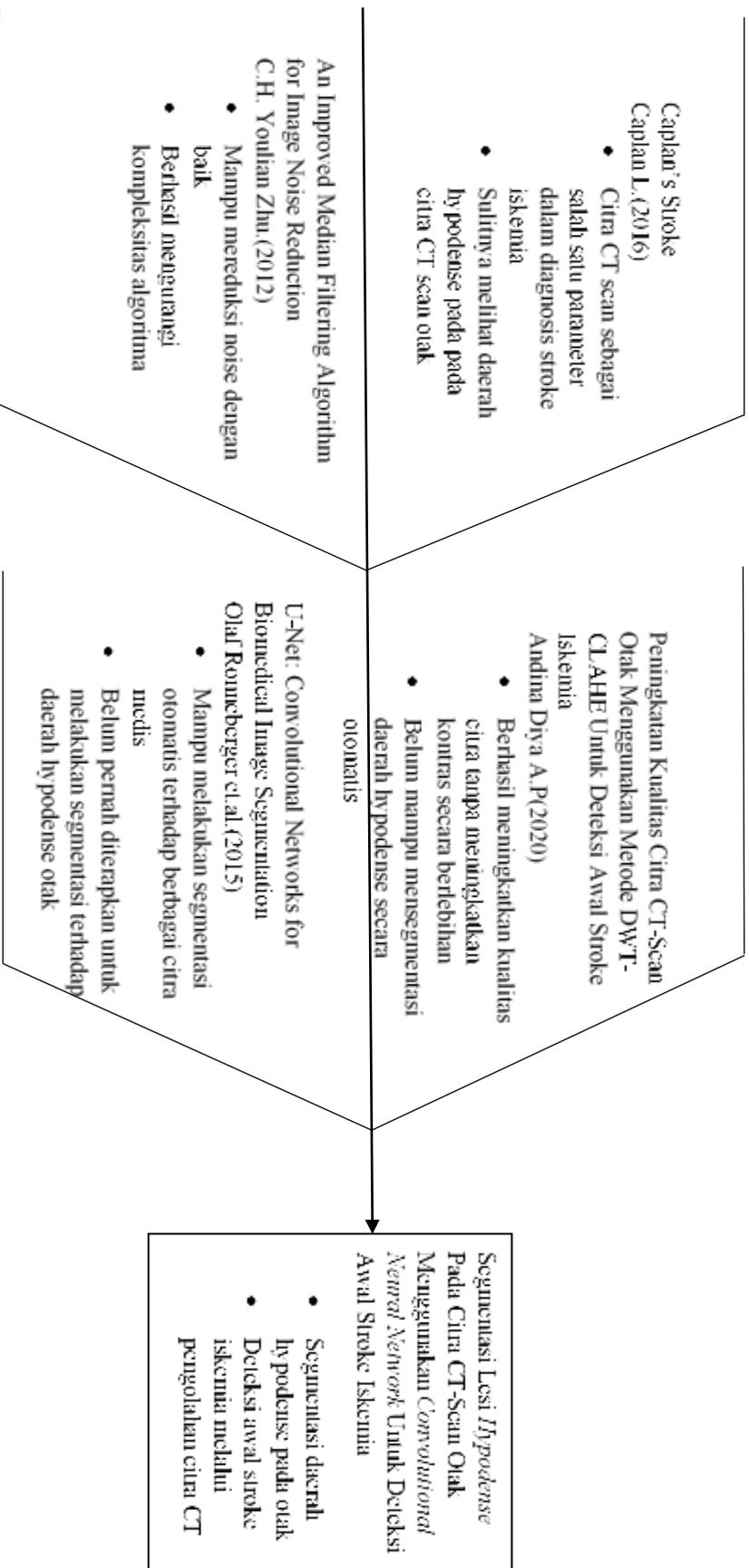
2.10. Rasional

Secara umum, rasional dalam penelitian ini digambarkan dalam *fish bone diagram* pada Gambar 2.23. Stroke mengacu pada kerusakan otak atau sumsum tulang belakang yang disebabkan oleh kelainan dalam suplai darah, seperti tersumbat atau yang disebut stroke iskemia atau pecah yang disebut dengan stroke hemoragik. Pendeteksian awal stroke iskemia sering menggunakan CT scan untuk memperoleh citra otak. Hal ini dikarenakan beberapa kelebihan yang dimiliki oleh CT jika dibandingkan dengan modalitas lainnya, seperti ketersediaan, biaya, dan waktu. Ketika stroke merupakan stroke iskemik, CT dapat menunjukkan *infarct* (cedera) sebagai lesi *hypodense* atau dapat pula pada awalnya tetap normal. Perubahan awal pada CT scan bisa sangat halus atau tidak terlalu terlihat dalam beberapa jam pertama timbulnya gejala. Dibalik kelebihan CT scan, terdapat kekurangan yaitu ketika dilakukan *scanning* awal, daerah *hypodense* pada otak pasien tidak bisa terlihat secara jelas. Daerah *hypodense* akan terlihat ketika dilakukan *scanning* berikutnya sekitar 1-2 jam setelah serangan stroke pertama kali. Ini dijelaskan oleh Caplan L. dalam bukunya “*Caplan’s Stroke*”.

Karena kesulitan dalam pendeteksian awal stroke iskemia ini, ada kebutuhan untuk metode yang dapat secara tepat mendeteksi tanda-tanda iskemia dini sesegera mungkin untuk memberikan *treatment* yang tepat [7]. Pendeteksian melalui segmentasi citra telah terbukti cukup baik dalam menentukan lokasi keabnormalan pada suatu citra. Untuk dapat melakukan segmentasi terhadap citra, sebelumnya perlu dilakukan *preprocessing* yang bertujuan untuk menghilangkan *noise* dan membatasi *Region of Interest*. Terdapat berbagai metode yang telah dikembangkan untuk melakukan *preprocessing*, diantaranya menggunakan filter median dan filter Laplacian. Dalam paper berjudul “*An Improved Median Filtering Algorithm for Image Noise Reduction*” hasil penelitian dari C.H. Youlian Zhu, metode filter median dijelaskan dengan cukup baik. Dari hasil penelitiannya, diperoleh kesimpulan bahwa metode ini mampu mereduksi *noise* dengan baik dan berhasil mengurangi kompleksitas algoritma. Setelah *preprocessing*, selanjutnya dilakukan peningkatan yang bertujuan untuk meningkatkan kontras citra. Citra otak merupakan salah satu citra yang cukup sensitif terhadap peningkatan kontras, sehingga apabila kontras ditingkatkan secara berlebihan, citra dapat menjadi rusak dan menghilangkan informasi penting yang terkandung didalamnya. Untuk mengatasi hal tersebut, terdapat sebuah metode yang menggabungkan antara *Discrete Wavelet Transform* dengan *Contrast Limited Adaptive Histogram Equalization* yang terbukti dapat meningkatkan kontras

otak dengan tidak berlebihan, sehingga lesi *hypodense* masih dapat didiferensiasi dengan jaringan lunak lainnya seperti yang diterapkan oleh Andina Diya A.P dkk dalam penelitiannya yang berjudul “*Peningkatan Kualitas Citra CT scan Otak Menggunakan Metode DWT-CLAHE untuk Deteksi Awal Stroke Iskemia*”. Metode yang diajukan ini berhasil meningkatkan kualitas citra tanpa meningkatkan kontras secara berlebihan. Pada penelitian ini juga dilakukan *thresholding-based segmentation* untuk mendeteksi daerah *hypodense*. Namun, belum mampu mendeteksi daerah *hypodense* yang ukurannya sangat kecil, dan belum mampu mensegmentasi secara jelas antara daerah *hypodense* dengan jaringan lunak normal dan ventrikel otak secara otomatis.

Salah satu metode yang cukup baik dalam melakukan segmentasi citra adalah *Convolutional Neural Network* dengan arsitektur U-Net, dimana citra dapat disegmentasi secara otomatis setelah CNN dilatih dengan menggunakan data pembelajaran yang telah diinputkan sebelumnya. Hal ini dijelaskan pada penelitian yang dilakukan oleh Olaf Ronneberger pada tahun 2015 yang berjudul “*U-Net: Convolutional Neural Network for Biomedical Image Segmentation*”. Metode pada penelitian ini mampu melakukan segmentasi otomatis terhadap citra medis, namun belum pernah diterapkan pada citra CT scan otak dari pasien stroke iskemia. Oleh sebab itu, pada penelitian ini akan dilakukan pengembangan terhadap pustaka-pustaka yang telah dijelaskan sebelumnya dengan melakukan segmentasi terhadap lesi *hypodense* otak dengan menggunakan CNN arsitektur U-Net.



Gambar 2.24 Fish bone diagram

BAB 3

METODE PENELITIAN

Pada penelitian ini dilakukan perancangan sistem yang mampu melakukan segmentasi pada lesi *hypodense* otak dengan metode *Convolutional Neural Network*. Kerangka kerja dari tugas akhir ini dapat dibagi menjadi 2 bagian utama yaitu: (1) pemrosesan awal pada *dataset* citra CT scan otak yang akan digunakan sebagai masukan model CNN serta bagian (2) perancangan CNN dengan arsitektur U-Net untuk mendeteksi *lesi hypodense*. Pada bab ini akan dipaparkan metode-metode yang akan digunakan dalam penelitian, meliputi desain sistem secara umum, *dataset* yang digunakan, *preprocessing* citra, pelatihan model CNN, dan pengujian hasil peningkatan dan segmentasi citra.

3.1. Diagram Blok Sistem

Pada penelitian ini diusulkan metode segmentasi lesi *hypodense* pada citra CT scan otak untuk deteksi awal stroke iskemia. Secara umum, alur kerja sistem yang akan dibuat dijelaskan dalam diagram blok pada Gambar 3.1. Pertama, data citra CT scan digunakan sebagai input dalam *preprocessing*. *Preprocessing* berfungsi untuk menghilangkan *noise* dan memperbaiki citra untuk mengoptimalkan kerja CNN sehingga dapat memberikan hasil segmentasi yang lebih baik. *Preprocessing* akan dibagi menjadi beberapa tahap. Tahap pertama adalah *grayscale* untuk mengubah citra RGB menjadi citra *grayscale* untuk memudahkan proses berikutnya. Selanjutnya dilakukan *filtering* terhadap citra dengan menggunakan filter median untuk menghilangkan *noise* yang terdapat pada citra dan filter morfologi untuk menghilangkan daerah-daerah diluar ROI serta menentukan batas *cropping* citra. Setelah itu dilakukan *autocropping* untuk memotong citra sesuai dengan ROI yang telah diperoleh melalui filter morfologi. Pada citra tersebut, terdapat bagian tulang dan otak sehingga dilakukan segmentasi terhadap bagian otak dengan menerapkan filter Laplacian untuk mencari batasnya kemudian dilakukan *thresholding* untuk mendapatkan bagian otak dari ROI tersebut. Bagian terakhir dari *preprocessing* adalah proses peningkatan citra (*image enhancement*) dengan metode DWT-CLAHE untuk meningkatkan kontras antara lesi *hypodense* dan jaringan lainnya. Keluaran dari blok *preprocessing* akan diaugmentasi atau diperbanyak dengan melakukan transformasi geometrik pada citra. Hasil augmentasi data selanjutnya digunakan sebagai *input* dalam melakukan pembelajaran untuk perancangan sistem CNN dengan arsitektur U-Net. Sedangkan untuk *desired output* akan dilakukan segmentasi manual terhadap citra awal dengan mengacu pada anotasi *ground truth* yang telah dideskripsikan pada *dataset*.

Pengujian performansi dilakukan dengan mengamati *loss* dan *Intersection of Union* saat pelatihan. Model *weight* akhir yang dihasilkan digunakan untuk melakukan segmentasi bagian lesi *hypodense* sehingga menghasilkan keluaran berupa gambar hasil prediksi. Hasil gambar tersebut dilakukan pengujian akhir dengan memanfaatkan *Dice Similarity Coefficient* (DSC) untuk menilai keberhasilan model dalam melakukan segmentasi dengan membandingkan *pixel overlap* pada gambar (hasil *predicted* dan *ground truth*).

Subbab 3.2 akan menjelaskan *dataset* yang digunakan untuk melatih dan menguji sistem secara umum. Subbab 3.3 menjelaskan tahapan metode yang digunakan dalam melakukan *preprocessing*. Subbab 3.4 akan menjelaskan proses penyiapan *dataset* yang digunakan sebagai

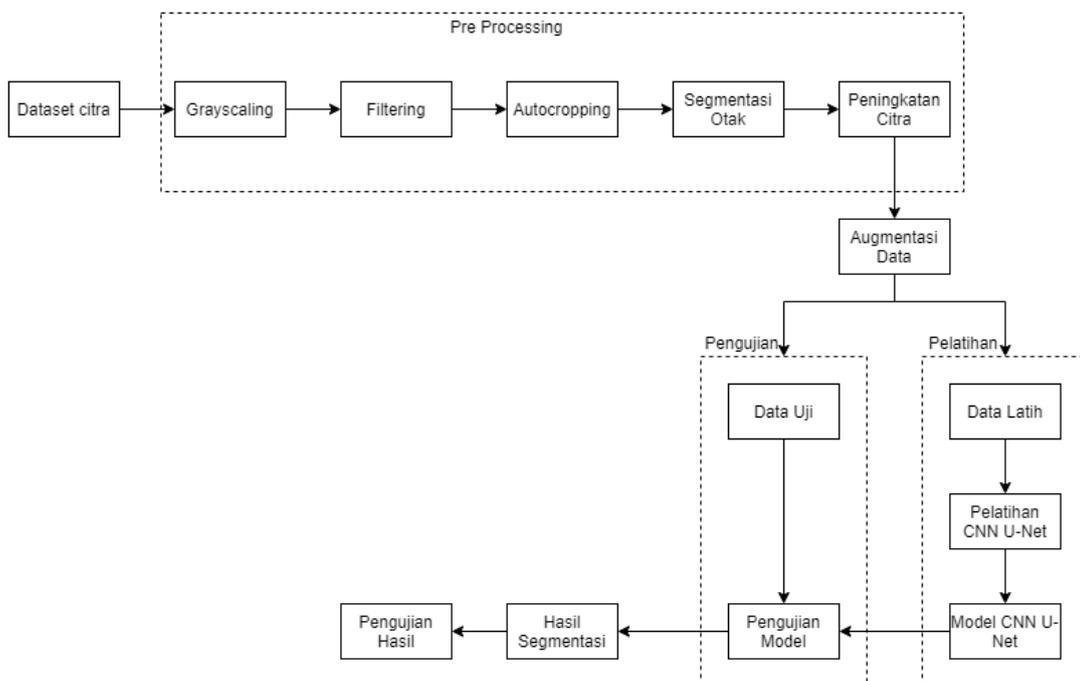
input sistem dan proses perancangan CNN dengan arsitektur U-Net untuk dapat melakukan segmentasi lesi *hypodense* dan terakhir pada subbab 3.5 menjelaskan tentang metode pengujian yang digunakan dalam melakukan validasi keluaran dari sistem program secara umum.

3.2. Data

Data yang digunakan dalam penelitian ini merupakan data yang diperoleh secara *online* dari radiopaedia.org [25]. Radiopaedia merupakan sumber data radiologi yang bersifat *open-edit educational* dengan data yang berasal dari radiologis dan praktisi kesehatan di seluruh dunia. Pada Radiopaedia terdapat banyak data citra kasus-kasus radiologi termasuk stroke iskemia. Setiap kasus telah dianalisis dan ditulis deskripsinya oleh dokter dan radiologis yang membagikan kasus tersebut. Untuk penelitian ini, dataset merupakan hasil citra CT scan otak pasien stroke iskemia dari beberapa kasus meliputi *acute infarction, ischemic stroke, PCA territory infarct, pontine infarct, subacute left middle cerebral artery infarct, dan posterior cerebral artery territory infarct*. Data yang digunakan adalah 87 citra CT scan dari *axial plane* yang belum dilakukan peningkatan atau pengolahan citra. Informasi mengenai data secara lengkap dapat dilihat pada lampiran.

3.3. Preprocessing

Sebelum diproses lebih lanjut, data harus melalui *preprocessing* untuk memudahkan CNN dalam melakukan pembelajaran untuk mencapai nilai konvergensi *error* dan akan memengaruhi performa dari model arsitektur UNET yang dihasilkan. Langkah *preprocessing* ini akan menghasilkan keluaran yang dihasilkan dari sub langkah dengan tujuan masing-masing dan akan digunakan sebagai *input* untuk langkah selanjutnya. Tahap pada blok *preprocessing* ini adalah sebagai berikut:



Gambar 3.1 Diagram blok sistem

3.3.1 Grayscale

Grayscale adalah proses mengubah citra dengan sistem pewarnaan tertentu menjadi citra *grayscale*. Citra *grayscale* adalah citra dengan komponen warna berada pada nilai 0 sampai 255 dengan 0 menunjukkan warna hitam dan 255 menunjukkan warna putih. Proses *grayscale* ini penting dilakukan untuk mereduksi dimensi dan kompleksitas pengolahan citra. Citra pada dataset terbaca sebagai citra RGB dengan *array* [lebar, panjang, 3] dengan 3 adalah *channel red, green, dan blue*. Sehingga, untuk memudahkan pemrosesan pada penelitian ini, citra masukan diubah menjadi *grayscale* atau dua dimensi yang memiliki *array* [lebar, panjang].

3.3.2 Filter median dan morfologi

Untuk mengurangi *noise* yang terdapat pada citra, akan digunakan filter median dan morfologi. Pada proses *median filtering*, data yang digunakan untuk menghitung median terdiri dari kumpulan data yang ganjil. Hal ini disebabkan karena dengan jumlah data yang ganjil, maka piksel yang akan diproses berada di tengah. Pada median filter digunakan matriks *neighborhood* atau tetangga piksel yang berdimensi $N \times N$ dapat juga disebut sebagai kernel. Data pada kernel tersebut akan diurutkan nilainya. Setelah itu, nilai piksel yang baru dicari dengan menggunakan perhitungan median. Nilai median inilah yang nantinya akan menggantikan piksel yang berada di posisi tengah dari matriks *kernel*.

Filter morfologi dapat lebih mudah dilakukan pada citra biner, karena dapat dengan jelas ditentukan *foreground* dan *background*-nya. *Foreground* pada citra adalah objek yang biasanya berwarna putih, sedangkan *background* adalah latar belakang citra yang biasanya berwarna hitam. Oleh sebab itu, sebelum menerapkan filter morfologi, citra diubah terlebih dahulu menjadi citra biner dengan menggunakan metode *thresholding* dimana diberikan batas tertentu untuk menentukan sebuah piksel adalah *foreground* atau *background*. Pada filter morfologi digunakan metode *opening*, yaitu erosi yang diikuti dengan dilasi. Erosi dilakukan untuk menghilangkan daerah-daerah diluar ROI dengan cara mengubah pixel *foreground* pada batas(*edge*) menjadi *background*. Selanjutnya diterapkan metode dilasi untuk mengisi kembali bagian-bagian yang ikut hilang dalam penerapan metode erosi.

3.3.3 Autocropping

Autocropping adalah proses yang dilakukan untuk memotong citra pada bagian ROI yang telah dihasilkan dari tahapan sebelumnya untuk mengurangi dimensi citra sehingga proses pembelajaran CNN dapat dilakukan dengan lebih efektif. Proses ini diawali dengan menentukan batas terluar dari citra yang diperoleh dengan cara mencari nilai kontur maksimum dari hasil filter morfologi. Mode pengambilan kontur memakai mode pengambilan eksternal yang akan mengambil kontur bagian luar dari objek pada citra. Terdapat 2 mode pendekatan pada penentuan kontur, yaitu *none* dan *simple*.

3.3.4 Segmentasi Otak

Pada citra CT scan otak, secara umum terdapat 2 bagian utama, yaitu tengkorak dan otak. Kedua bagian ini memiliki batasan yang cukup baik karena intensitas tengkorak dan otak sangat berbeda. Untuk memaksimalkan kerja CNN agar fokus pada pembelajaran bagian otak, dilakukan segmentasi untuk menghilangkan bagian tengkorak dari citra. Ini dilakukan dengan menentukan batas yang diperoleh dengan menerapkan filter Laplacian pada citra dan kemudian

dilakukan *thresholding* terhadap daerah diluar batas sehingga diperoleh citra baru yang memiliki batas sesuai dengan Laplacian filter.

3.3.5 Peningkatan Citra

Proses peningkatan citra berfungsi untuk memperjelas batas antara lesi *hypodense* dan daerah disekelilingnya. Dengan melakukan peningkatan citra, kontras antara daerah *hypodense* dan jaringan normal dapat terlihat dengan lebih jelas. Pada penelitian ini, peningkatan citra dilakukan dengan menggunakan metode DWT-CLAHE. Metode ini dipilih karena telah terbukti mampu meningkatkan intensitas citra dengan tidak berlebihan sehingga tidak merusak informasi yang terkandung didalam citra. Citra yang dihasilkan dari proses sebelumnya akan didekomposisi menjadi frekuensi rendah dan frekuensi tinggi. Selanjutnya pada frekuensi rendah akan diterapkan metode CLAHE untuk meningkatkan kontrasnya, sedangkan pada frekuensi tinggi tidak diterapkan metode apapun. Selanjutnya kedua frekuensi ini direkonstruksi sehingga menghasilkan citra dengan kontras yang telah ditingkatkan. Karena ukuran citra dari data yang diperoleh sangat bervariasi, terakhir dilakukan *resize* pada citra sehingga keluarannya menjadi citra berukuran 512x512. Ini penting dilakukan karena CNN membutuhkan citra dengan ukuran yang sama dalam proses *learning*.

3.4. Perancangan CNN

Arsitektur yang digunakan pada penelitian ini merupakan modifikasi dari arsitektur U-Net yang diajukan oleh Olaf Ronneberger pada tahun 2015. Model arsitektur yang digunakan dapat dilihat pada Gambar 3.3.

Arsitektur CNN dengan U-Net secara umum terdiri dari *contracting path* dan *expansive path*. *Input* model adalah berupa citra dengan ukuran 512x512. Pada penelitian ini, ukuran citra sebelum *preprocessing* bervariasi antara 512x512 hingga 614x614, oleh sebab itu pada proses *preprocessing* dilakukan *resizing* untuk menyamakan ukuran dari data input.

Data input ini akan melalui proses *encoder* atau ekstraksi fitur untuk mendapatkan informasi fitur dari citra. Proses ekstraksi fitur dilakukan dengan mengkonvolusi citra input dengan filter tertentu untuk memperoleh fitur dari citra. Proses konvolusi akan meningkatkan kedalaman dengan cara melakukan peningkatan jumlah *channel* yang digunakan pada setiap proses *layer* dari sebuah gambar yang diikuti oleh *max pooling* sebagai *subsampling* dan berfungsi untuk mengurangi dimensi dari gambar.

Pada penelitian ini, untuk melakukan ekstraksi fitur pada *contracting path* digunakan *residual network*. *Residual network* merupakan arsitektur yang digunakan untuk mengatasi permasalahan *vanishing moment* pada CNN dengan jumlah *layer* yang banyak. *Vanishing moment* merupakan keadaan dimana proses pelatihan berhenti untuk mengupdate *weight model*. Ini dilakukan dengan menambahkan *skip connection* setelah *convolution layer*.

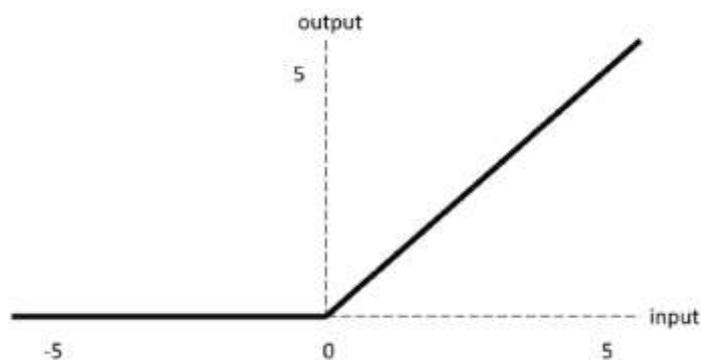
Pada arsitektur yang digunakan, setiap *convolution layer* didahului oleh *batch normalization*, *activation*, dan *zero padding*. *Batch normalization* dilakukan untuk membuat CNN lebih cepat dan stabil dengan melakukan normalisasi terhadap *layer* melalui proses *recentering* dan *rescaling* pada setiap *batch*. *Activation function* ditambahkan pada *neural network* untuk membantu *neural network* dalam mempelajari data yang kompleks. Dengan menerapkan *activation function*, *neural network* akan mempelajari data dengan bentuk *nonlinear*. *Activation* digunakan untuk menentukan apakah *neuron* pada CNN akan diaktifkan

atau tidak. Ini ditentukan berdasarkan transformasi yang dilakukan oleh *activation function* terhadap input untuk memperoleh *output* pada *range* tertentu. *Range* ini ditentukan oleh jenis *activation function* yang digunakan. Pada penelitian ini *activation function* yang digunakan adalah ReLU. ReLU (*Rectified Linear Unit*) adalah *activation function* sederhana yang akan mengubah *input* menjadi 0 apabila nilainya negatif dan mengambil nilai input itu sendiri sebagai *output* apabila nilainya positif. Bentuk *activation function* ReLU dapat dilihat pada Gambar 3.2.

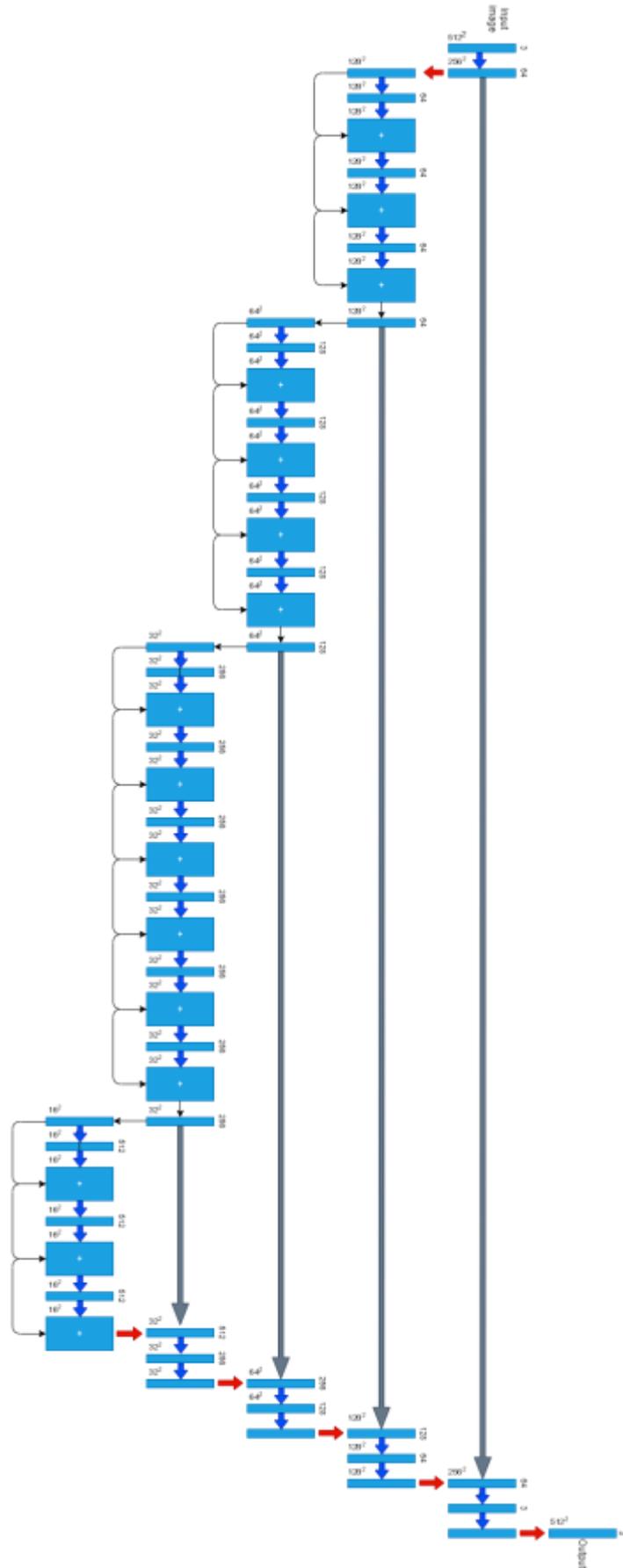
Hasil konvolusi citra awal berukuran 512x512 dengan kernel 3x3 dan *stride* atau pergeseran 2 adalah fitur berukuran 256x256x64. Hasil ini kemudian dilakukan *max pooling* untuk mengurangi jumlah fitur dengan mengambil nilai maksimum dari setiap kernel sehingga diperoleh hasil *max pooling* berukuran 128x128x64.

Selanjutnya dilakukan 4 *stage encoder* dengan *stage 1* terdiri dari 2 *convolution layer* dan 1 *skip connections* yang diulangi sebanyak 3 kali. Hasil dari 2 *convolution layer* dijumlahkan dengan *skip connections* sesuai dengan arsitektur Resnet. Ini juga dilakukan pada *stage 2, 3,* dan 4 dengan jumlah *convolution layer* pada *stage 2* adalah 8, pada *stage 3* adalah 12, dan *stage 4* adalah 6.

Hasil ekstraksi fitur dari *stage 4* selanjutnya akan masuk ke bagian *expansive path* untuk di-*upsampling*. *Expansive path* dilakukan untuk mengembalikan data citra menjadi ukuran asli dengan cara melakukan proses *upsampling*, sehingga didapatkan informasi dari lokasi fitur pada suatu gambar. Formula dari *expansive path* adalah melakukan *transpose convolution 2D*, *concatenate* dan 2 buah *convolution layer*. *Transposed convolution* merupakan teknik konvolusi atau dapat dikatakan proses dalam melakukan *upsampling* pada data sehingga akan memperluas ukuran dari gambar dan mendapatkan informasi mengenai lokasi dari sebuah objek yang disegmentasi pada gambar. *Input channel* pada *expansive path* adalah 512. *Concatenate* merupakan bagian untuk menggabungkan gambar pada *contracting path* dengan keluaran dari *layer* sebelumnya untuk mendapatkan prediksi yang presisi. Hingga langkah terakhir dilakukan *convolution layer* dengan *filter* ukuran 1x1 yang merupakan *output* dari CNN berupa *image* hasil prediksi berupa data gambar dengan ukuran yang sama dengan masukan gambar yaitu 512x512. Setiap akhir dari proses konvolusi dilakukan *padding* sehingga keluaran dan masukan data citra memiliki ukuran yang sama.



Gambar 3.2 ReLU activation function



Gambar 3.3 Arsitektur U-Net dengan encoder Resnet

3.5. Pelatihan CNN

Pelatihan CNN akan diawali dengan penyiapan data input. Secara umum, penyiapan *dataset* dilakukan dengan memanfaatkan keluaran dari *preprocessing* pada subbab 3.3. CNN arsitektur U-Net ini bertujuan untuk melakukan segmentasi dengan melakukan *pixel-wise classification*. *Pixel-wise classification* adalah klasifikasi yang dilakukan terhadap setiap piksel pada citra. *Ground truth* yang disegmentasi secara manual yang telah didapatkan sebelumnya digunakan sebagai *desired output* dari model.

Pada CNN, jumlah data training sangat mempengaruhi akurasi dan performansi sistem, sementara data yang diperoleh hanya sebanyak 87 citra. Oleh sebab itu, sebelum pelatihan dilakukan augmentasi data yang bertujuan untuk memperbanyak data dengan melakukan transformasi geometrik dan diperoleh jumlah data hasil augmentasi sebanyak 1000 citra. Selanjutnya citra-citra ini dibagi secara *random* menjadi data latih dan data uji dengan perbandingan 75:25 sehingga diperoleh 750 data latih dan 250 data uji. Data latih ini digunakan pada proses pembelajaran sedangkan data uji digunakan sebagai proses pengujian sistem CNN secara keseluruhan serta untuk menghasilkan gambar prediksi hasil segmentasi.

Pada proses pelatihan, masukan yang digunakan yaitu berupa data pelatihan yang digunakan serta pemberian nilai *hyperparameter*. *Hyperparameter* adalah parameter yang mengontrol proses *learning* pada CNN. Dalam pembuatan model CNN, terdapat beberapa *hyperparameter* yang menjadi masukan fungsi yaitu batasan *epoch* dan nilai *batch size* pada proses *fitting*. *Epoch* adalah ketika seluruh *dataset* melalui proses *training* sampai kembali ke awal untuk satu kali putaran. Karena *epoch* terlalu besar untuk dimasukkan ke dalam komputasi komputer, maka perlu dibagi dalam satuan kecil yang disebut *batch*. Kemudian, diperlukan menentukan *learning rate* sebagai batasan dalam mencapai konvergensi yang menggunakan metode ADAM (*ADaptive Moment*). ADAM merupakan suatu jenis *optimizer* yang berfungsi untuk mengurangi waktu proses pembelajaran yang dibutuhkan dalam mencapai konvergensi menuju nilai *learning rate*.

3.6. Pengujian Performansi

Pengujian performansi akan dilakukan terhadap 2 proses yaitu peningkatan citra dan segmentasi citra.

3.6.1. Peningkatan Citra

Peningkatan kualitas citra digital bisa bersifat subjektif sehingga untuk mengukur efek dari peningkatan citra, langkah-langkah perhitungan secara kuantitatif perlu dilakukan. Pada penelitian ini dilakukan perhitungan metrik *Universal Image Quality Index* (UIQI) untuk menghitung nilai distorsi dari citra yang diolah, *Mean Square Error* (MSE) untuk menghitung nilai error antara citra asli dan citra yang diolah, *Peak-to-peak Signal to Noise Ratio* (PSNR) untuk mengukur besarnya *noise* terdistorsi, *Structural Similarity Index* (SSIM) untuk menghitung indeks kesamaan struktural antara citra asli dan citra yang telah diolah, *Relative Dimensionless Global Error* (ERGAS) untuk mengukur kualitas dari citra yang telah diproses berdasarkan nilai *error* rata-rata yang dinormalisasi, *spatial correlation coefficient* untuk menghitung kesamaan fitur spasial antara citra asli dengan citra yang telah diproses, *Multiscale SSIM*, *Visual Information Fidelity* (VIF), dan entropi diskrit untuk mengukur berapa banyak bit yang diperlukan untuk pengkodean data citra.

UIQI digunakan untuk menghitung jumlah transformasi yang dilakukan terhadap citra original. Metrik ini memiliki range dari -1 hingga 1 dimana 1 mengindikasikan bahwa citra original dan citra yang diproses memiliki banyak kesamaan. Wang dan Bovik mengusulkan cara untuk mengukur indeks kualitas gambar secara universal [26]. Perhitungan ini memecah perbandingan antara gambar asli dan gambar terdistorsi menjadi tiga perbandingan yaitu pencahayaan, kontras dan perbandingan struktural, seperti yang ditunjukkan pada persamaan berikut ini yang dijelaskan pada [27].

$$l(x, y) = \frac{2\mu_x\mu_y}{\mu_x^2 + \mu_y^2} \quad (3.1)$$

$$c(x, y) = \frac{2\sigma_x\sigma_y}{\sigma_x^2 + \sigma_y^2} \quad (3.2)$$

$$s(x, y) = \frac{2\sigma_{xy}}{\sigma_x + \sigma_y} \quad (3.3)$$

dimana $\mu_x\mu_y$ menyatakan nilai *mean* dari gambar asli dan gambar terdistorsi. Kemudian, $\sigma_x\sigma_y$ merupakan standar deviasi dari gambar asli dan terdistorsi, dan σ_{xy} adalah kovarian dari kedua gambar.

Berdasarkan ketiga perbandingan tersebut, UIQI dinyatakan dalam Persamaan 3.4. UIQI akan mengukur intensitas pencahayaan pada suatu gambar.

$$UIQI(x, y) = l(x, y)c(x, y)s(x, y)$$

$$UIQI(x, y) = \frac{4\mu_x\mu_y\mu_{xy}}{(\mu_x^2 + \mu_y^2) + (\sigma_x^2 + \sigma_y^2)} \quad (3.4)$$

MSE adalah nilai *error* kuadrat rata-rata antara citra original dan citra yang telah diproses. *Peak Signal to Noise Ratio* atau PSNR adalah perbandingan antara nilai maksimum dari citra yang diukur dengan besarnya *noise* terdistorsi yang berpengaruh pada citra tersebut. PSNR diukur dalam satuan desibel logaritmik. Pada penelitian ini, PSNR digunakan untuk mengetahui perbandingan kualitas citra masukan sebelum dan sesudah diterapkan peningkatan. Untuk menentukan PSNR, terlebih dahulu harus ditentukan *Mean Square Error* (MSE). MSE adalah nilai *error* kuadrat rata-rata antara kedua citra tersebut. Representasi matematis dari MSE dan PSNR ditunjukkan pada Persamaan 3.5 dan 3.6.

$$MSE = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [l(i, j) - K(i, j)]^2 \quad (3.5)$$

$$PSNR = 10 \log_{10} \left(\frac{MAX_i^2}{MSE} \right) \quad (3.6)$$

dengan m adalah jumlah baris, n adalah jumlah kolom dari citra, dan MAX_i adalah fluktuasi maksimal dari citra input. Nilai PSNR menunjukkan rasio antara sinyal dan *noise* pada citra. Nilai PSNR terbaik pada sebuah citra berada pada *range* 30 sampai 50 dB [28]. Nilai PSNR yang lebih dari 50 menunjukkan bahwa nilai *error* antara citra asli dan citra setelah peningkatan sangat rendah yang mengimplikasikan bahwa peningkatan tidak banyak mengubah citra (peningkatan tidak maksimal). Nilai PSNR yang kurang dari 30 menunjukkan bahwa nilai *error* tinggi yang mengimplikasikan bahwa terdapat banyak *noise* pada citra tersebut.

Wang, *et al.* [29], mengusulkan *Structural Similarity Index* (SSIM) untuk memperbaiki kekurangan dari UIQI. Pertama, gambar asli dan terdistorsi dibagi menjadi blok berukuran 8×8 dan kemudian blok dikonversi menjadi vektor. Kemudian, dua nilai *mean* dan dua nilai standar derivasi dan satu nilai kovarian dihitung dari gambar. Terakhir, perbandingan pencahayaan, kontras dan struktur berdasarkan nilai statistik dihitung seperti pada perhitungan UIQI, indeks kesamaan struktural antara citra x dan y dihitung berdasarkan Persamaan 3.7. SSIM menunjukkan tingkat kemiripan suatu citra dengan citra aslinya.

$$SSIM(x, y) = \frac{(2\mu_x\mu_y+c_1)(2\sigma_x\sigma_y+c_2)}{(\mu_x^2+\mu_y^2+c_1)(\sigma_x^2+\sigma_y^2+c_1)} \quad (3.7)$$

Multiscale SSIM adalah pengembangan lebih lanjut dari SSIM yang menggunakan lebih dari satu skala dan lebih dari satu *stage* untuk proses *sub-sampling*. *Multiscale SSIM* memiliki performansi yang lebih baik daripada SSIM pada beberapa subjek citra.

ERGAS adalah metrik yang digunakan untuk mengukur kualitas dari citra yang telah diproses berdasarkan nilai *error* rata-rata yang dinormalisasi terhadap setiap *band*. Tingginya nilai ERGAS menunjukkan bahwa terjadi banyak distorsi pada citra sedangkan nilai ERGAS yang rendah menunjukkan bahwa citra yang diproses memiliki banyak kesamaan dengan citra asli.

Spatial correlation coefficient digunakan untuk menghitung kesamaan fitur spasial antara citra asli dengan citra yang telah diproses. Nilai *spatial correlation coefficient* yang mendekati 1 mengindikasikan bahwa citra asli dan citra yang telah diproses memiliki banyak kesamaan. VIFP adalah metrik yang digunakan untuk penilaian kualitas gambar asli berdasarkan statistik natural *scene* dan informasi citra yang diekstraksi oleh sistem visual manusia.

Entropi adalah ukuran konten informasi dari sebuah citra yang dapat diartikan sebagai ketidakpastian rata-rata dari sumber informasi. Entropi diskrit adalah penjumlahan dari probabilitas hasil dikalikan dengan *log* kebalikan dari probabilitas hasil, dengan mempertimbangkan semua hasil yang memungkinkan $\{1, 2, \dots, n\}$ dalam kejadian $\{x_1, x_2, \dots, x_n\}$ dengan n adalah tingkat abu-abu atau *gray level*; $p(i)$ adalah probabilitas pada tingkat abu-abu dari i , yang berisi semua jumlah histogram. Entropi diskrit dapat ditulis sebagai Persamaan 3.8 dan 3.9

$$H(X) = - \sum_{i=1}^k p(i) \log_2 p(i) \quad (3.8)$$

$$\sum_{i=1}^k p(i) = 1 \quad (3.9)$$

Entropi diskrit adalah perhitungan statistik dari keacakan atau *randomness*. Untuk pemrosesan citra, entropi diskrit adalah sebuah ukuran berapa banyak bit yang diperlukan untuk pengkodean data citra. Entropi mewakili jumlah rata-rata informasi yang disampaikan dari setiap citra. Entropi dari citra yang telah disempurnakan atau ditingkatkan lebih tinggi dari citra aslinya. Dari aspek lain, ini menunjukkan bahwa citra yang disempurnakan relatif lebih kompleks. Citra yang relatif kompleks memiliki entropi yang lebih tinggi daripada citra yang relatif sederhana. Ketika piksel dalam citra didistribusikan di antara lebih banyak tingkat abu-abu, nilai dari entropi diskrit meningkat[27].

3.6.2. Segmentasi Citra

Pada tahap pengujian Tugas Akhir ini diaplikasikan metode pengujian untuk mengamati hasil performansi sistem dalam melakukan segmentasi semantik. Pengujian dilakukan khususnya pada blok diagram model pembelajaran U-Net. Untuk penghitungan performansi proses segmentasi semantik CNN dari data prediksi yang telah dihasilkan digunakan *Dice Similarity Coefficient* (DSC) yang merupakan *spatial overlap index* dengan rentang nilai dari 0 hingga 1 yaitu 0 menunjukkan tidak ada bagian *pixel* yang tumpang tindih antara dua gambar biner (prediksi segmentasi dan *ground truth*) serta 1 menunjukkan bahwa piksel tumpang tindih sepenuhnya. Metode pengujian DSC menggunakan Persamaan 3.10 dengan X adalah area anotasi *ground truth* dan Y adalah area segmentasi yang didapatkan dari CNN serta perhitungan *loss* menggunakan *Dice Loss*.

$$DSC = \frac{2|X \cap Y|}{|X| + |Y|} \quad (3.10)$$

BAB 4

PENGUJIAN SISTEM

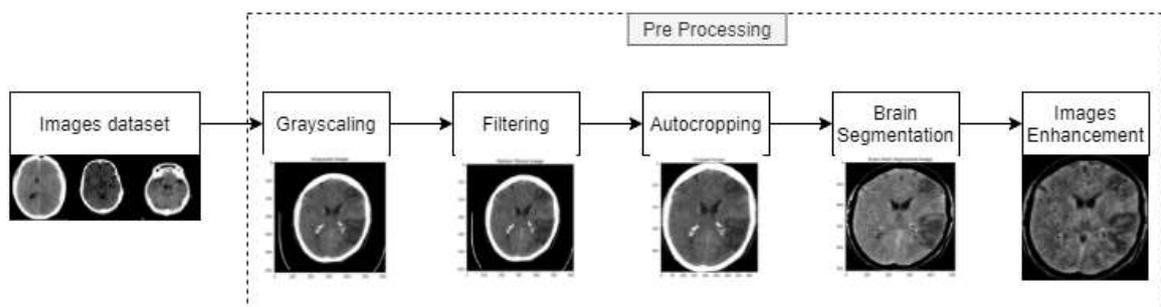
4.1 *Preprocessing*

Preprocessing merupakan tahap pertama yang dilakukan terhadap citra yang terdiri atas proses *grayscale*, *filtering*, *autocropping*, segmentasi otak, dan peningkatan citra seperti yang ditunjukkan oleh Gambar 4.1. Pada tahap ini citra yang digunakan diambil dari dataset yang ada pada website *radiopaedia.org*. Dari website tersebut diperoleh 6 kasus yang sesuai dengan kebutuhan pada penelitian ini, yaitu citra CT scan pasien stroke iskemia. Kasus-kasus ini yaitu *acute infarction*, *ischemic stroke*, *PCA territory infarct*, *pontine infarct*, *subacute left middle cerebral artery infarct*, dan *posterior cerebral artery territory infarct*. Setiap kasus memiliki jumlah data yang berbeda, dan tidak semua *slice* memiliki lesi *hypodense* pada citranya. Untuk mengurangi *non trainable* parameter pada pelatihan CNN, dipilihlah data yang memiliki lesi *hypodense* dan diperoleh sebanyak 87 citra. Pada pengujian ini 6 citra yang mewakili setiap kasus dipilih untuk kemudian dilakukan pengujian. Citra tersebut kemudian dibaca dengan menggunakan *library* OpenCV pada *workspace* Google Colab. *Workspace* Google Colab dipilih karena fitur GPU yang disediakannya sehingga dapat membuat proses *training* CNN menjadi lebih cepat.

Sebelum tahap *preprocessing* dimulai, dilakukan pengecekan terhadap citra yang akan digunakan pada penelitian ini. Ini dilakukan untuk menentukan proses yang dibutuhkan dalam tahap *preprocessing*. Lokasi citra berada pada Google Drive, sehingga Google Colab harus dihubungkan dengan Google Drive dengan cara *mounting* data. Pengecekan dilakukan dengan melihat tipe data, ukuran, dan histogram intensitas citra. Dari pengecekan ini diperoleh bahwa citra memiliki tipe data *uint8* dengan ukuran dan sebaran intensitas yang bervariasi. Hasil pengecekan terhadap citra dapat dilihat pada Gambar 4.2.

4.1.1 *Grayscale*

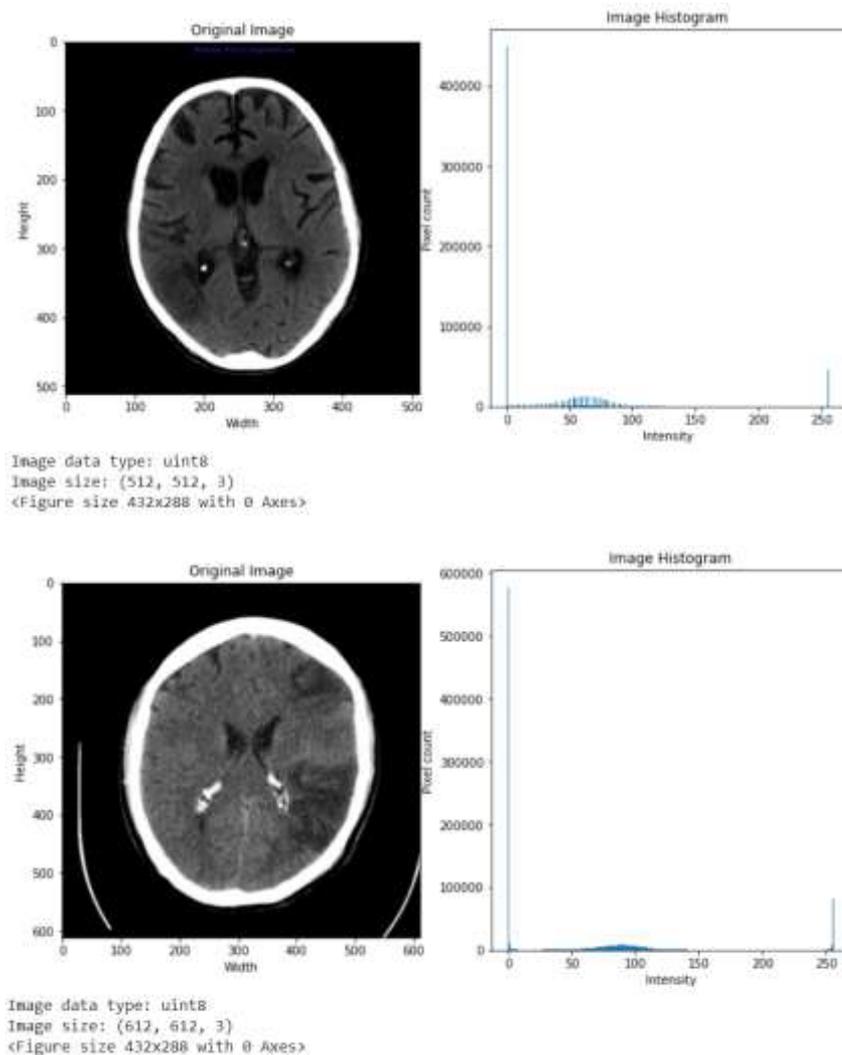
Pada tahap ini, citra dipanggil dalam program lalu ditampilkan dalam format *grayscale*. Citra tersebut terbaca sebagai sebuah *array* dengan ukuran [lebar, panjang, 3]. Nilai 3 pada matriks menunjukkan bahwa citra masih memiliki tipe kanal warna RGB, sehingga untuk memudahkan pemrosesan pada penelitian ini, masukan citra diubah ke dalam citra *grayscale* atau citra dua dimensi yang memiliki *array* dengan ukuran [lebar, panjang]. Citra hasil *grayscale* dapat dilihat pada Gambar 4.3.



Gambar 4.1 Tahap *preprocessing*

4.1.2 Filter median dan morfologi

Setelah citra diubah menjadi *grayscale*, tahapannya selanjutnya adalah melakukan *filtering* untuk mengurangi *noise* yang terdapat pada citra. Pada penelitian ini digunakan 2 macam filter dengan fungsi berbeda yaitu median filter untuk menghilangkan *salt and pepper noise*, yaitu *noise* berupa titik-titik berwarna hitam dan putih pada citra yang bukan merupakan bagian dari citra dan filter morfologi untuk menghilangkan daerah diluar ROI. Untuk median filter, digunakan kernel dengan ukuran 3x3 yang memiliki performansi yang lebih baik untuk menghilangkan *noise* pada citra stroke iskemia dibandingkan dengan kernel ukuran lainnya[6]. Untuk filter morfologi digunakan metode *opening*, yaitu erosi yang diikuti oleh dilasi dengan iterasi erosi 4 dan iterasi dilasi 2. Iterasi ini dipilih setelah melakukan beberapa percobaan terhadap beberapa nilai iterasi dan diperoleh nilai iterasi terbaik seperti yang telah disebutkan sebelumnya. Agar filter morfologi dapat bekerja dengan maksimal untuk menentukan batas citra, citra diubah sementara menjadi citra biner. Setelah batas citra ditentukan, batas ini akan diterapkan pada citra hasil median filter. Hasil dari filter median dapat dilihat pada Gambar 4.4 dan hasil filter morfologi dapat dilihat pada Gambar 4.5.



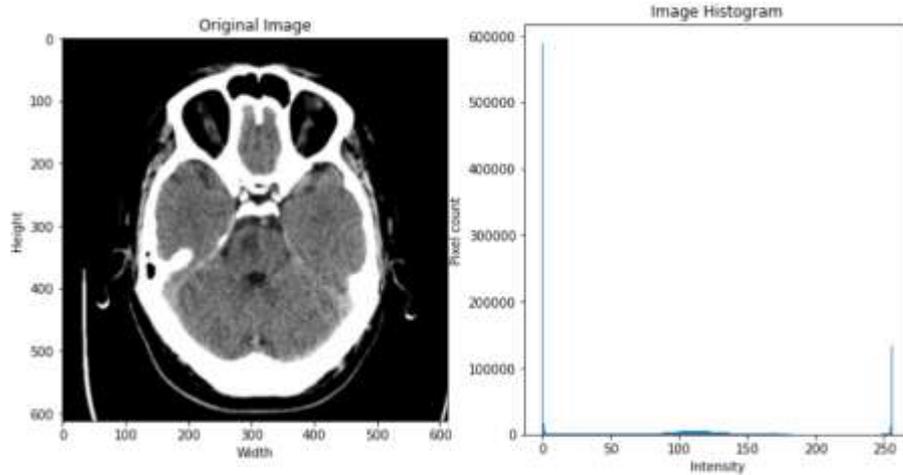


Image data type: uint8
 Image size: (612, 613, 3)
 <Figure size 432x288 with 0 Axes>

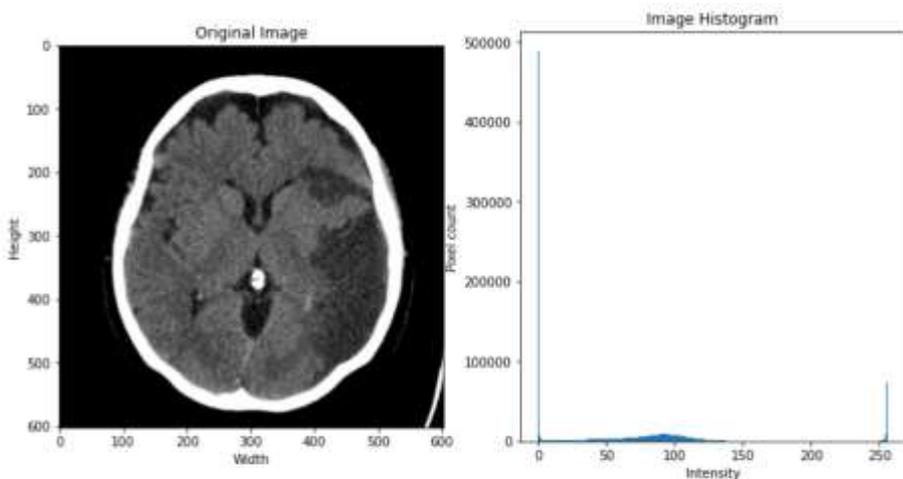


Image data type: uint8
 Image size: (603, 603, 3)
 <Figure size 432x288 with 0 Axes>

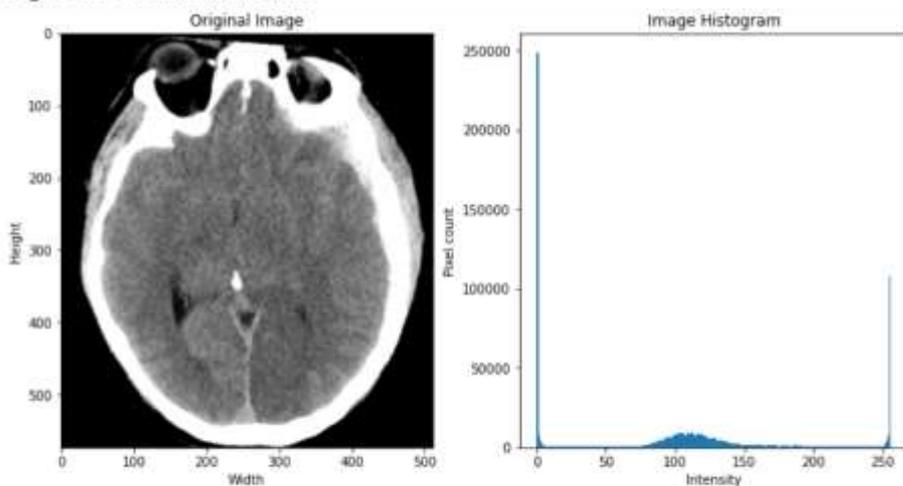
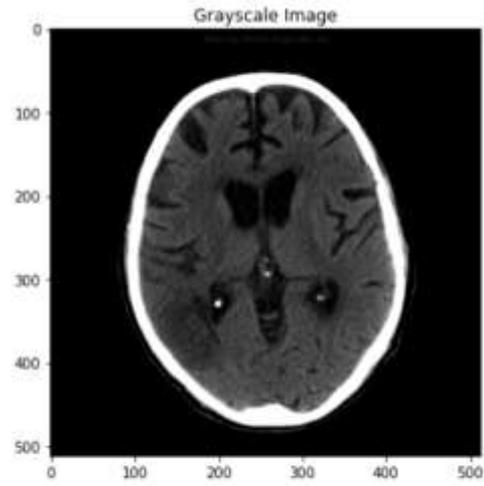
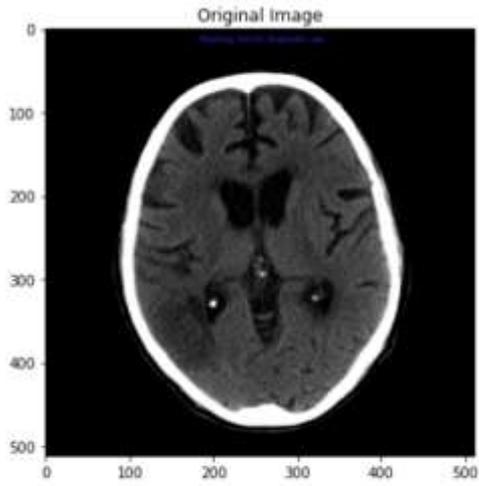
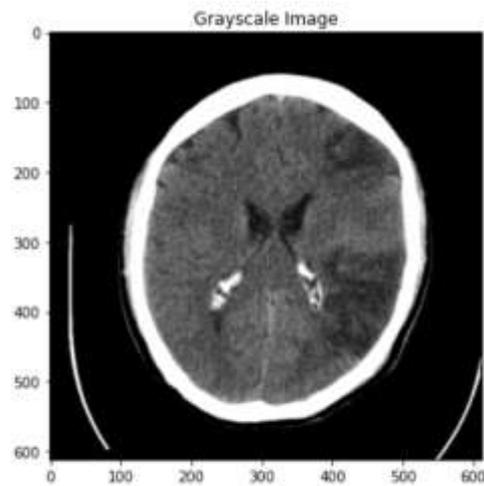
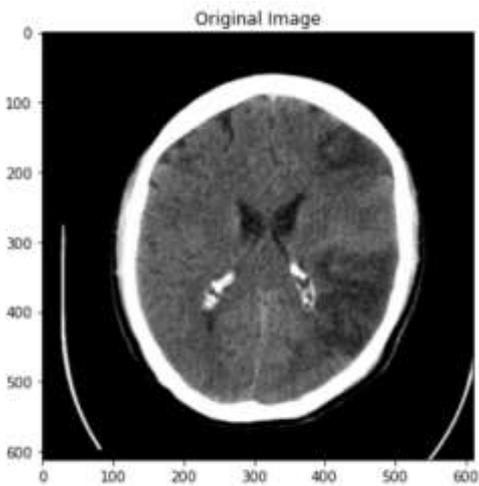


Image data type: uint8
 Image size: (574, 512, 3)
 <Figure size 432x288 with 0 Axes>

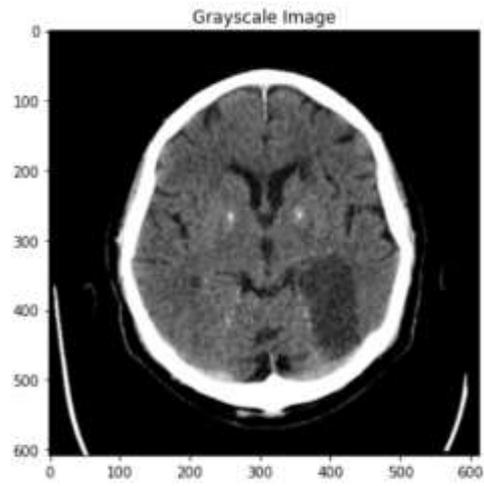
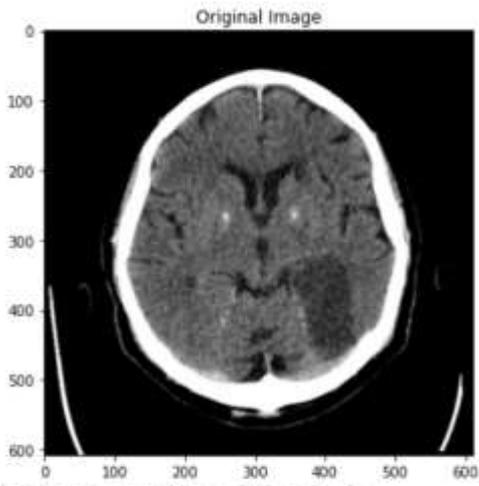
Gambar 4.2 Hasil pengecekan data citra



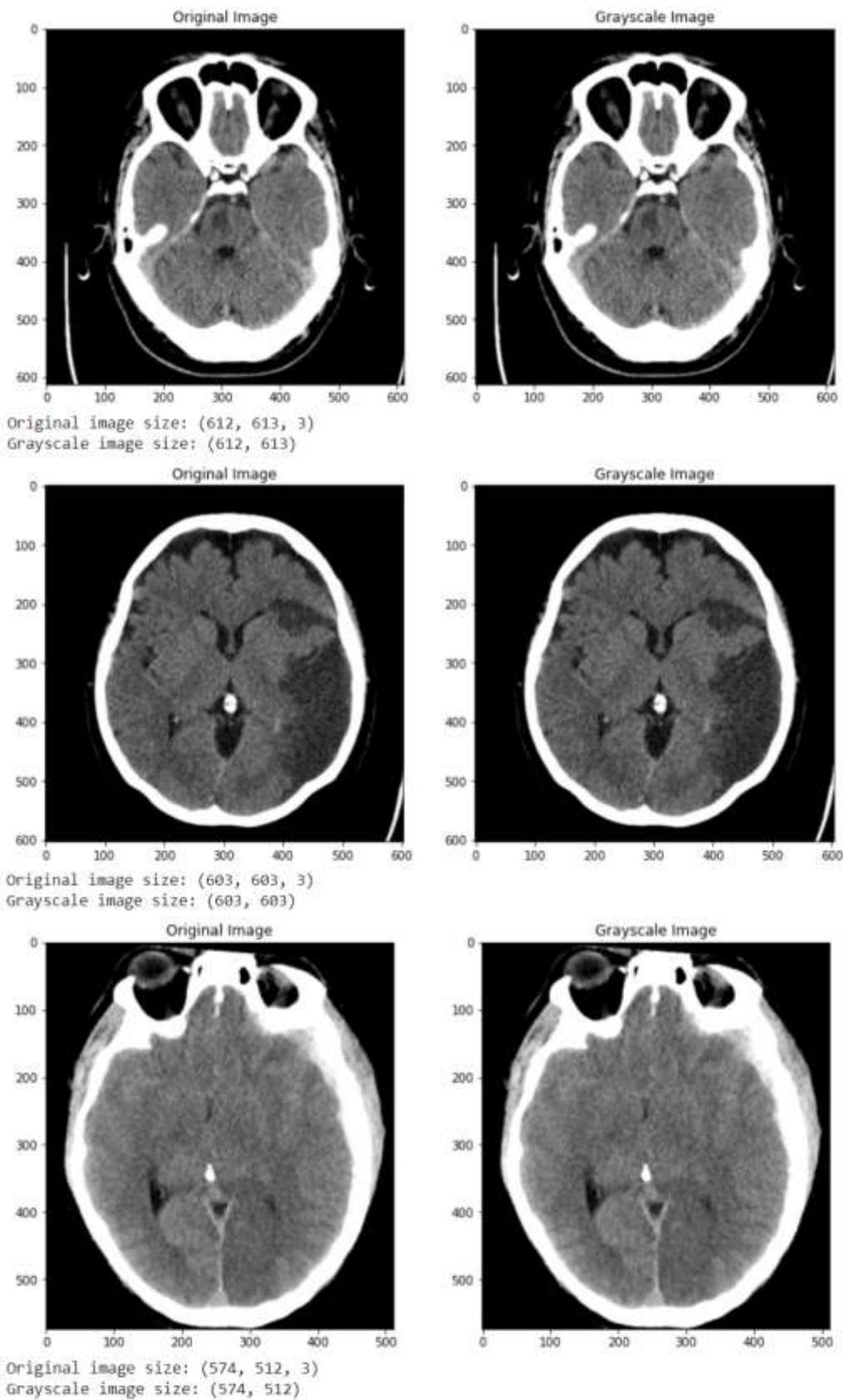
Original image size: (512, 512, 3)
Grayscale image size: (512, 512)



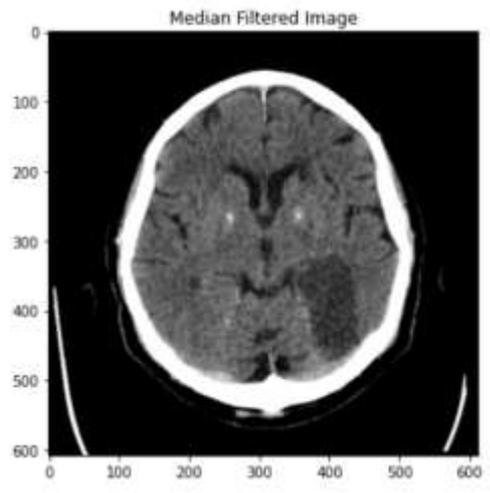
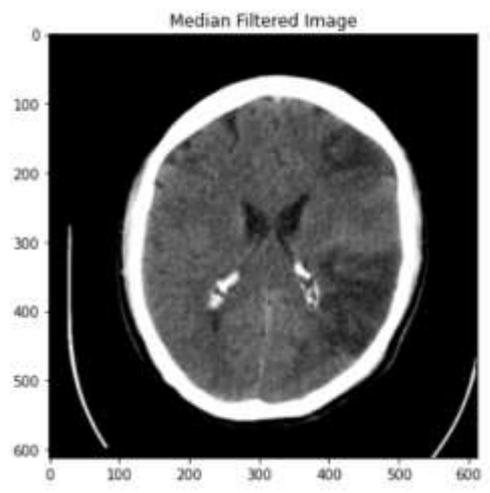
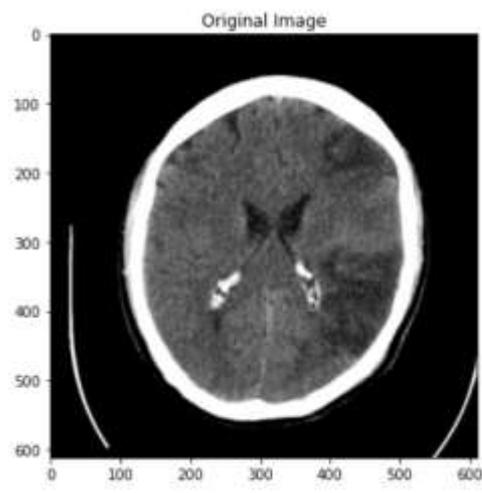
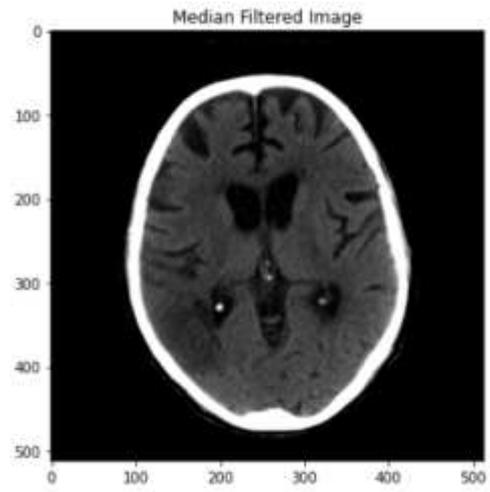
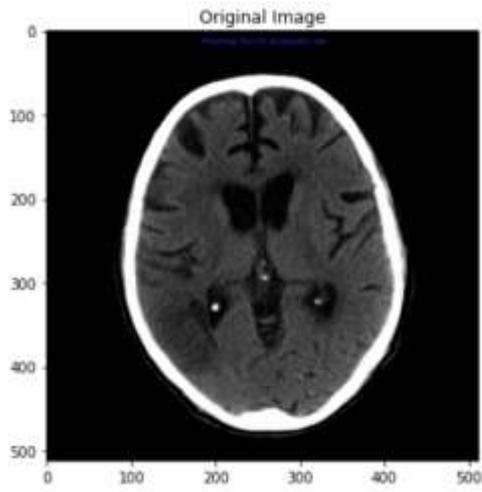
Original image size: (612, 612, 3)
Grayscale image size: (612, 612)

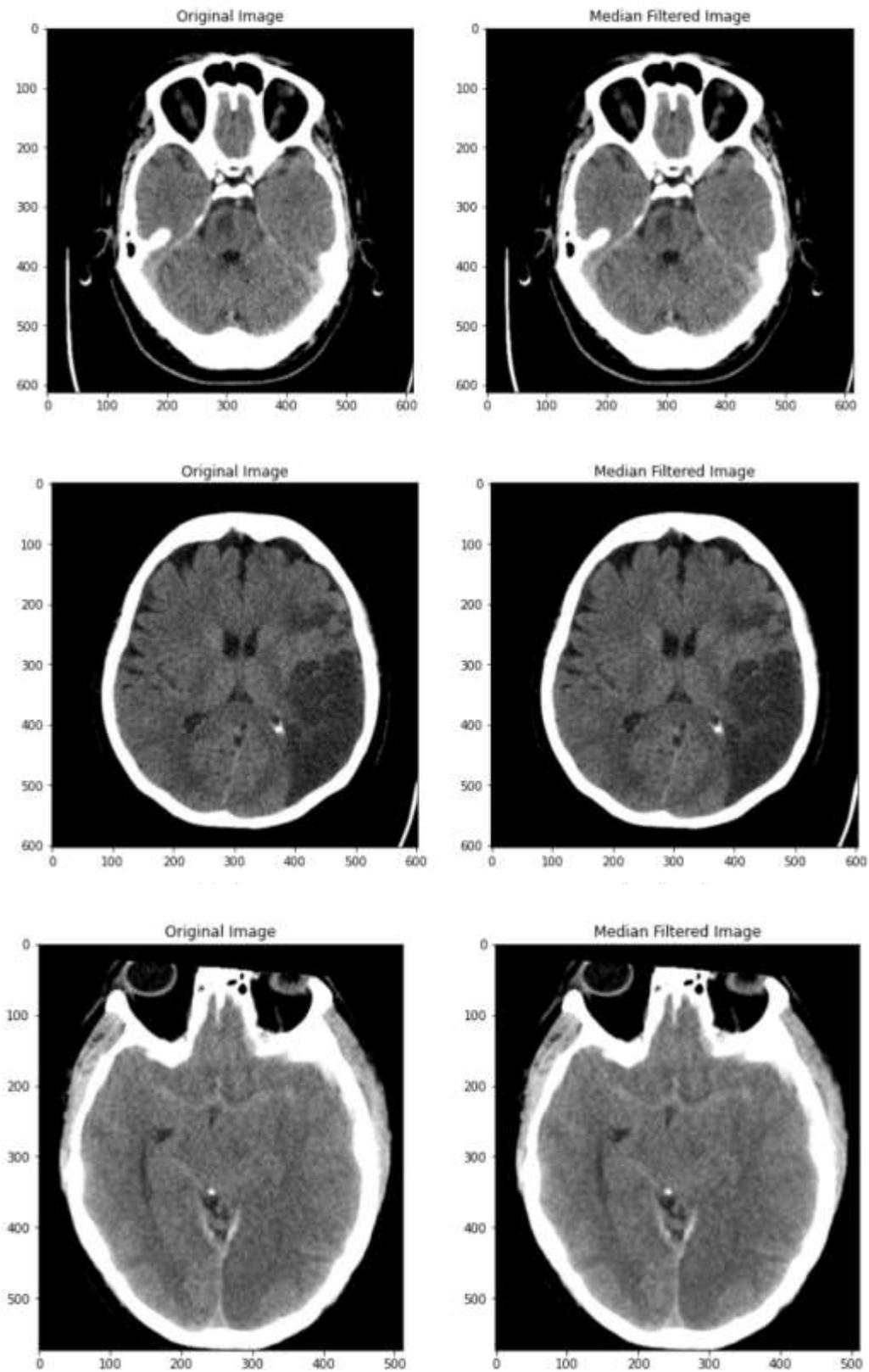


Original image size: (608, 612, 3)
Grayscale image size: (608, 612)

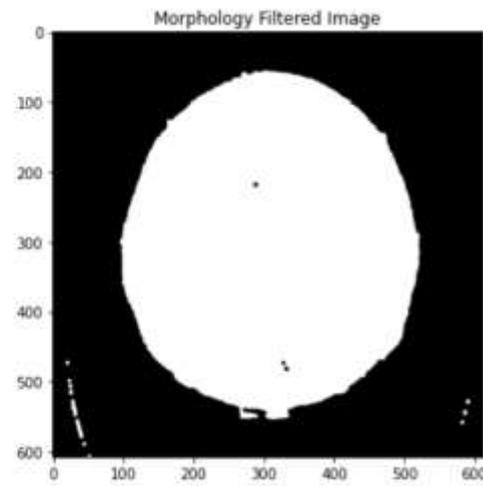
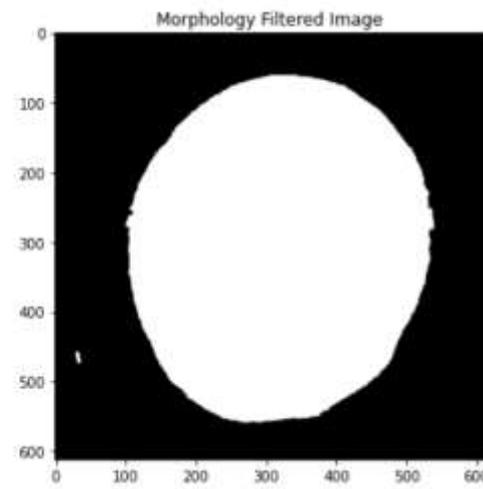
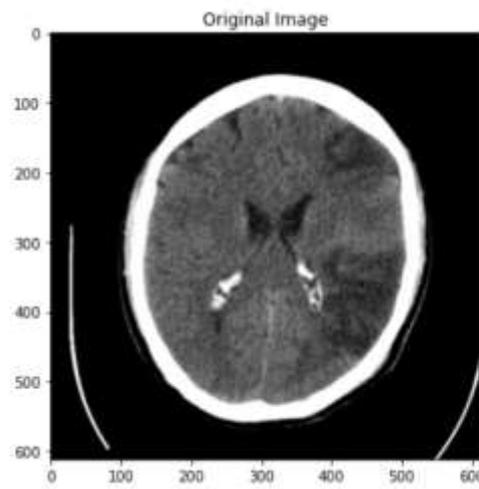
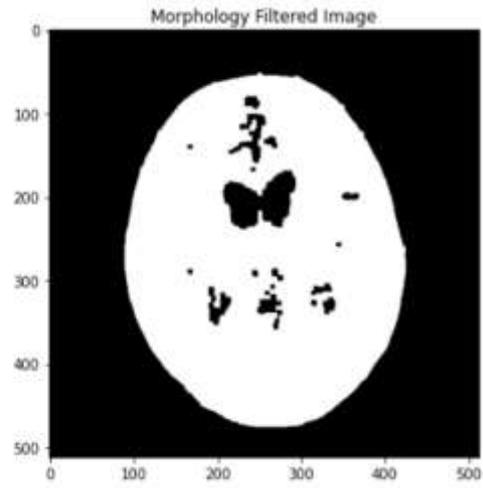
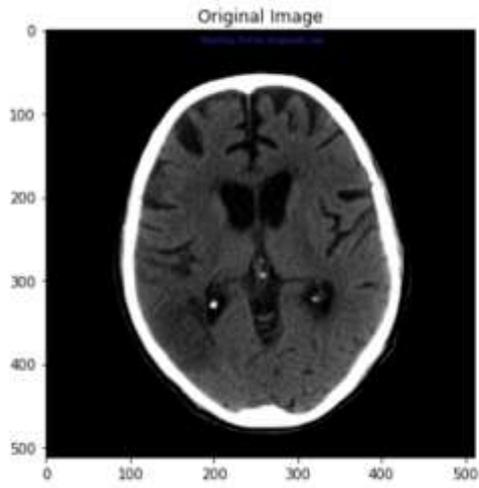


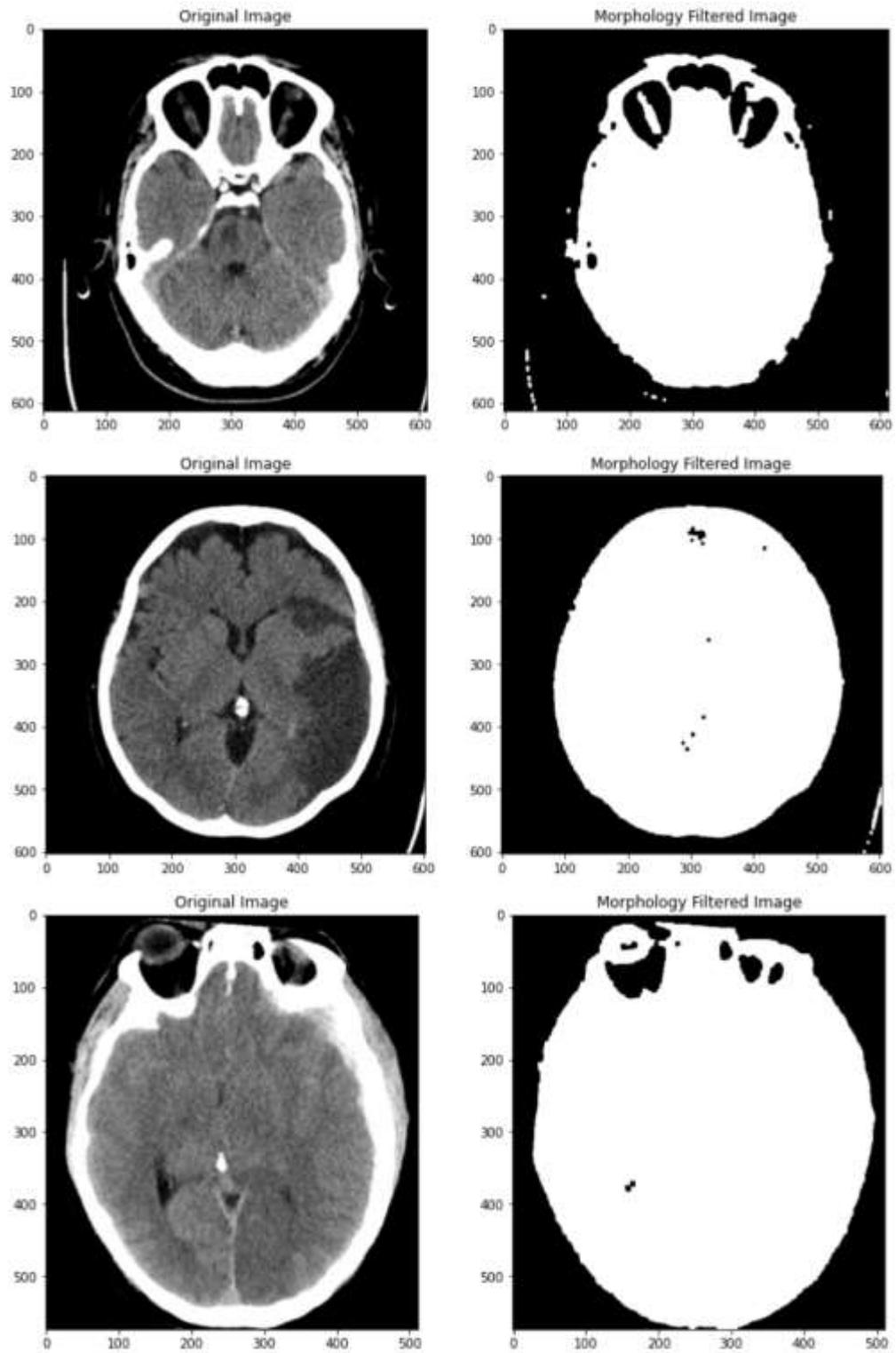
Gambar 4.3 Hasil grayscaling





Gambar 4.4 Hasil median filter





Gambar 4.5 Hasil morfologi filter

4.1.3 Autocropping

Proses selanjutnya adalah *autocropping*, yang terdiri dari 2 tahapan, yaitu penentuan batas melalui kontur dan penentuan kontur maksimum sebagai batas *cropping*. Kontur berfungsi untuk menemukan batasan citra yang akan diproses di tahap selanjutnya dengan cara memberikan gambaran pada garis tepi terluar dari citra tersebut. Proses ini berguna untuk menentukan titik-titik yang akan digunakan dalam proses *autocropping*. Mode pengambilan kontur memakai mode pengambilan eksternal yang akan mengambil kontur bagian luar dari objek pada citra. Mode pendekatan yang digunakan yaitu metode *simple*. Metode ini dipilih karena metode *simple* menghilangkan titik-titik yang tidak dibutuhkan pada penentuan kontur. Contohnya pada kontur dengan garis lurus, pada mode pendekatan *none*, setiap poin akan dicatat sehingga akan lebih boros memori. Sedangkan pada mode pendekatan *simple*, hanya diambil 2 titik ujung dan dihubungkan antar titik tersebut. Setelah nilai kontur maksimal diperoleh, nilai ini akan dijadikan batas untuk memotong citra pada proses *autocropping*. Proses ini dilakukan untuk mendapatkan titik paling atas, paling bawah, paling kanan dan paling kiri untuk memotong tepian luar *contour* dengan *background*-nya. Proses ini diterapkan pada hasil filter median. Filter morfologi hanya digunakan untuk menjembatani proses ini. Hasil akhir dari proses *autocropping* akan di-*resize* menjadi 512x512 untuk menyesuaikan ukuran dengan CNN. Hasil *autocropping* dapat dilihat pada Gambar 4.6.

4.1.4 Segmentasi Otak

Segmentasi otak dari tulang tengkorak dilakukan dengan menentukan batas antara otak dan tengkorak menggunakan Laplacian filter. Untuk kernel dari Laplacian filter digunakan nilai alpha 0.5 sehingga kernel Laplace pada penelitian ini adalah sebagai berikut:

$$h = \frac{4}{\alpha+1} \begin{bmatrix} \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \\ \frac{1-\alpha}{4} & -1 & \frac{1-\alpha}{4} \\ \frac{\alpha}{4} & \frac{1-\alpha}{4} & \frac{\alpha}{4} \end{bmatrix}; \alpha = 0.5$$

$$h = 2.667 \begin{bmatrix} 0.125 & 0.125 & 0.125 \\ 0.125 & -1 & 0.125 \\ 0.125 & 0.125 & 0.125 \end{bmatrix}$$

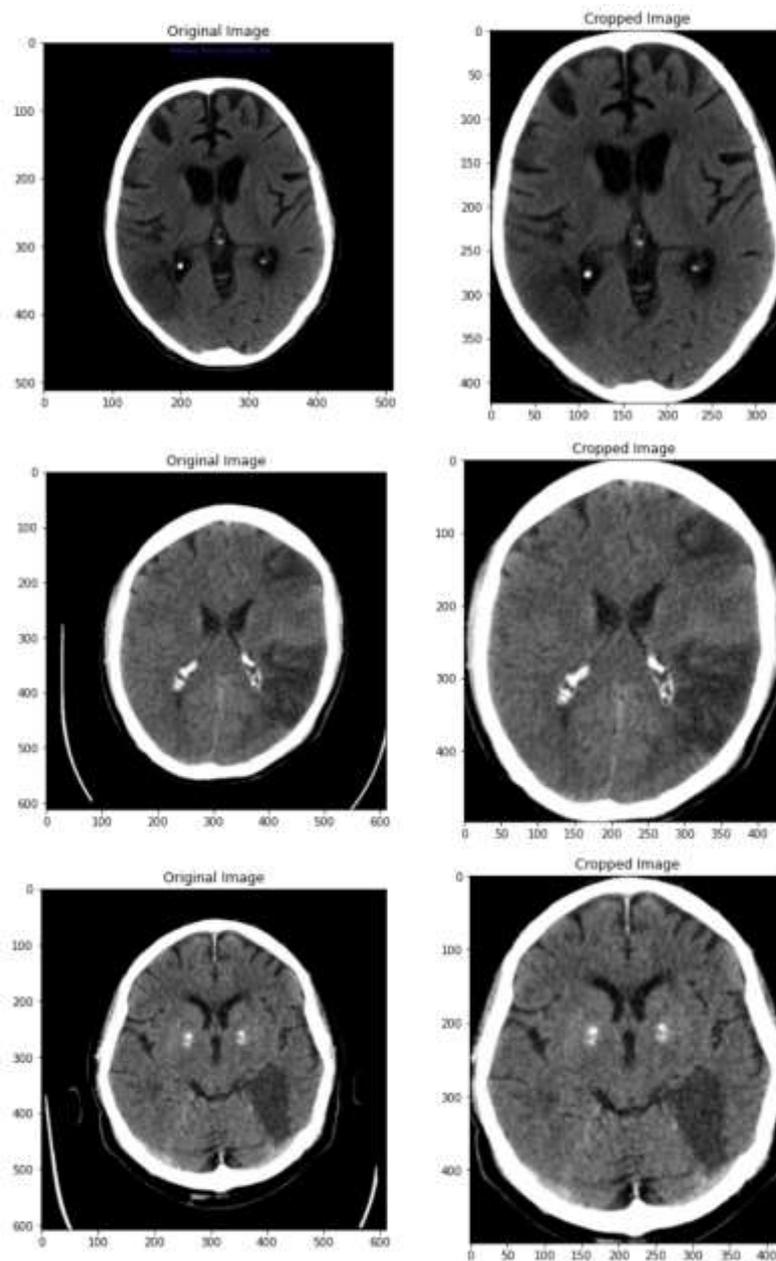
Kernel ini dikonvolusi dengan citra hasil *autocropping* sehingga diperoleh batas antara otak dengan tulang tengkorak. Setelah batas diperoleh, dilakukan *thresholding* terhadap citra tersebut sehingga diperoleh hasil berupa bagian otak yang telah tersegmentasi dari tulangnya. Hasil segmentasi otak dapat dilihat pada Gambar 4.7.

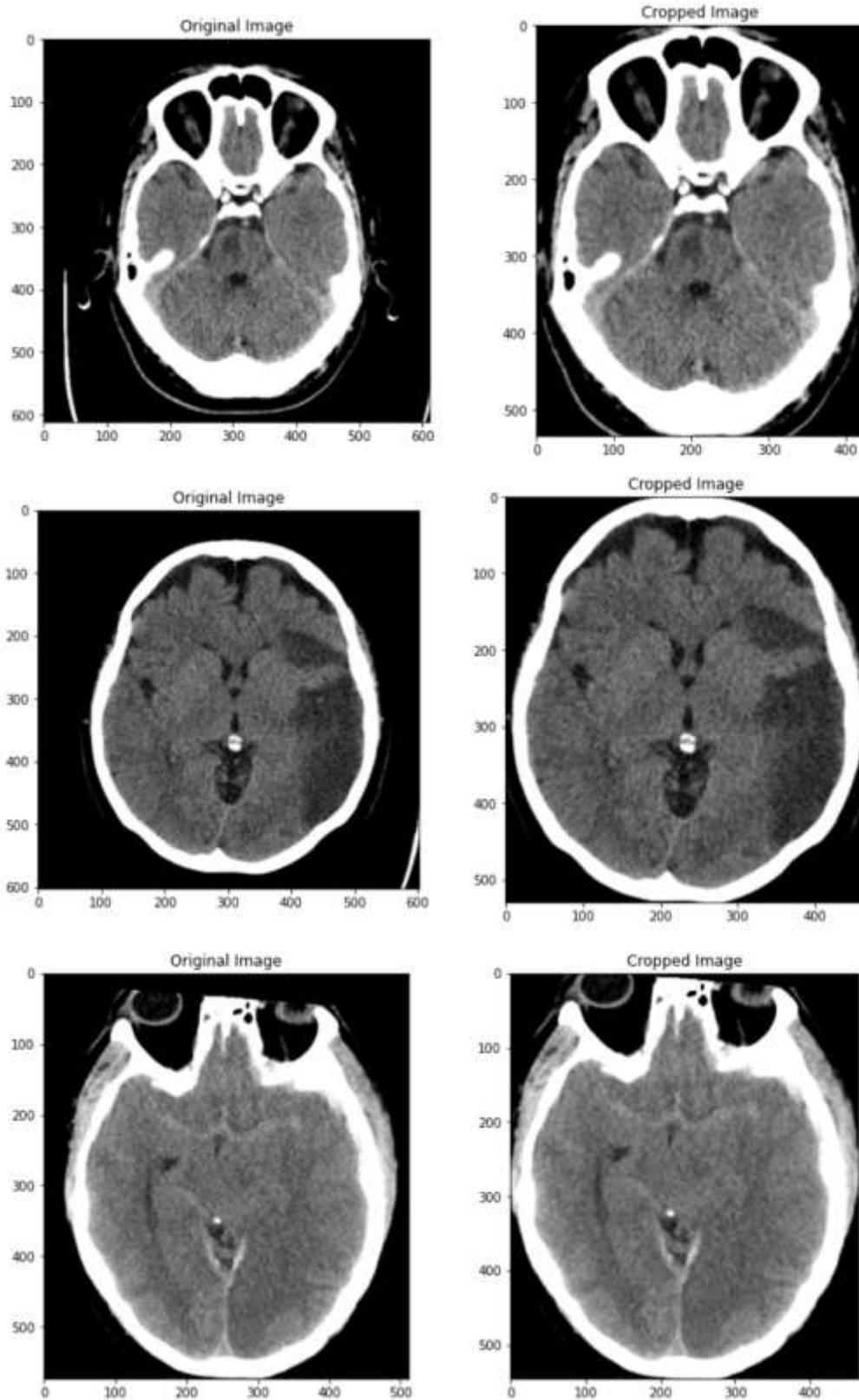
4.1.5 Peningkatan Citra

Tahap ini diawali dengan penerapan metode *Discrete Wavelet Transform* pada citra. Langkah pertama yang dilakukan adalah memilih jenis *mother wavelet* yang akan digunakan. Ada beberapa *mother wavelet* yang coba diterapkan pada penelitian ini, yaitu Biorthogonal, Coiflet, Daubechies, Discrete meyer, Haar, Reverse biorthogonal dan Symlet dengan skala yang berbeda-beda. Ini dilakukan untuk memilih diantara kelima jenis wavelet dan skalanya tersebut yang paling baik performanya dalam *denoising* citra CT scan otak. Untuk pengujian *mother wavelet* Biorthogonal dan Reverse biorthogonal, digunakan skala 2.2, 3.3, 4.4, 5.5, dan 6.8.

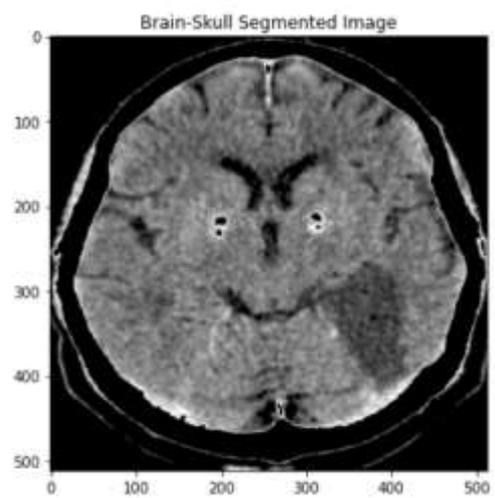
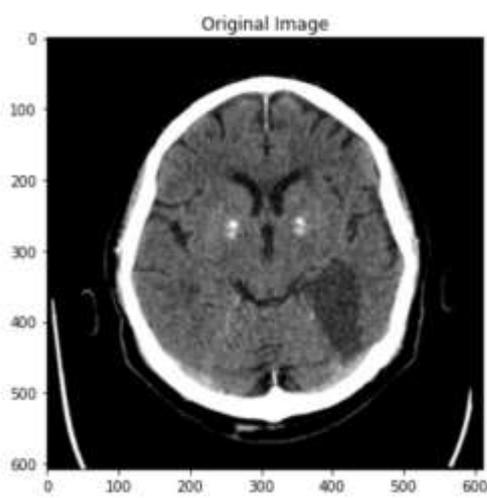
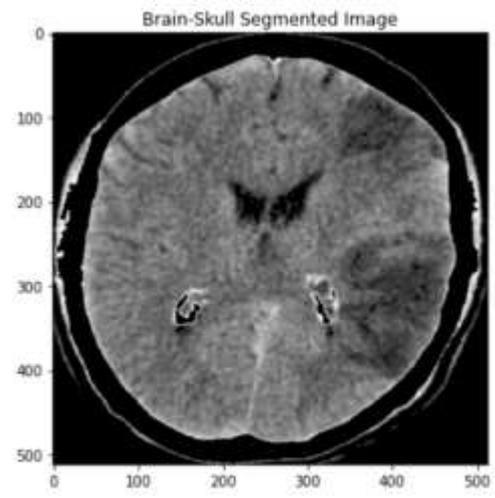
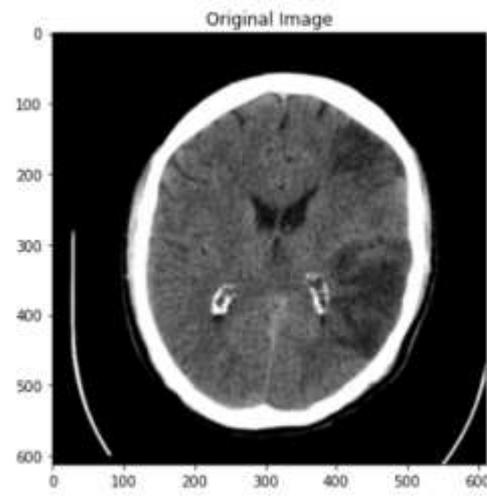
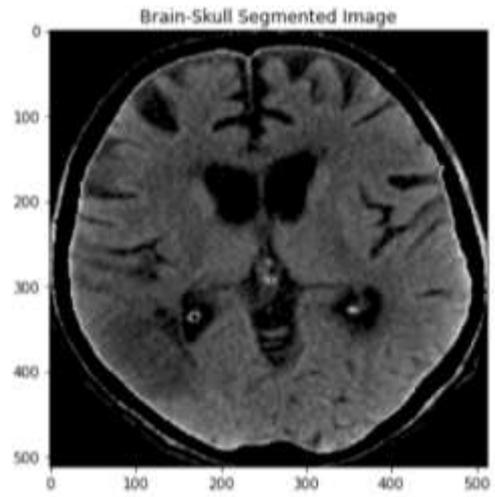
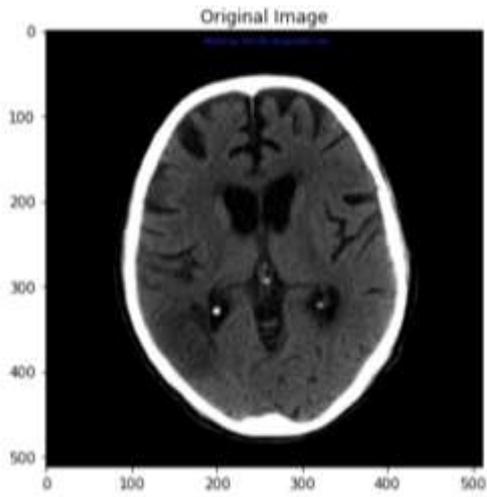
Untuk pengujian *mother wavelet* Coiflet digunakan skala 1 hingga 5. Untuk pengujian *mother wavelet* Daubechies dan Symlet digunakan skala 1, 5, 10, 15, dan 20. Sedangkan untuk discrete meyer dan Haar hanya memiliki 1 skala. Perbandingan dilakukan terhadap skala didalam *mother wavelet*, antar *mother wavelet*, dan antar level dekomposisi.

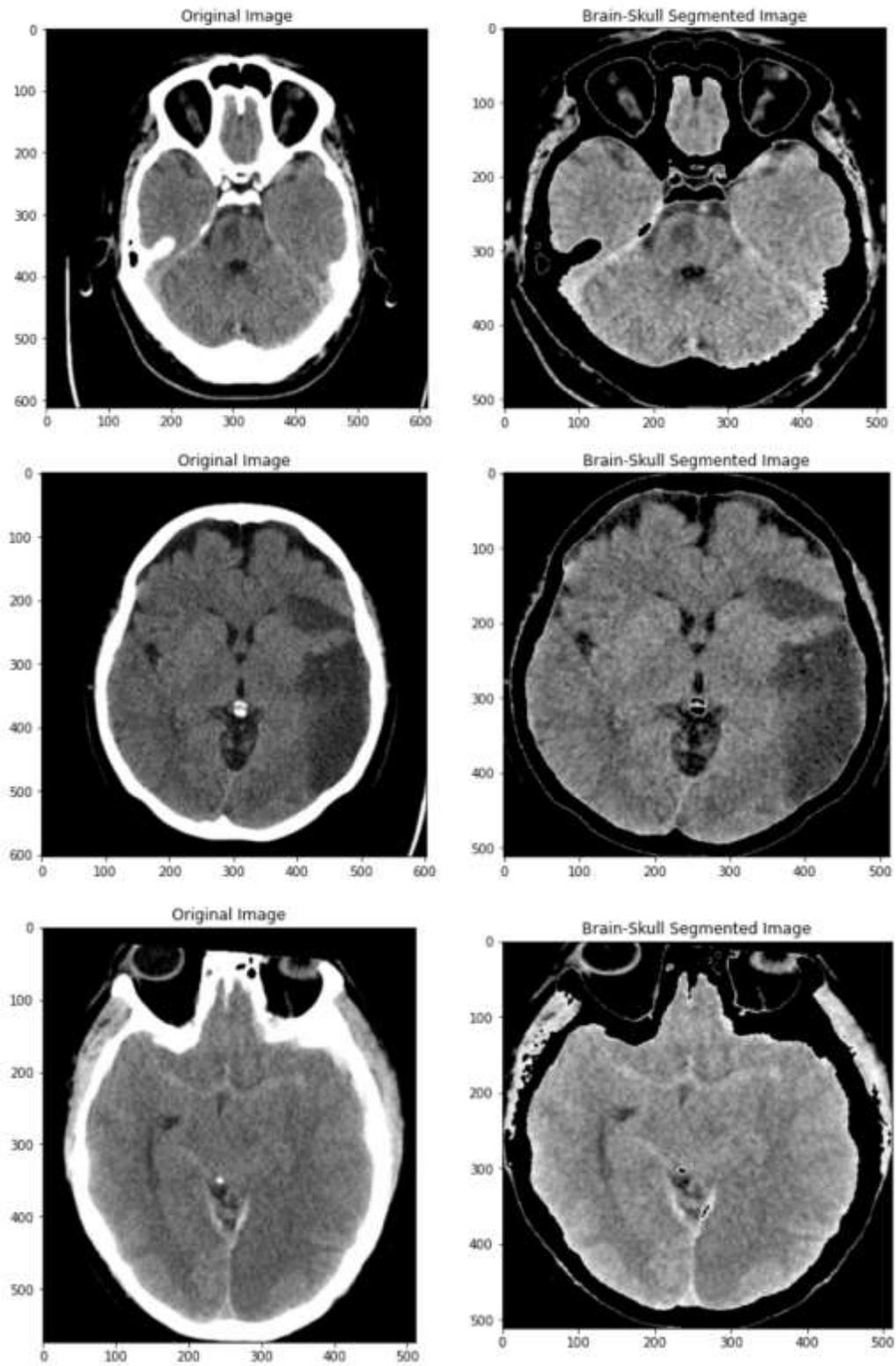
Perbandingan dilakukan dengan perhitungan metrik *Universal Image Quality Index* (UIQI), *Mean Square Error* (MSE), *Peak-to-peak Signal to Noise Ratio* (PSNR), *Structural Similarity Index* (SSIM), *Relative Dimensionless Global Error* (ERGAS), *spatial correlation coefficient*, *Multiscale SSIM*, *Visual Information Fidelity* (VIF), dan entropi diskrit untuk mengetahui jenis *mother wavelet* yang memiliki hasil rekonstruksi yang terbaik. Metrik ini diterapkan pada 87 citra dan ditentukan *mother wavelet* yang memiliki nilai metrik paling tinggi pada mayoritas citra. Ini dilihat dari jumlah citra yang memiliki nilai metrik tertinggi pada *mother wavelet* tersebut. Hasilnya dapat dilihat pada Gambar 4.8 hingga 4.35.



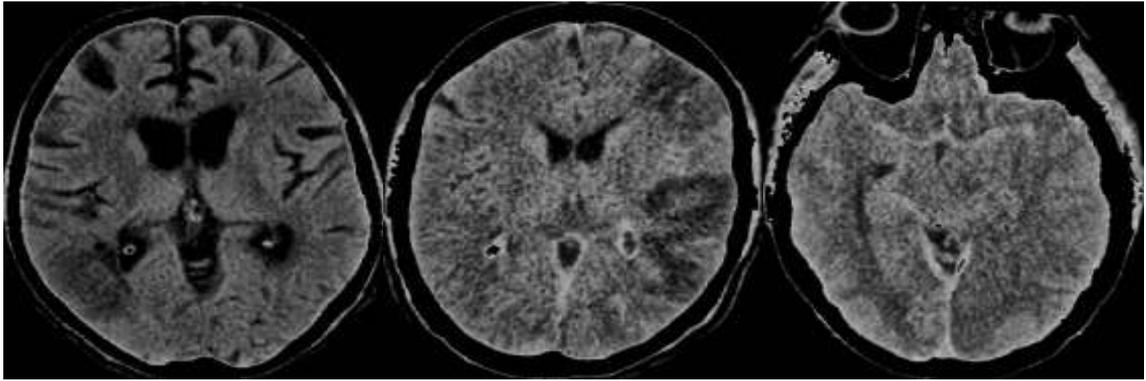


Gambar 4.6 Hasil autocropping

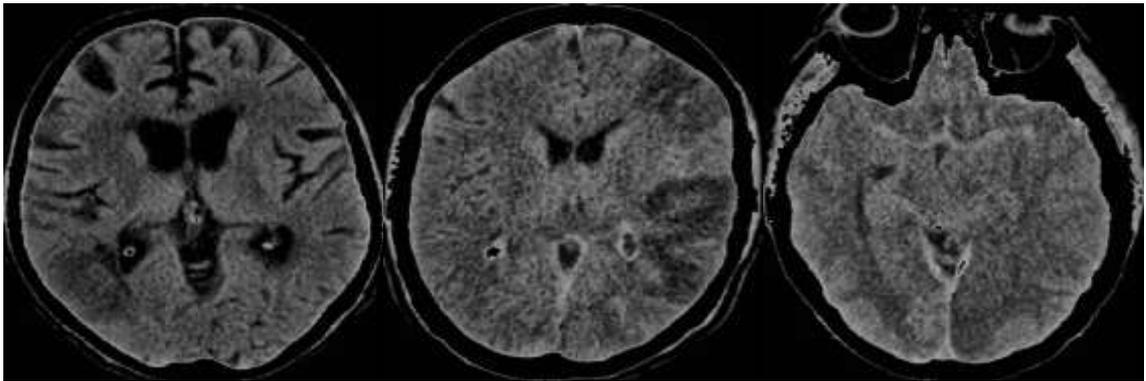




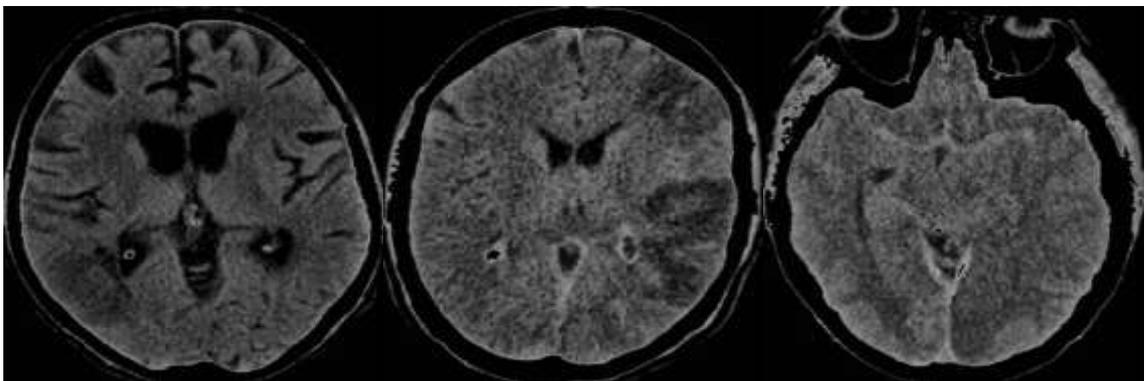
Gambar 4.7 Hasil segmentasi otak dan tengkorak



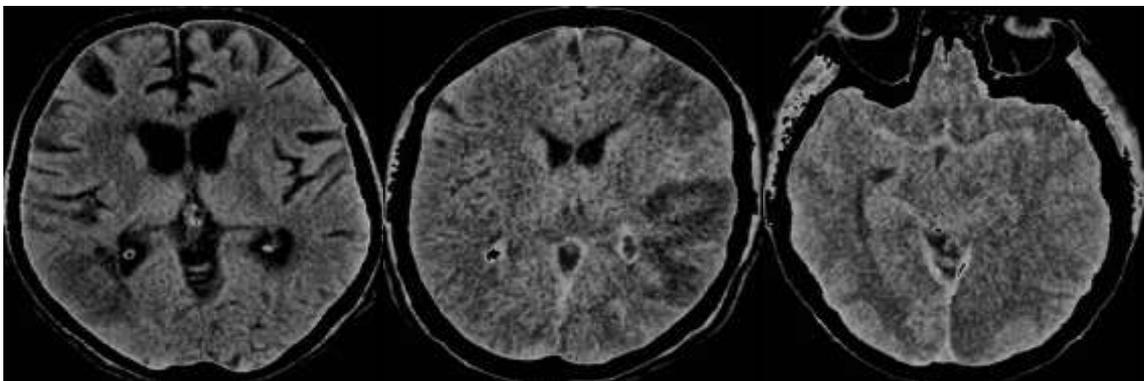
Gambar 4.8 Biorthogonal 1.1



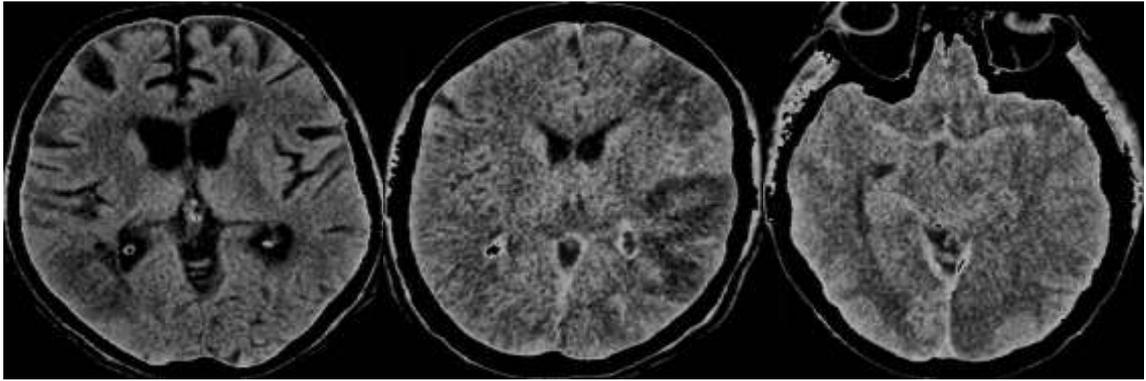
Gambar 4.9 Biorthogonal 2.2



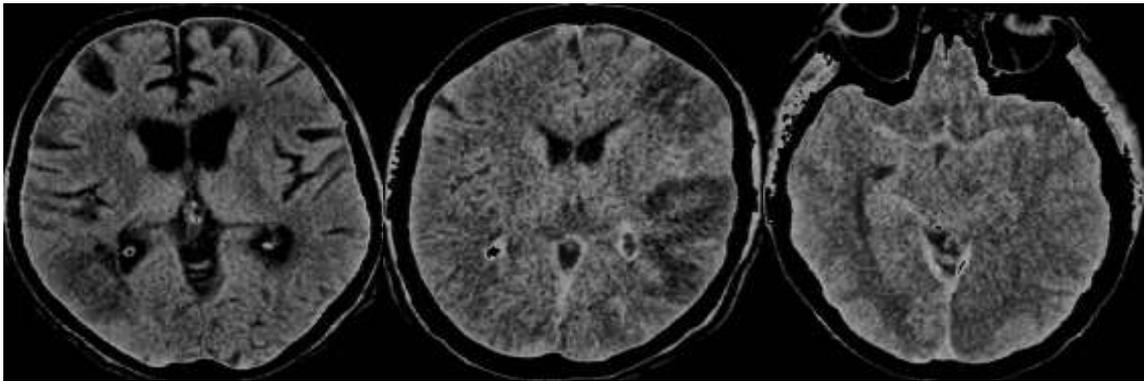
Gambar 4.10 Biorthogonal 3.3



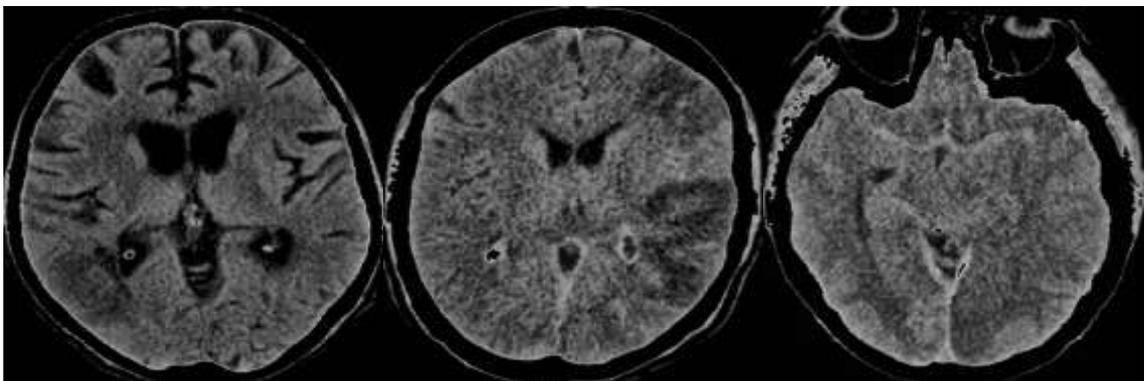
Gambar 4.11 Biorthogonal 4.4



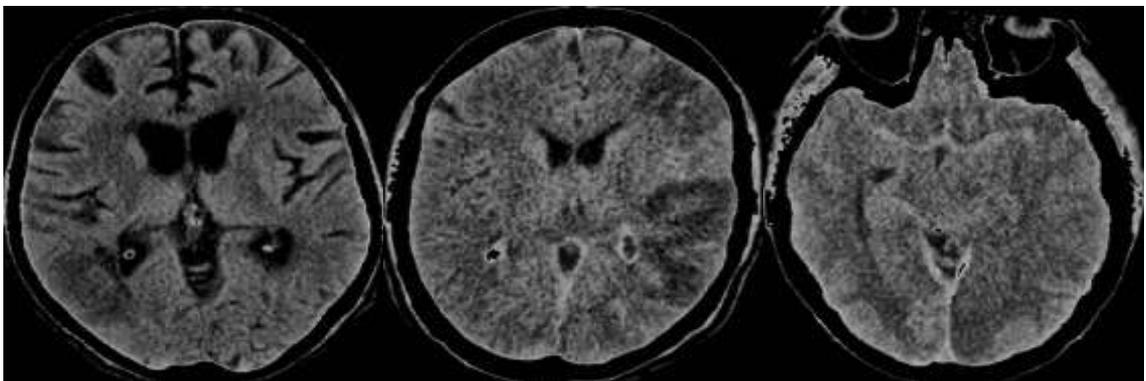
Gambar 4.12 Biorthogonal 5.5



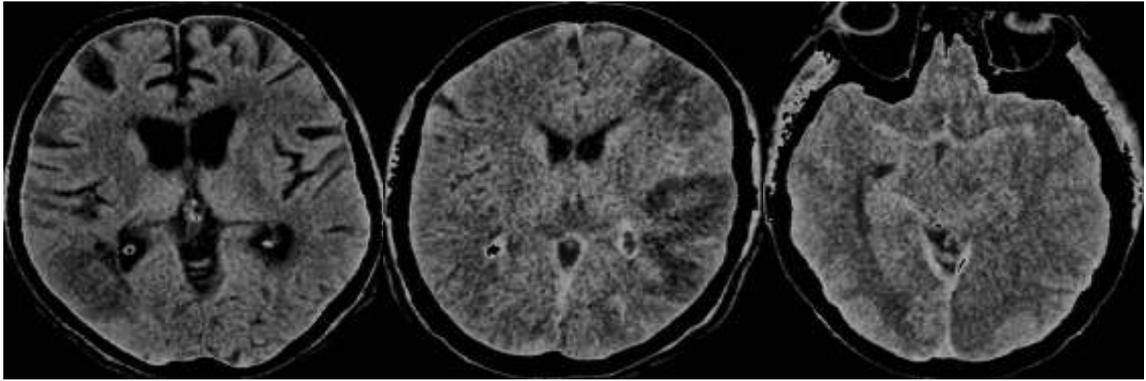
Gambar 4.13 Biorthogonal 6.8



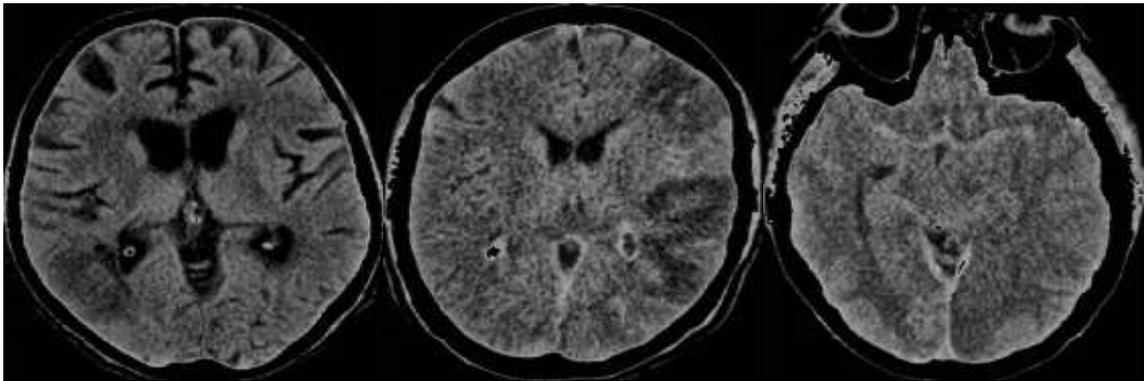
Gambar 4.14 Coiflet 1



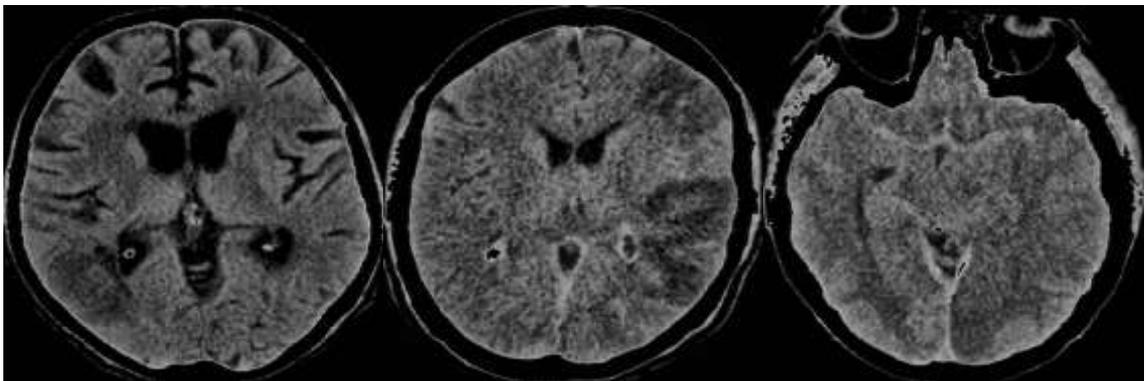
Gambar 4.15 Coiflet 2



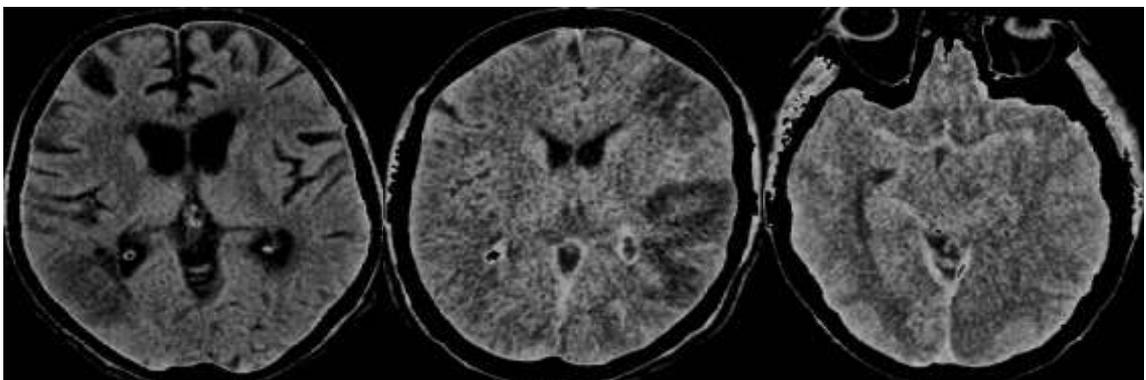
Gambar 4.16 Coiflet 3



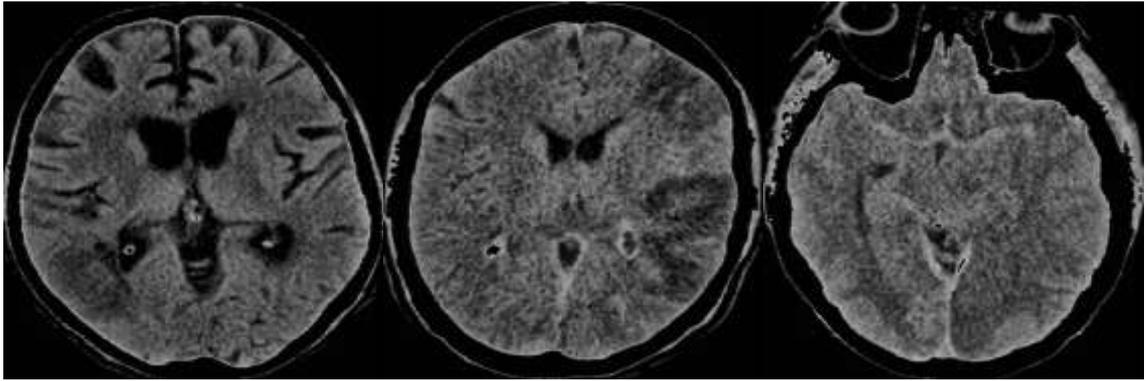
Gambar 4.17 Coiflet 4



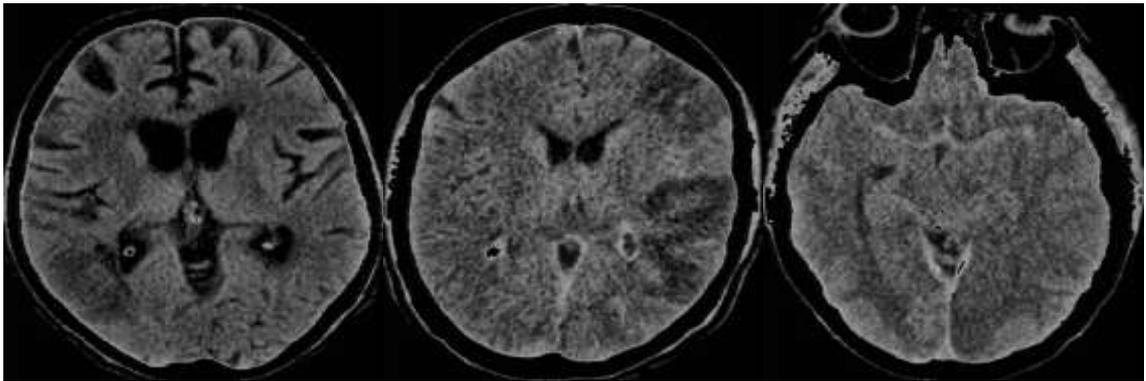
Gambar 4.18 Coiflet 5



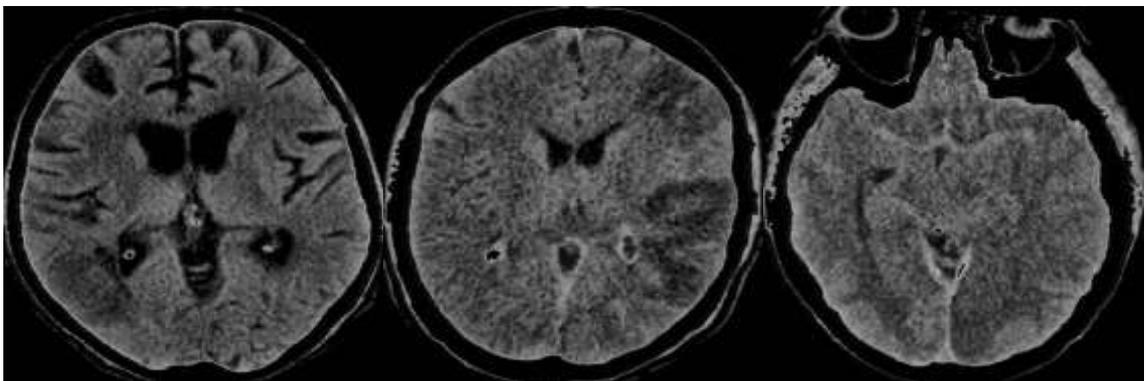
Gambar 4.19 Daubechies 1



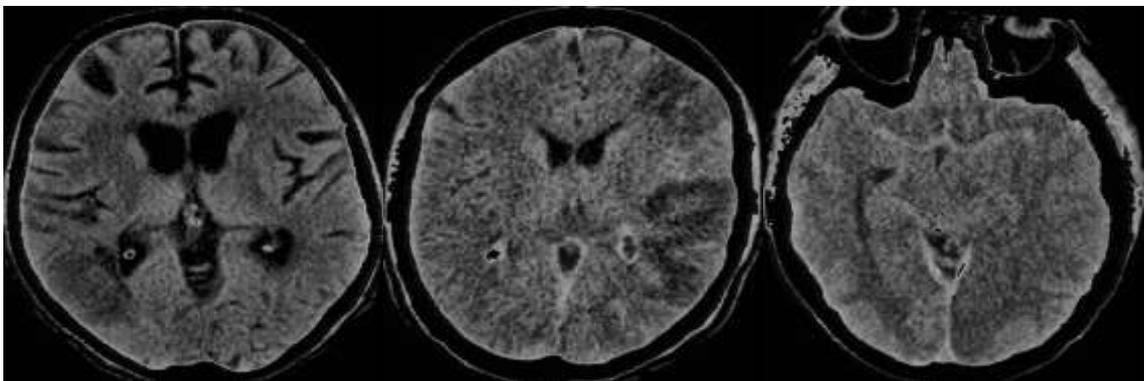
Gambar 4.20 Daubechies 5



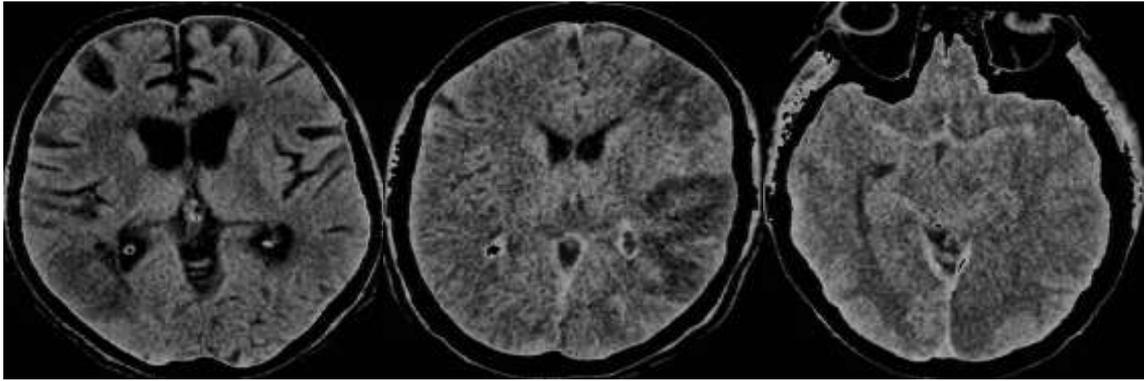
Gambar 4.21 Daubechies 10



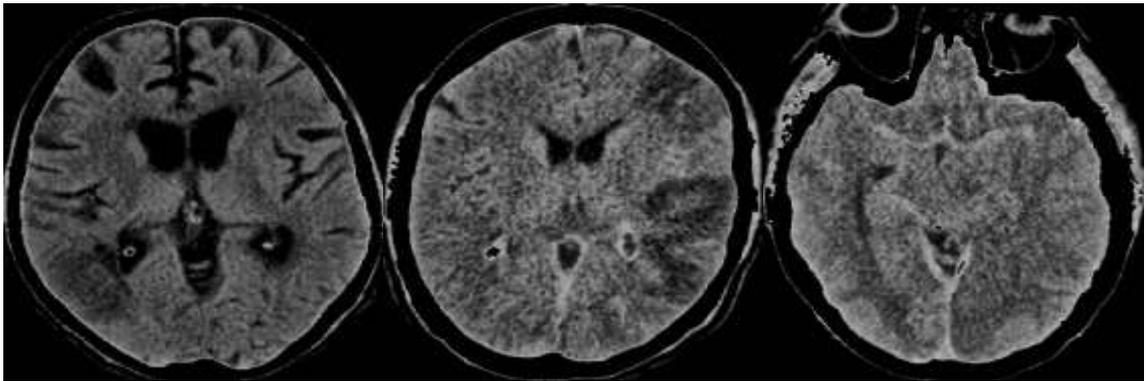
Gambar 4.22 Daubechies 15



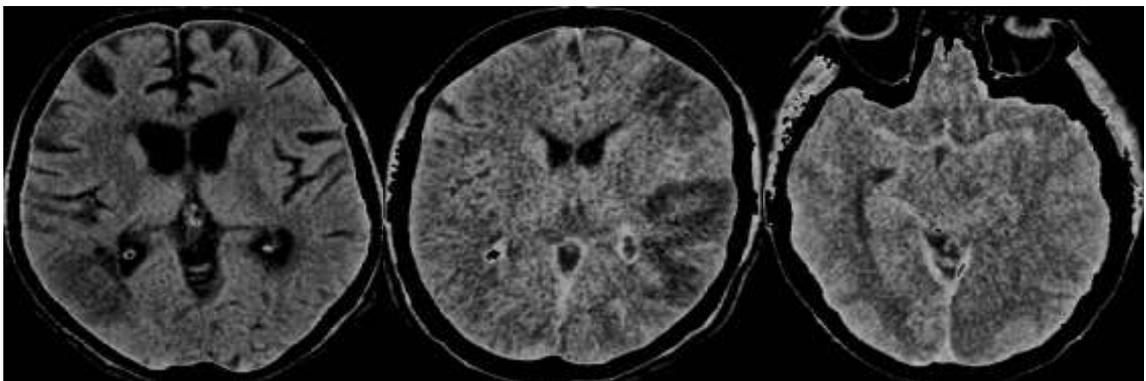
Gambar 4.23 Daubechies 20



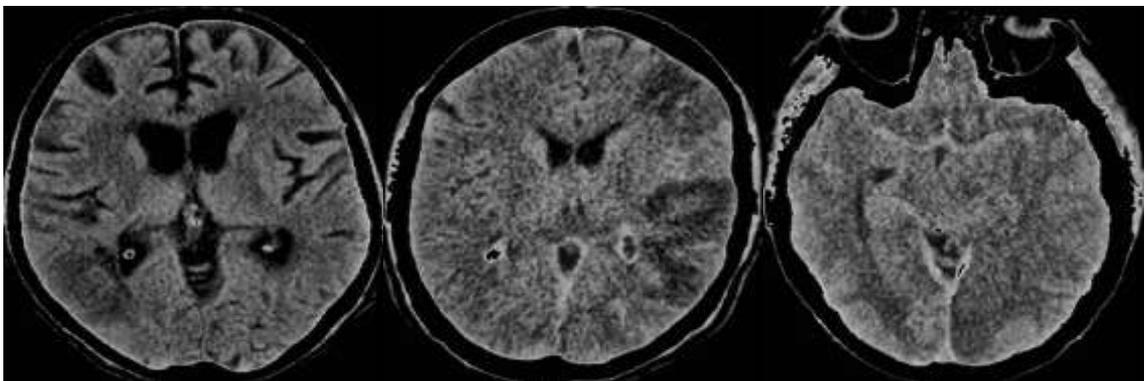
Gambar 4.24 Discrete meyer



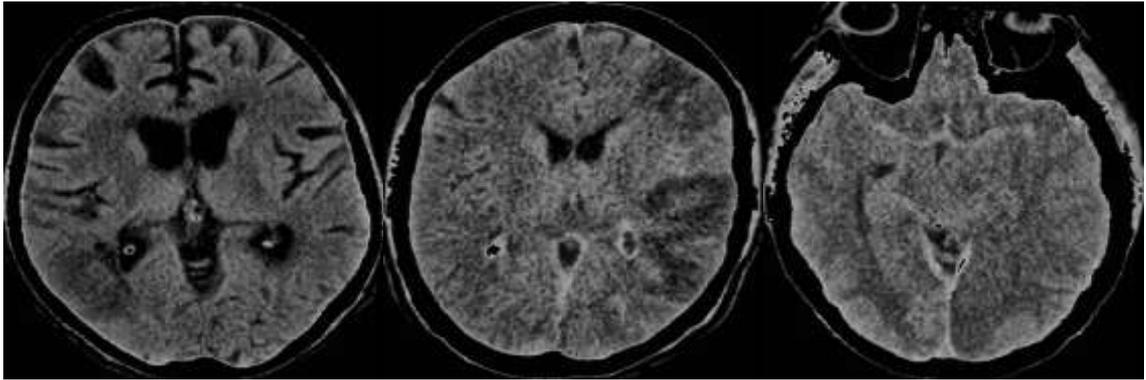
Gambar 4.25 Haar



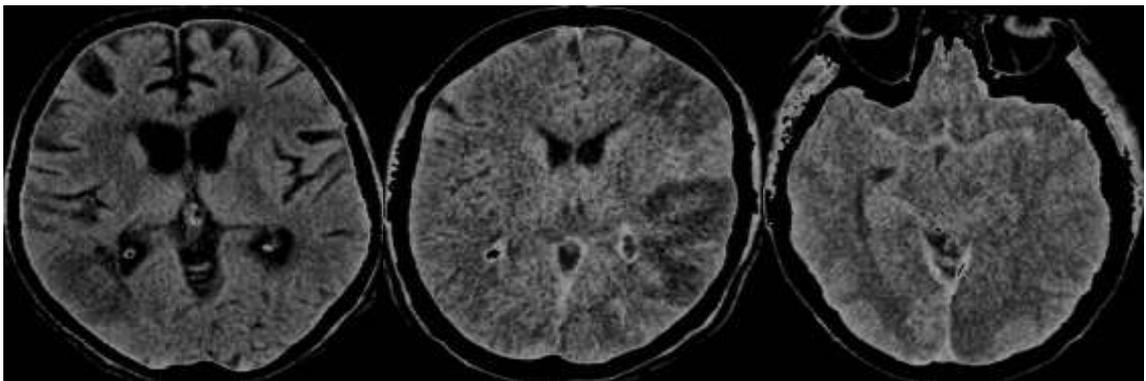
Gambar 4.26 Reverse biorthogonal 1.1



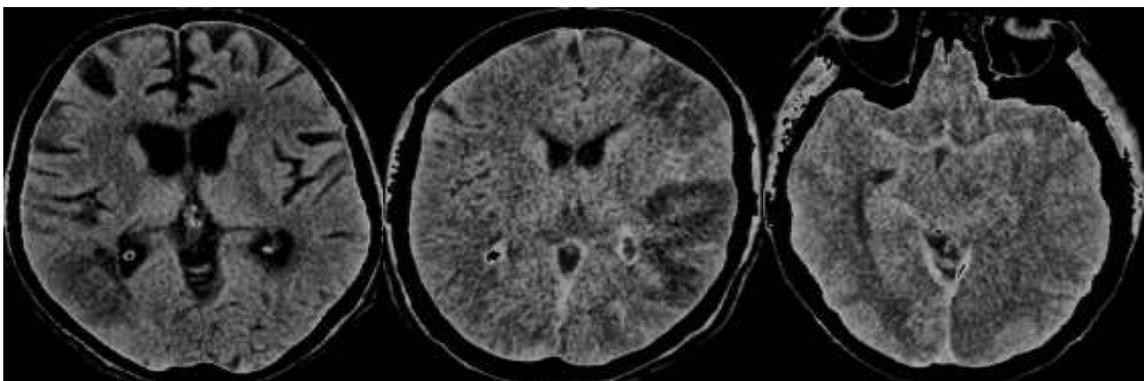
Gambar 4.27 Reverse biorthogonal 3.3



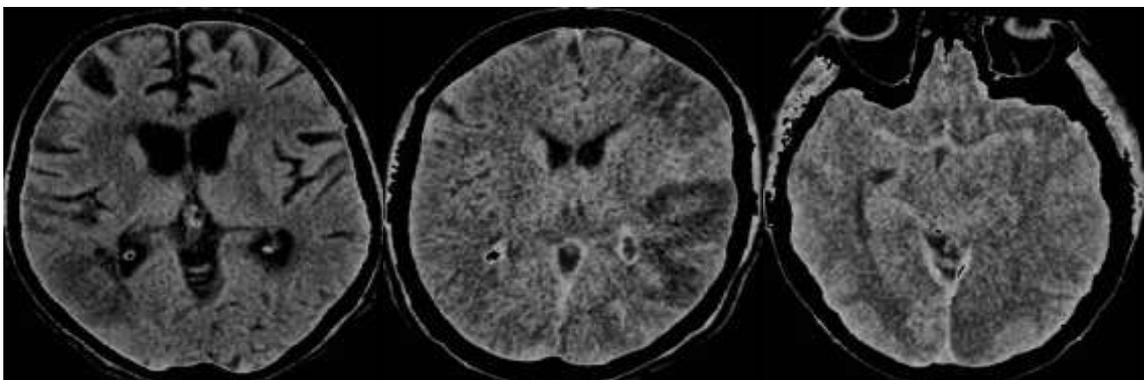
Gambar 4.28 Reverse biorthogonal 4.4



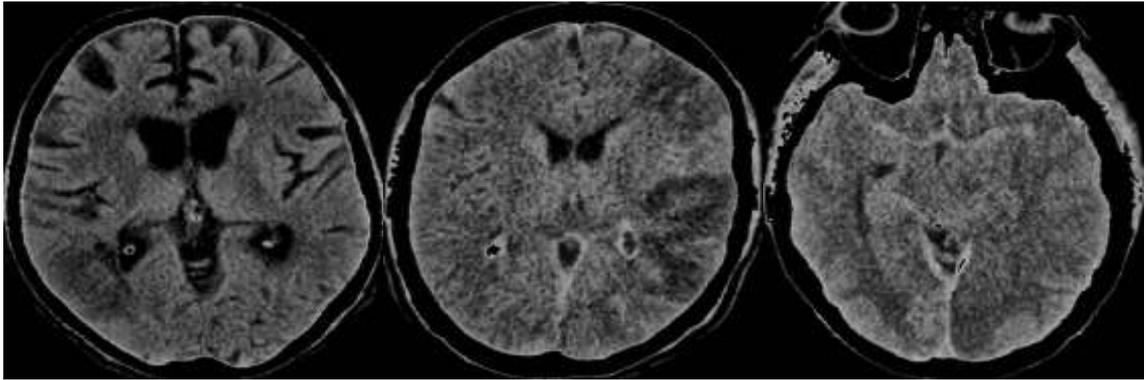
Gambar 4.29 Reverse biorhogonal 5.5



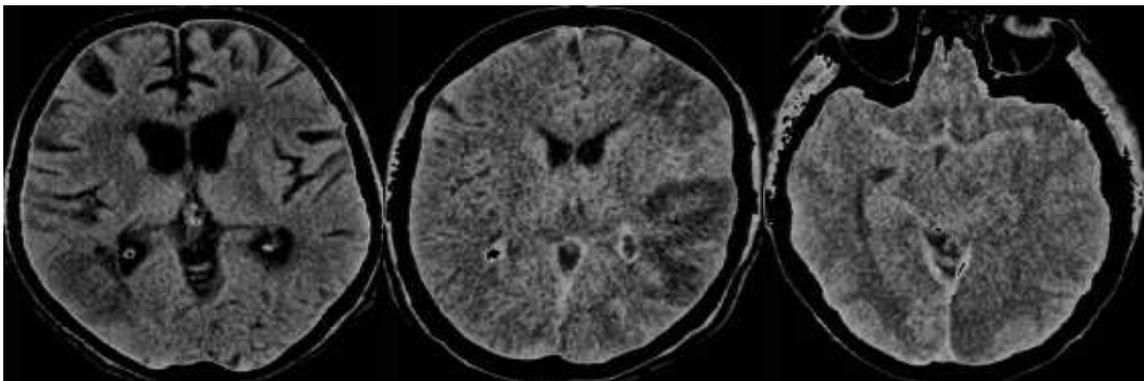
Gambar 4.30 Reverse biorhogonal 6.8



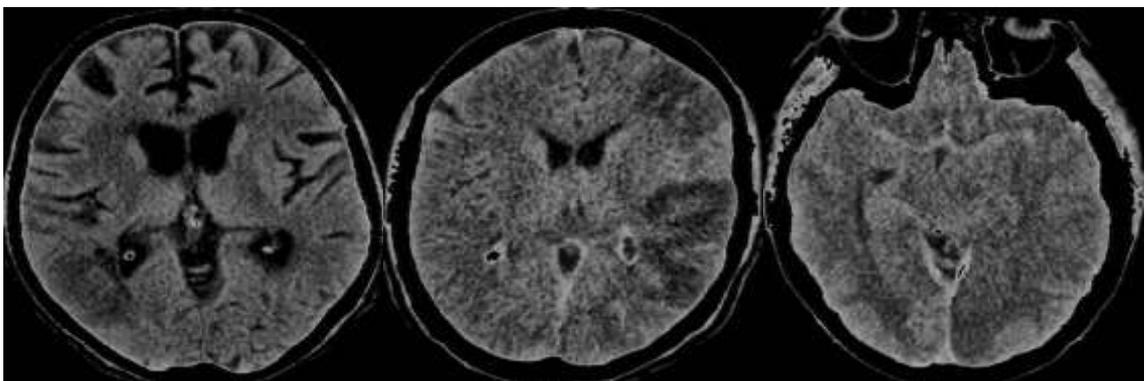
Gambar 4.31 Symlet 2



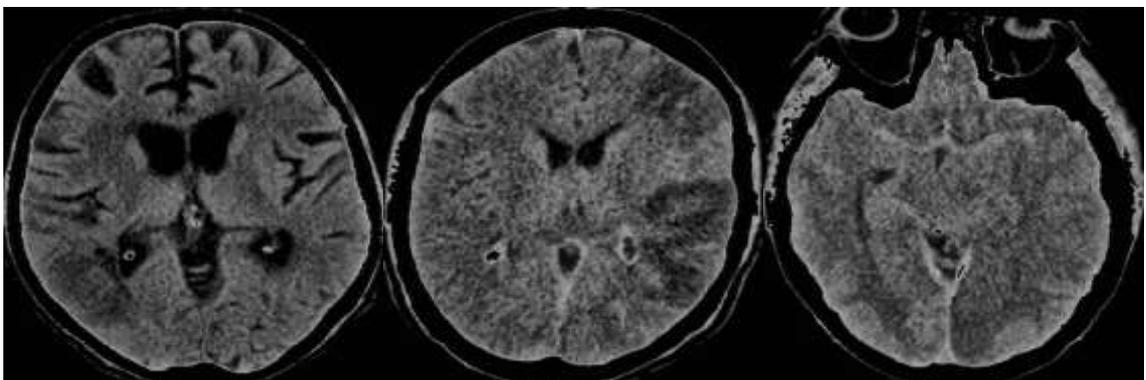
Gambar 4.32 Symlet 5



Gambar 4.33 Symlet 10



Gambar 4.34 Symlet 15



Gambar 4.35 Symlet 20

4.2 Pelatihan CNN

Metode CNN (*Convolutional Neural Network*) yang digunakan berfungsi untuk melakukan segmentasi semantik pada gambar. Pada proses CNN masukan adalah berupa *set of images* dan *set of ground truth* sebagai *desired output* berukuran 512x512 dan menghasilkan keluaran berupa gambar dengan ukuran piksel yang sama. Pada pelatihan CNN, jumlah data sangat berpengaruh terhadap performansi CNN. Oleh sebab itu, perlu dilakukan augmentasi terhadap data. Pada penelitian ini, metode augmentasi yang digunakan adalah transformasi geometrik. Transformasi geometrik yang dilakukan yaitu *flip*, rotasi dan distorsi dengan probabilitas 0.4, 0.4 dan 0.5 dan jumlah data hasil augmentasi 1000 sehingga pada akhir augmentasi diperoleh data sebanyak 1000 citra dan 1000 *mask*. Selanjutnya citra-citra ini dibagi secara random menjadi data latih dan data uji dengan perbandingan 75:25 sehingga diperoleh 750 data latih dan 250 data uji. Data latih ini digunakan pada proses pembelajaran sedangkan data uji digunakan sebagai proses pengujian sistem CNN secara keseluruhan serta untuk menghasilkan gambar prediksi hasil segmentasi.

Proses pelatihan untuk mendapatkan model *weight* dengan arsitektur UNET menggunakan *library segmentation-models* dengan *backend Tensorflow* serta memanfaatkan komputasi parallel *Graphical Processing Unit (GPU)* yang tersedia pada *Google Colab*. *Dataset* yang digunakan merupakan keluaran bagian *image preprocessing* yang telah disimpan dalam ekstensi *npz*. Keseluruhan *dataset* tersebut dibagi menjadi data pelatihan dan data pengujian.

Arsitektur yang diaplikasikan pada proses segmentasi ini ditunjukkan pada gambar 3.3 dan dijabarkan pada Tabel 4.1 dengan keterangan jenis *layer*, tipe, ukuran, dan parameter pada setiap *layer*. Arsitektur ini terdiri dari 4 *stage encoder* dan 4 *stage decoders*. Pada proses *encoder* atau ekstraksi fitur, digunakan *backbone Resnet (residual network)*. Resnet merupakan arsitektur CNN yang menambahkan *skip connections* pada setiap *encoder* dan telah teruji mampu meningkatkan performansi CNN khususnya dalam *vanishing moment*, yaitu keadaan dimana CNN berhenti mengoptimalisasi *weight* sehingga proses *learning* tidak menunjukkan kenaikan akurasi. Sedangkan pada *stage decoder* digunakan arsitektur U-Net biasa tanpa ada tambahan *skip connections*.

Tabel 4.1 Model summary

<i>Part</i>	<i>Type</i>	<i>Stage</i>	<i>Output Shape</i>	<i>Parameter</i>
<i>Input</i>	<i>Input</i>		(None, None, None, 3)	0
<i>Contracting Path</i>	<i>Batch Normalization</i>	0	(None, None, None, 3)	9
	<i>Zero Padding</i>	0	(None, None, None, 3)	0
	<i>Convolution</i>	0	(None, None, None, 6)	9408
	<i>Batch Normalization</i>	0	(None, None, None, 6)	256
	<i>Activation</i>	0	(None, None, None, 6)	0
	<i>Zero Padding</i>	0	(None, None, None, 6)	0
	<i>Max Pooling</i>	0	(None, None, None, 6)	0
	<i>Batch Normalization</i>	0	(None, None, None, 6)	256
	<i>Activation</i>	1	(None, None, None, 6)	0
	<i>Zero Padding</i>	1	(None, None, None, 6)	0

<i>Convolution</i>	1	(None, None, None, 6)	36864
<i>Batch Normalization</i>	1	(None, None, None, 6)	256
<i>Activation</i>	1	(None, None, None, 6)	0
<i>Zero Padding</i>	1	(None, None, None, 6)	0
<i>Convolution</i>	1	(None, None, None, 6)	36864
<i>Convolution</i>	1	(None, None, None, 6)	4096
<i>Adder</i>	1	(None, None, None, 6)	0
<i>Batch Normalization</i>	1	(None, None, None, 6)	256
<i>Activation</i>	1	(None, None, None, 6)	0
<i>Zero Padding</i>	1	(None, None, None, 6)	0
<i>Convolution</i>	1	(None, None, None, 6)	36864
<i>Batch Normalization</i>	1	(None, None, None, 6)	256
<i>Activation</i>	1	(None, None, None, 6)	0
<i>Zero Padding</i>	1	(None, None, None, 6)	0
<i>Convolution</i>	1	(None, None, None, 6)	36864
<i>Adder</i>	1	(None, None, None, 6)	0
<i>Batch Normalization</i>	1	(None, None, None, 6)	256
<i>Activation</i>	1	(None, None, None, 6)	0
<i>Zero Padding</i>	1	(None, None, None, 6)	0
<i>Convolution</i>	1	(None, None, None, 6)	36864
<i>Batch Normalization</i>	1	(None, None, None, 6)	256
<i>Activation</i>	1	(None, None, None, 6)	0
<i>Zero Padding</i>	1	(None, None, None, 6)	0
<i>Convolution</i>	1	(None, None, None, 6)	36864
<i>Adder</i>	1	(None, None, None, 6)	0
<i>Batch Normalization</i>	2	(None, None, None, 6)	256
<i>Activation</i>	2	(None, None, None, 6)	0
<i>Zero Padding</i>	2	(None, None, None, 6)	0
<i>Convolution</i>	2	(None, None, None, 1)	73728
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0
<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Convolution</i>	2	(None, None, None, 1)	8192
<i>Adder</i>	2	(None, None, None, 1)	0
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0
<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0

<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Adder</i>	2	(None, None, None, 1)	0
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0
<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0
<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Adder</i>	2	(None, None, None, 1)	0
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0
<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Batch Normalization</i>	2	(None, None, None, 1)	512
<i>Activation</i>	2	(None, None, None, 1)	0
<i>Zero Padding</i>	2	(None, None, None, 1)	0
<i>Convolution</i>	2	(None, None, None, 1)	147456
<i>Adder</i>	2	(None, None, None, 1)	0
<i>Batch Normalization</i>	3	(None, None, None, 1)	512
<i>Activation</i>	3	(None, None, None, 1)	0
<i>Zaro Padding</i>	3	(None, None, None, 1)	0
<i>Convolution</i>	3	(None, None, None, 2)	294912
<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
<i>Activation</i>	3	(None, None, None, 2)	0
<i>Zero Padding</i>	3	(None, None, None, 2)	0
<i>Convolution</i>	3	(None, None, None, 2)	589824
<i>Convolution</i>	3	(None, None, None, 2)	32768
<i>Adder</i>	3	(None, None, None, 2)	0
<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
<i>Activation</i>	3	(None, None, None, 2)	0
<i>Zero Padding</i>	3	(None, None, None, 2)	0
<i>Convolution</i>	3	(None, None, None, 2)	589824
<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
<i>Activation</i>	3	(None, None, None, 2)	0
<i>Zero Padding</i>	3	(None, None, None, 2)	0
<i>Convolution</i>	3	(None, None, None, 2)	589824
<i>Adder</i>	3	(None, None, None, 2)	0
<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
<i>Activation</i>	3	(None, None, None, 2)	0
<i>Zero Padding</i>	3	(None, None, None, 2)	0

	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Adder</i>	3	(None, None, None, 2)	0
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Adder</i>	3	(None, None, None, 2)	0
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Adder</i>	3	(None, None, None, 2)	0
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	3	(None, None, None, 2)	0
	<i>Zero Padding</i>	3	(None, None, None, 2)	0
	<i>Convolution</i>	3	(None, None, None, 2)	589824
	<i>Adder</i>	3	(None, None, None, 2)	0
	<i>Batch Normalization</i>	3	(None, None, None, 2)	1024
	<i>Activation</i>	4	(None, None, None, 2)	0
	<i>Zero Padding</i>	4	(None, None, None, 2)	0
	<i>Convolution</i>	4	(None, None, None, 5)	1179648
	<i>Batch Normalization</i>	4	(None, None, None, 5)	2048
	<i>Activation</i>	4	(None, None, None, 5)	0
	<i>Zero Padding</i>	4	(None, None, None, 5)	0
	<i>Convolution</i>	4	(None, None, None, 5)	2359296
	<i>Convolution</i>	4	(None, None, None, 5)	131072

	<i>Adder</i>	4	(None, None, None, 5)	0
	<i>Batch Normalization</i>	4	(None, None, None, 5)	2048
	<i>Activation</i>	4	(None, None, None, 5)	0
	<i>Zero Padding</i>	4	(None, None, None, 5)	0
	<i>Convolution</i>	4	(None, None, None, 5)	2359296
	<i>Batch Normalization</i>	4	(None, None, None, 5)	2048
	<i>Activation</i>	4	(None, None, None, 5)	0
	<i>Zero Padding</i>	4	(None, None, None, 5)	0
	<i>Convolution</i>	4	(None, None, None, 5)	2359296
	<i>Adder</i>	4	(None, None, None, 5)	0
	<i>Batch Normalization</i>	4	(None, None, None, 5)	2048
	<i>Acivation</i>	4	(None, None, None, 5)	0
	<i>Zero Padding</i>	4	(None, None, None, 5)	0
	<i>Convolution</i>	4	(None, None, None, 5)	2359296
	<i>Batch Normalization</i>	4	(None, None, None, 5)	2048
	<i>Activation</i>	4	(None, None, None, 5)	0
	<i>Zero Padding</i>	4	(None, None, None, 5)	0
	<i>Convolution</i>	4	(None, None, None, 5)	2359296
	<i>Adder</i>	4	(None, None, None, 5)	0
<i>Bottle neck</i>	<i>Batch Normalization</i>		(None, None, None, 5)	2048
	<i>Activation</i>		(None, None, None, 5)	0
	<i>Upsampling</i>		(None, None, None, 5)	0
<i>Expanding path</i>	<i>Concatenate</i>	0	(None, None, None, 7)	0
	<i>Conv. Layer</i>	0	(None, None, None, 2)	1769472
	<i>Batch Normalization</i>	0	(None, None, None, 2)	1024
	<i>Activation</i>	0	(None, None, None, 2)	0
	<i>Convolution</i>	0	(None, None, None, 2)	589824
	<i>Batch Normalization</i>	0	(None, None, None, 2)	1024
	<i>Activation</i>	0	(None, None, None, 2)	0
	<i>Upsampling</i>	0	(None, None, None, 2)	0
	<i>Concatenate</i>	1	(None, None, None, 3)	0
	<i>Convolution</i>	1	(None, None, None, 1)	442368
	<i>Batch Normalization</i>	1	(None, None, None, 1)	512
	<i>Activation</i>	1	(None, None, None, 1)	0
	<i>Convolution</i>	1	(None, None, None, 1)	147456
	<i>Batch Normalization</i>	1	(None, None, None, 1)	512
	<i>Activation</i>	1	(None, None, None, 1)	0
	<i>Upsampling</i>	1	(None, None, None, 1)	0
	<i>Concatenate</i>	2	(None, None, None, 1)	0
	<i>Convolution</i>	2	(None, None, None, 6)	110592
	<i>Batch Normalization</i>	2	(None, None, None, 6)	256
	<i>Activation</i>	2	(None, None, None, 6)	0

	<i>Convolution</i>	2	(None, None, None, 6)	36864
	<i>Batch Normalization</i>	2	(None, None, None, 6)	256
	<i>Activation</i>	2	(None, None, None, 6)	0
	<i>Upsampling</i>	2	(None, None, None, 6)	0
	<i>Concatenate</i>	3	(None, None, None, 1)	0
	<i>Convolution</i>	3	(None, None, None, 3)	36864
	<i>Batch Normalization</i>	3	(None, None, None, 3)	128
	<i>Activation</i>	3	(None, None, None, 3)	0
	<i>Convolution</i>	3	(None, None, None, 3)	9216
	<i>Batch Normalization</i>	3	(None, None, None, 3)	128
	<i>Activation</i>	3	(None, None, None, 3)	0
	<i>Upsampling</i>	3	(None, None, None, 3)	0
	<i>Convolution</i>	4	(None, None, None, 1)	4608
	<i>Batch Normalization</i>	4	(None, None, None, 1)	64
	<i>Activation</i>	4	(None, None, None, 1)	0
	<i>Convolution</i>	4	(None, None, None, 1)	2304
	<i>Batch Normalization</i>	4	(None, None, None, 1)	64
	<i>Activation</i>	4	(None, None, None, 1)	0
	<i>Convolution</i>	4	(None, None, None, 1)	145
<i>Output</i>	<i>Activation</i>		(None, None, None, 1)	0

Total parameters: 24,456,154

Trainable parameters: 24,438,804

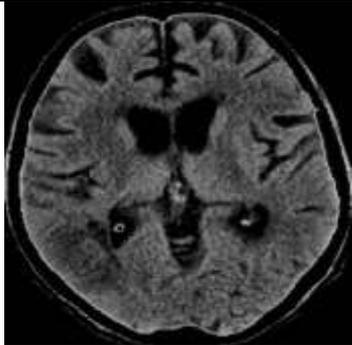
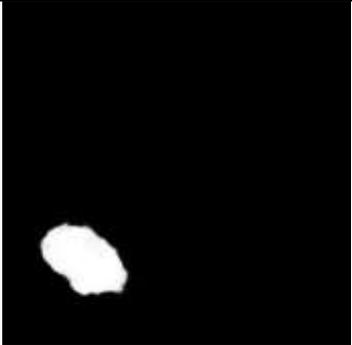
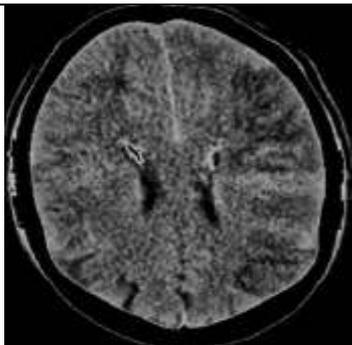
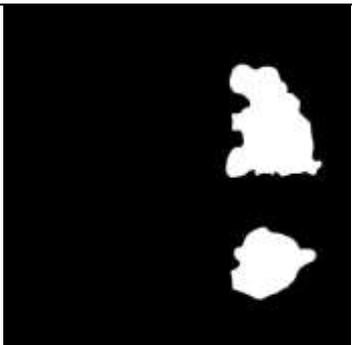
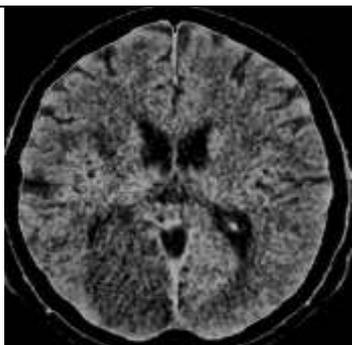
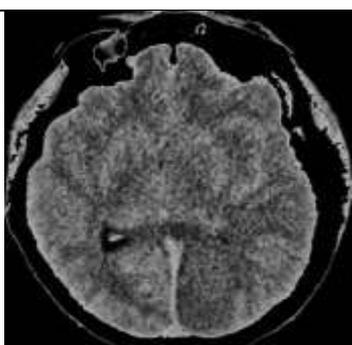
Non-trainable parameters: 17,350

Proses *learning* dimulai dengan melakukan ekstraksi fitur atau encoder. Proses ini diawali dengan melakukan konvolusi terhadap citra input dengan menggunakan kernel berukuran 3x3 berisi nilai *weight* random yang diinisialisasi menggunakan *kernel initializer* *he_uniform*. Nilai *weight* yang digunakan pada penelitian ini merupakan hasil *transfer learning* dari dataset Imagenet sehingga *weight* awal sudah mengetahui fitur umum dari sebuah citra.

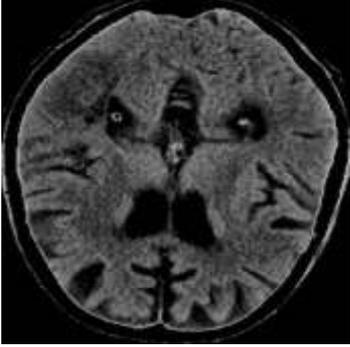
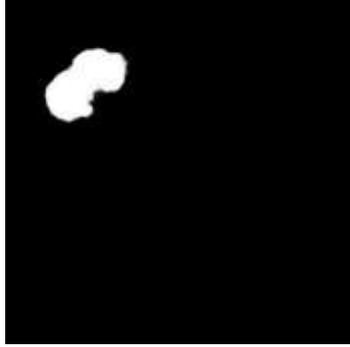
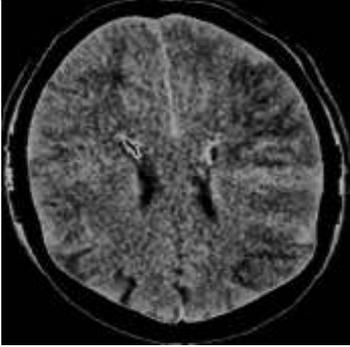
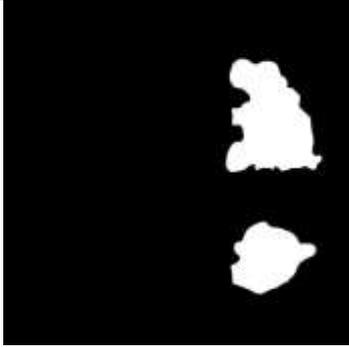
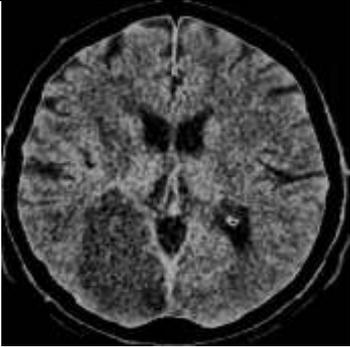
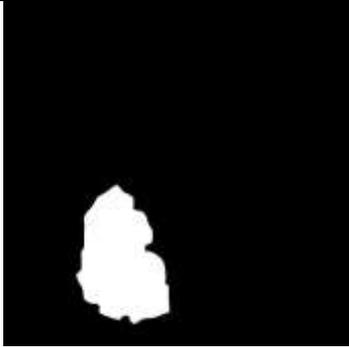
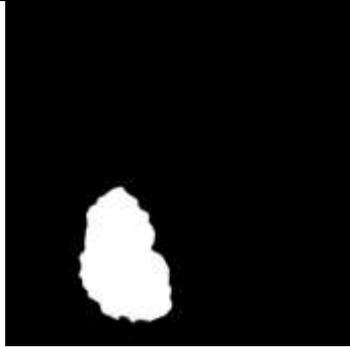
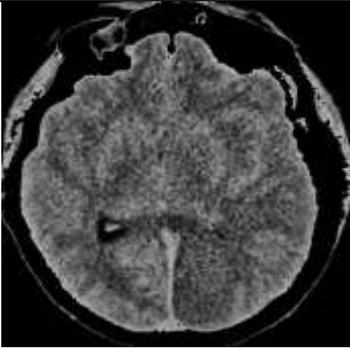
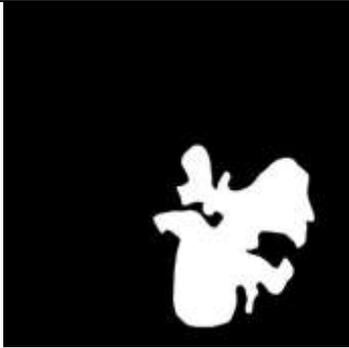
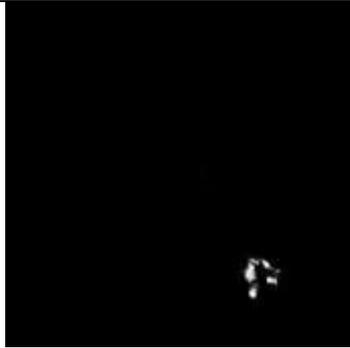
Model ini dilatih secara berulang dengan melakukan *backpropagation* pada hasil akhir setiap *epoch*. *Backpropagation* adalah perhitungan yang dilakukan untuk menentukan *error* yang terjadi antara hasil yang diharapkan dengan hasil yang diperoleh dari pelatihan. *Epoch* adalah ketika semua dataset melewati satu kali proses *forward* dan *backward* melewati semua *node neural network* 1 kali. Jumlah *epoch* sangat menentukan performansi CNN sehingga pada penelitian ini dilakukan percobaan menggunakan jumlah epoch 10, 50, dan 100 dengan *optimizer* ADAM (*Adaptive Moment*). ADAM adalah algoritma optimasi pengganti untuk *stochastic gradient descent* untuk *training model*. *Backpropagation* dilakukan dengan mengamati *loss binary crossentropy* dan *metrics Intersection of Union (IoU)*. *Loss* menghitung jumlah *error* antara hasil yang diharapkan dengan hasil yang diperoleh, sedangkan *IoU* menghitung persentase jumlah piksel yang *overlapping* antara hasil yang diharapkan dengan hasil yang diperoleh. Pengamatan *loss* dan *accuracy* pada saat proses pembelajaran model CNN dalam mendapatkan *weight* menggunakan *binary crossentropy* dan *metrics Intersection of Union (IoU)*.

Dari proses *training* ini diperoleh sebuah *model weight* yang dapat melakukan segmentasi terhadap lesi *hypodense* secara otomatis. Untuk menguji hasil segmentasi yang diperoleh, digunakan *Dice Coefficient Similarity* yang menghitung kemiripan antara hasil segmentasi dengan hasil yang seharusnya (*ground truth*). Hasil segmentasi yang diperoleh pada melalui proses pelatihan dengan jumlah *epoch* 10, 50 dan 100 dapat dilihat pada Tabel 4.2 hingga 4.4.

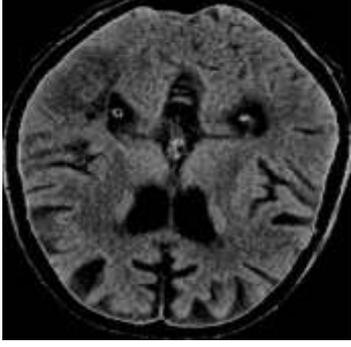
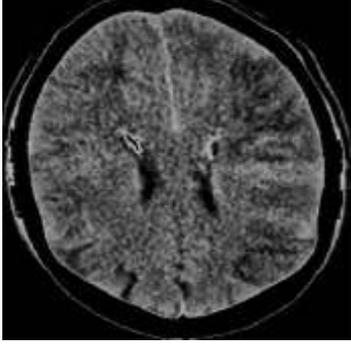
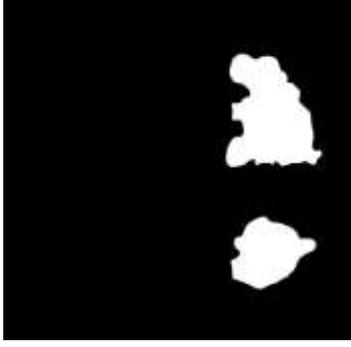
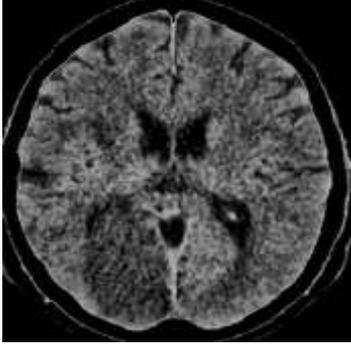
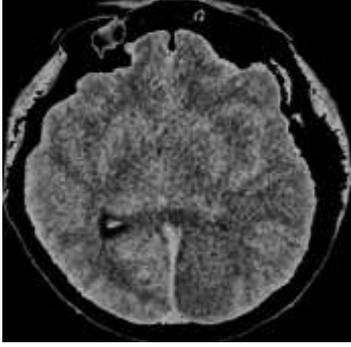
Tabel 4.2 Hasil segmentasi CNN dengan *epoch* 10

<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		

Tabel 4.3 Hasil segmentasi CNN dengan *epoch* 50

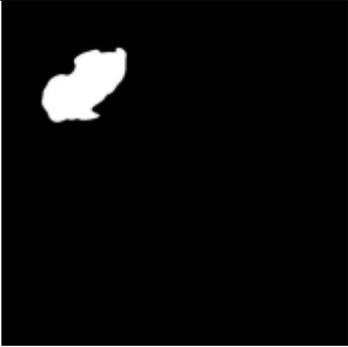
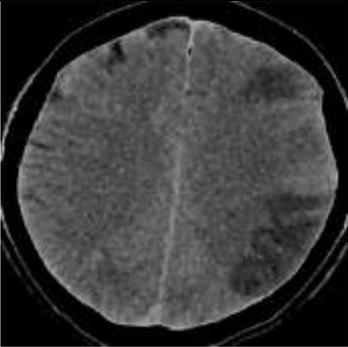
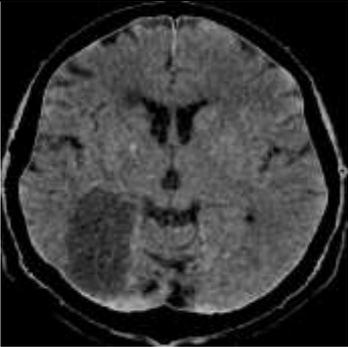
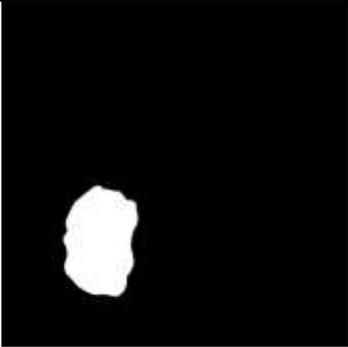
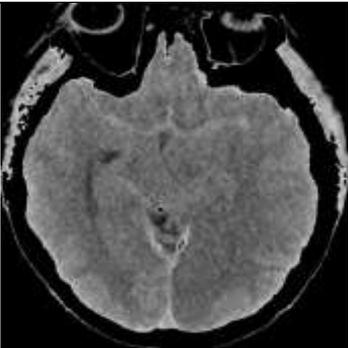
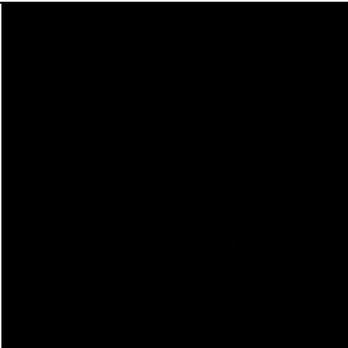
<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		

Tabel 4.4 Hasil segmentasi CNN dengan *epoch* 100

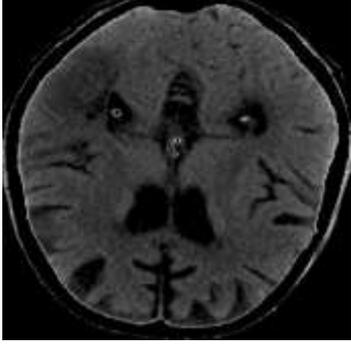
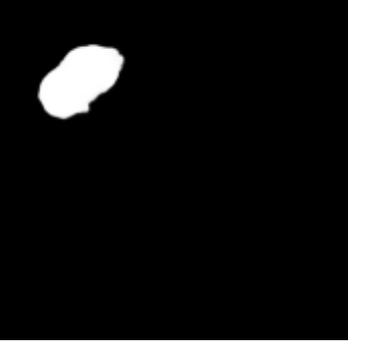
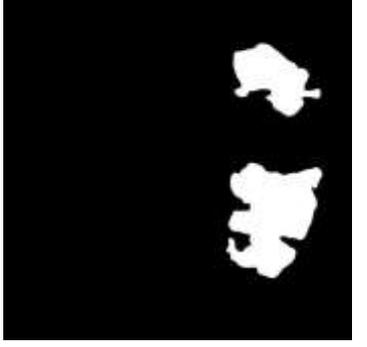
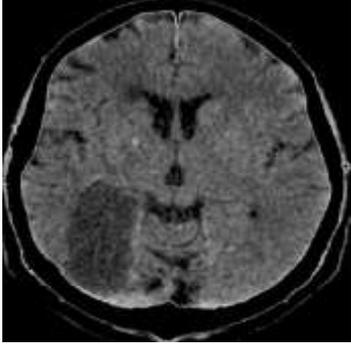
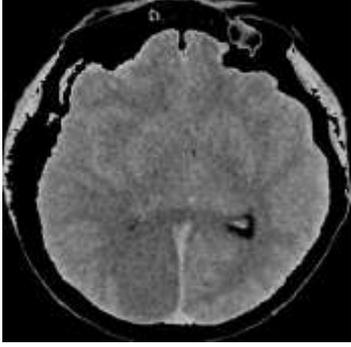
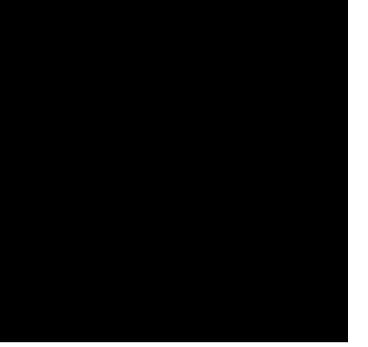
<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		

Untuk menguji efektifitas peningkatan citra terhadap proses pelatihan, dilakukan pula percobaan dengan menggunakan citra tanpa peningkatan sebagai input pelatihan. Hasil yang diperoleh pada percobaan ini dapat dilihat pada Tabel 4.5 hingga 4.7.

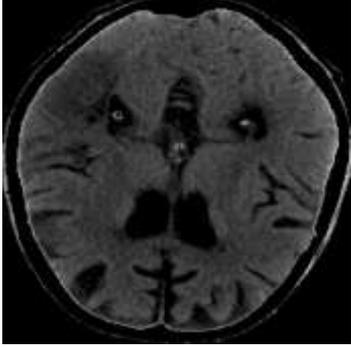
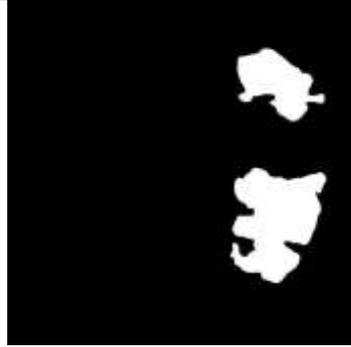
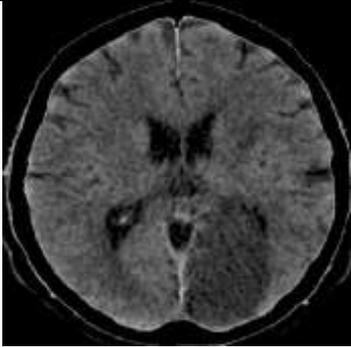
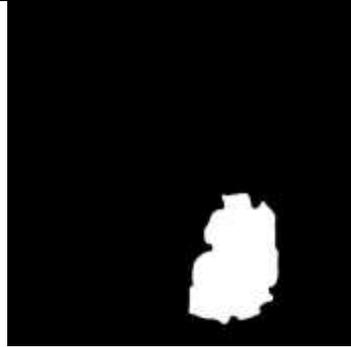
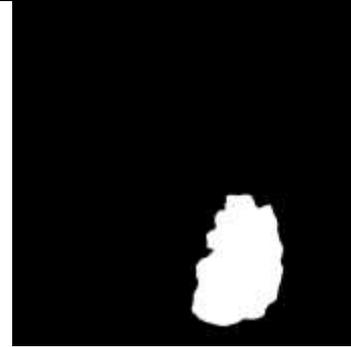
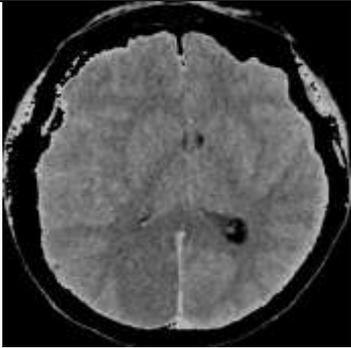
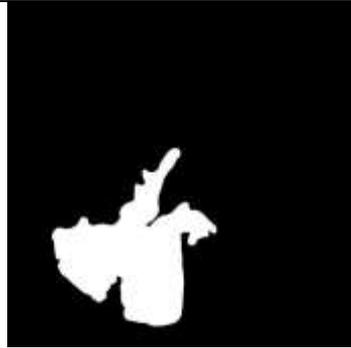
Tabel 4.5 Hasil segmentasi CNN *epoch* 10 dengan *input* citra tanpa peningkatan

<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		

Tabel 4.6 Hasil segmentasi CNN *epoch* 50 dengan *input* citra tanpa peningkatan

<i>Test Case</i>	Ground Truth	Predicted Image
		
		
		
		

Tabel 4.7 Hasil segmentasi CNN *epoch* 100 dengan input citra tanpa peningkatan

<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		

BAB 5 PEMBAHASAN

5.1. *Preprocessing*

Pada tahap pengujian, data citra sebagai dari keluaran *preprocessing* digunakan sebagai masukan untuk pengolahan metode selanjutnya. Pada tahap pembahasan ini akan dijelaskan mengenai hasil yang dikeluarkan dan pengaruh nya terhadap sistem untuk mendeteksi lesi *hypodense*.

5.1.1. *Grayscale*

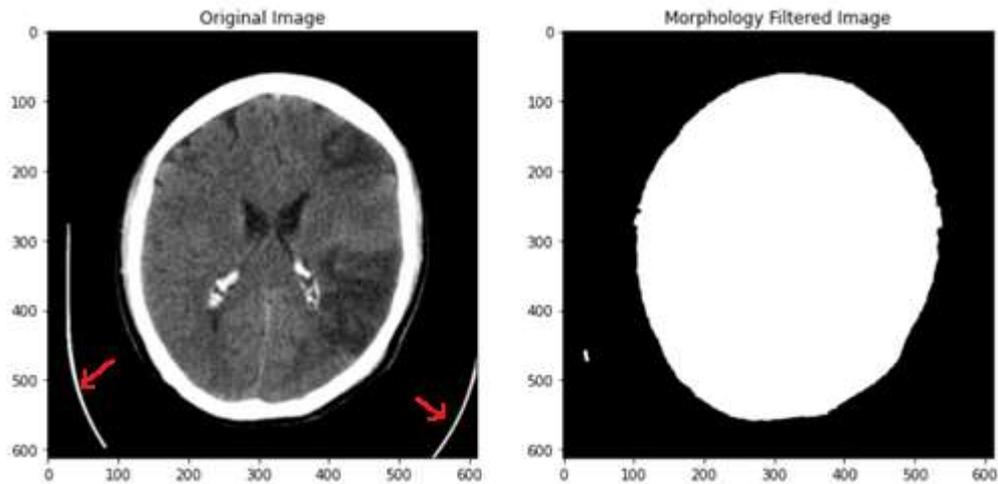
Tahap *grayscale* dilakukan untuk mengubah citra awal yang memiliki intensitas RGB menjadi citra dengan intensitas *grayscale*. RGB merupakan tipe intensitas citra dengan 3 kanal, yaitu *red*, *blue*, dan *green*. Untuk mengolah citra RGB, setiap operasi dilakukan terhadap setiap kanal, sehingga komputasi akan semakin kompleks. Oleh sebab itu, perlu dilakukan *grayscale* pada citra untuk efektivitas komputasi.

Gambar 4.2 menunjukkan citra hasil *grayscale*. Secara visual tidak ada perbedaan antar kedua citra dikarenakan citra CT scan hanya terdiri atas intensitas 0-255, namun jumlah *channel* dapat dicek dengan melihat ukuran citra. Pada Gambar 4.2 terlihat bahwa dimensi dari citra yang semula terdiri atas *array* 3 dimensi [lebar, panjang, *channel*] berubah menjadi *array* 2 dimensi [lebar, panjang].

5.1.2. **Filter median dan morfologi**

Filtering merupakan salah satu metode dasar pengolahan citra yang bertujuan untuk mengurangi *noise* pada citra. Disini digunakan 2 filter yaitu filter median dan morfologi. Filter median diterapkan untuk mengurangi *noise* berupa *salt and pepper noise* dan filter morfologi diterapkan untuk menghilangkan area diluar ROI dan menetapkan batas untuk proses *autocropping*.

Pada Gambar 4.3 terlihat hasil dari filter median yang menunjukkan bahwa *noise* yang terdapat pada citra menjadi lebih berkurang. Dengan berkurangnya *noise* pada citra, citra menjadi lebih jelas. Sedangkan pada Gambar 4.4 menunjukkan hasil dari filter morfologi. Tahap pertama pada proses ini ialah *thresholding*, tahap ini dapat mengubah citra *grayscale* menjadi citra biner atau hitam putih. Citra ini terbagi menjadi dua, yaitu citra berwarna putih yang mewakili *foreground* atau dalam hal ini adalah objek dari citra dan citra berwarna hitam yang mewakili *background* citra. Kegunaan dari *thresholding* adalah untuk mengelompokkan wilayah citra otak dengan *background*-nya. Setelah citra diubah menjadi citra biner, dilakukan operasi erosi dan dilasi untuk menghilangkan daerah diluar ROI sehingga nantinya pada proses *autocropping*, kontur akan terfokus pada daerah ROI dan memotong gambar sesuai dengan daerah ROI yang seharusnya. Apabila filter morfologi tidak diterapkan, daerah diluar ROI seperti pada daerah yang ditandai di Gambar 5.1 akan terdeteksi sebagai batas citra dan proses *autocropping* akan memotong citra dibatas tersebut.



Gambar 5.1 Daerah diluar ROI (ditandai dengan marker merah)

5.1.3. Autocropping

Setelah citra melalui filter morfologi, tahap selanjutnya adalah mencari kontur pada citra tersebut dan mengambil daerah terbesarnya. Kontur digunakan untuk mendeteksi tepi pada citra. Dari citra yang telah melalui tahap *thresholding*, diterapkan *library* OpenCV dengan mode pengambilan kontur eksternal, sehingga diambil bagian tepi terluar dari citra. Tahap ini berfungsi untuk menentukan titik ekstrim pada citra kontur.

Tahap terakhir yaitu *autocropping*, dari nilai kontur yang telah diperoleh akan dicari titik ekstrimnya yang akan digunakan sebagai batas pemotongan citra. Setelah titik-titik ini ditemukan, maka *automatic cropping* akan memotong bagian tersebut seperti pada Gambar 4.5. *Automatic cropping* digunakan untuk memfokuskan citra masukan pada objek yang akan diolah. Fokus penelitian ini adalah citra CT scan kepala atau otak. Citra hasil *automatic cropping* ini yang akan menjadi citra masukan untuk tahap segmentasi otak.

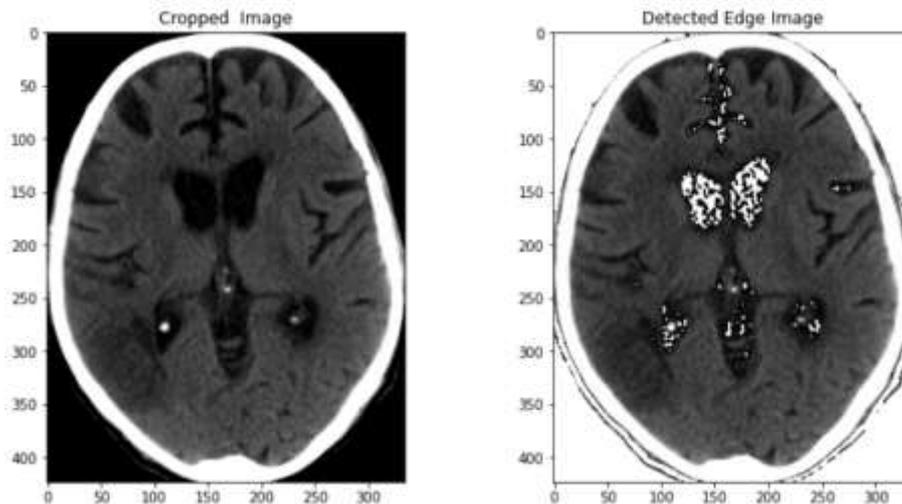
5.1.4. Segmentasi Otak

Segmentasi otak dilakukan untuk menghilangkan daerah tengkorak dari citra CT scan otak. Ini dilakukan dengan cara menentukan batas antara otak dan tengkorak menggunakan *edge detector* berupa filter Laplacian. Dengan dilakukannya tahap ini, akan membuat citra yang diproses lebih terfokus pada daerah otak saja sehingga akan meningkatkan interpretasi citra CT scan otak. Batas yang terdeteksi oleh filter Laplacian dapat dilihat pada Gambar 5.2.

Dari batas yang telah diperoleh tersebut, dilakukan *thresholding* dengan batas 200, dimana piksel dengan intensitas lebih dari 200 akan diubah menjadi intensitas 0 sehingga bagian tengkorak yang berwarna putih akan hilang. Hasil dari segmentasi otak dapat dilihat pada Gambar 4.6.

5.1.5. Peningkatan Citra

Proses peningkatan citra dilakukan dengan menggunakan metode DWT-CLAHE. Metode ini dipilih karena terbukti dapat meningkatkan kualitas citra secara tepat dan tidak berlebihan sehingga tidak merusak informasi yang terdapat didalam citra. Metode ini diawali dengan menentukan jenis *mother wavelet* yang akan digunakan. Terdapat berbagai macam *mother*



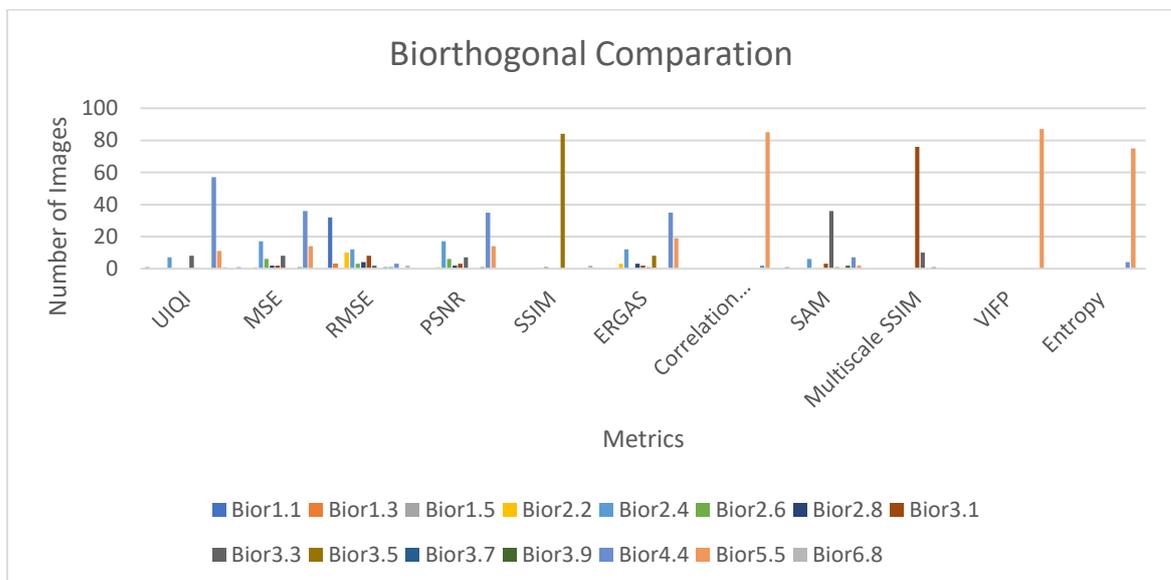
Gambar 5.2 Deteksi tepi dengan Laplacian filter

wavelet yang telah ditemukan, dan performansi *mother wavelet* terbaik tergantung dari jenis citra yang digunakan. Pada penelitian ini dilakukan percobaan terhadap 7 *mother wavelet* dengan skala yang berbeda-beda lalu dibandingkan performansinya menggunakan 11 metrik seperti yang telah dijelaskan pada bab 4.

Percobaan pertama dilakukan untuk menguji skala terbaik pada sebuah *mother wavelet*. Pada *mother wavelet* Biorthogonal terdapat 15 skala yaitu 1.1, 1.3, 1.5, 2.2, 2.4, 2.6, 2.8, 3.1, 3.3, 3.5, 3.7, 3.9, 4.4, 5.5, 6.8. Dua buah angka pada skala ini menunjukkan skala dekomposisi pada angka pertama dan skala rekonstruksi pada angka kedua. *Mother wavelet* yang dihasilkan oleh masing-masing skala dapat dilihat pada Gambar 2.17.

Setelah dilakukan percobaan terhadap skala-skala tersebut diperoleh hasil seperti pada Gambar 4.7 sampai 4.12. Secara visual, tidak terdapat perbedaan antar skala-skala tersebut, namun setelah dilakukan pengujian kuantitatif menggunakan metrik diperoleh grafik perbandingan seperti pada Gambar 5.3.

Pada grafik tersebut sumbu x menunjukkan metrik yang digunakan dan sumbu y menunjukkan jumlah citra dengan nilai metrik tertinggi pada setiap skala. Dari grafik ini dapat



Gambar 5.3 Perbandingan mother wavelet Biorthogonal

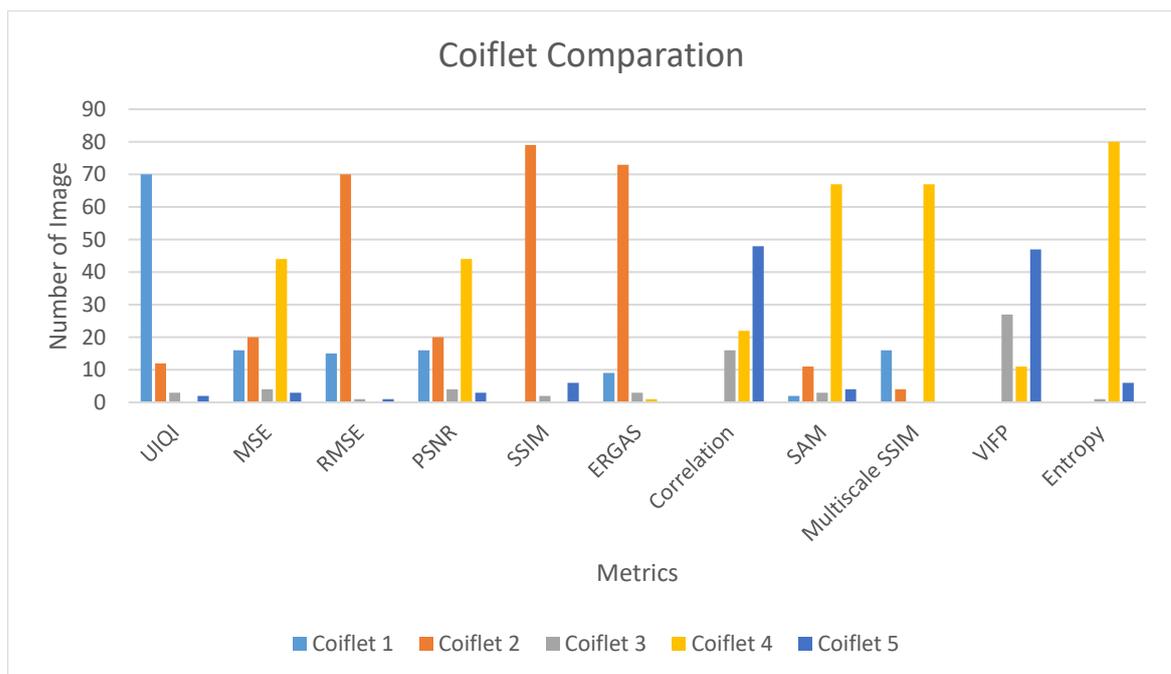
dilihat bahwa *mother wavelet* Biorthogonal dengan skala 5.5 memiliki nilai *correlation coefficient*, VIFP, dan entropi tertinggi pada mayoritas citra. Sedangkan skala 4.4 unggul dalam metrik UIQI, MSE, PSNR dan ERGAS namun hanya pada sebagian kecil citra, sehingga dapat dikatakan bahwa skala 4.4 tidak bersifat universal dan hanya memiliki performa yang baik pada citra-citra tertentu saja.

Selanjutnya dilakukan percobaan untuk menguji skala terbaik pada *mother wavelet* Coiflet. Pada *mother wavelet* Coiflet terdapat 5 skala yaitu Coiflet 1, 2, 3, 4, 5. Angka ini menunjukkan skala dekomposisi sekaligus rekonstruksi. *Mother wavelet* yang dihasilkan oleh masing-masing skala dapat dilihat pada Gambar 2.14.

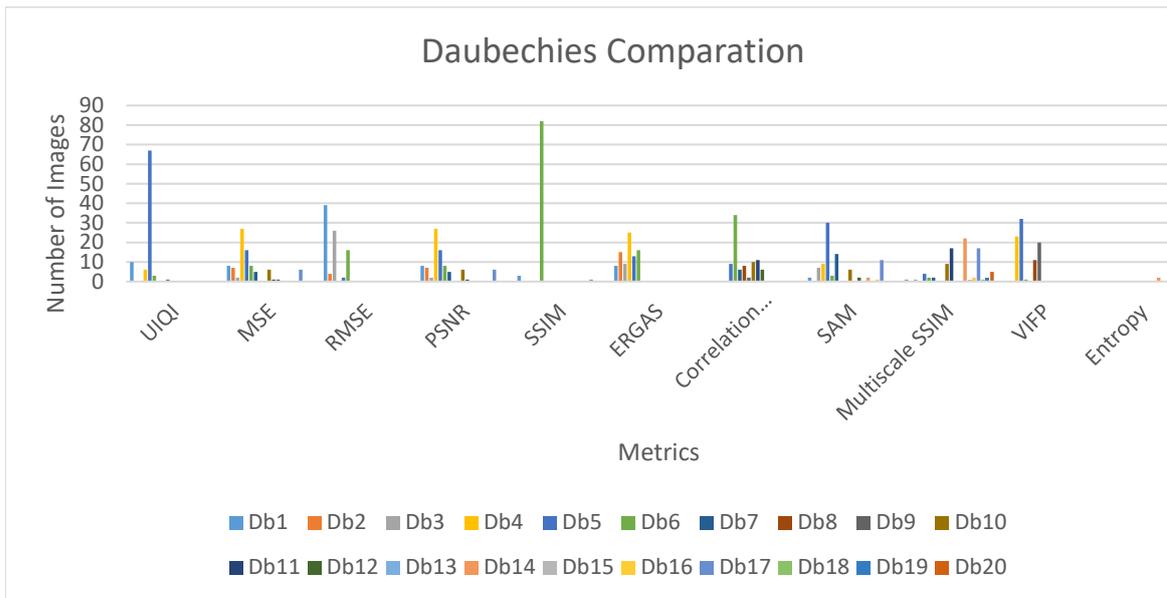
Setelah dilakukan percobaan terhadap skala-skala tersebut diperoleh hasil seperti pada Gambar 4.13 sampai 4.17. Selanjutnya dilakukan pengujian kuantitatif menggunakan metrik dan diperoleh grafik perbandingan seperti pada Gambar 5.4. Dari grafik ini dapat dilihat bahwa *mother wavelet* Coiflet dengan skala 4 memiliki nilai MSE, PSNR, SAM, *Multiscale SSIM*, dan entropi tertinggi pada mayoritas citra.

Percobaan selanjutnya dilakukan pada *mother wavelet* Daubechies pada skala 1 hingga 20. *Mother wavelet* yang dihasilkan oleh masing-masing skala dapat dilihat pada Gambar 2.13. Setelah dilakukan percobaan terhadap skala-skala tersebut diperoleh hasil seperti pada Gambar 4.18 sampai 4.22. Selanjutnya dilakukan pengujian kuantitatif menggunakan metrik dan diperoleh grafik perbandingan seperti pada Gambar 5.5. Dari grafik ini dapat dilihat bahwa rata-rata *mother wavelet* memiliki performansi yang hampir sama. Tidak ada *mother wavelet* yang menonjol dan memiliki nilai terbaik daripada yang lainnya.

Mother wavelet Haar dan *discrete meyer* hanya memiliki 1 jenis wavelet. Hasil peningkatan citra menggunakan kedua wavelet ini dapat dilihat pada Gambar 4.23 dan 4.24.



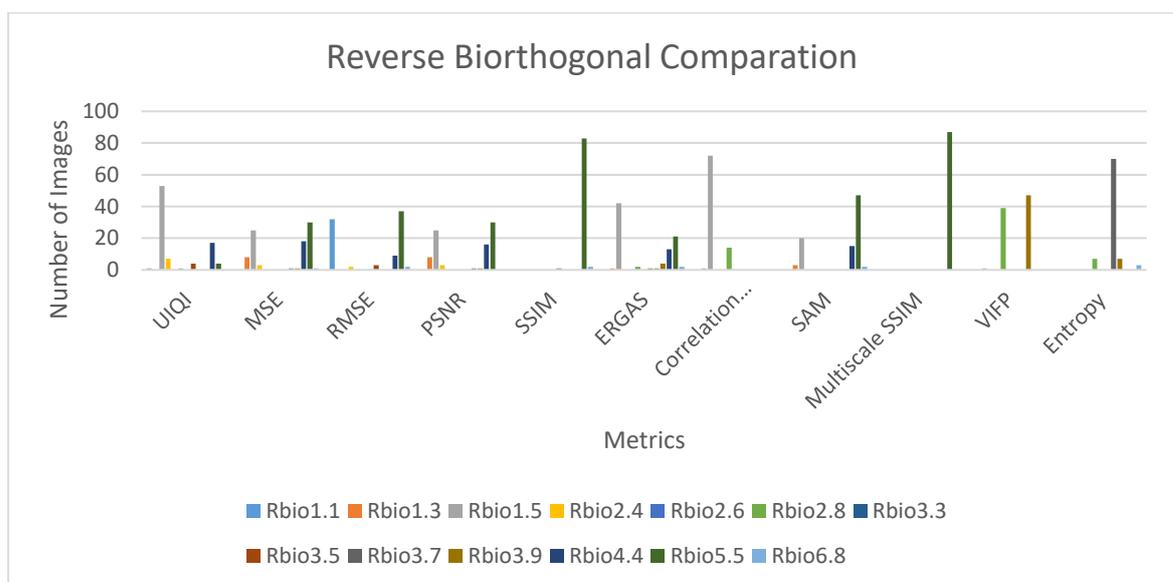
Gambar 5.4 Perbandingan mother wavelet Coiflet



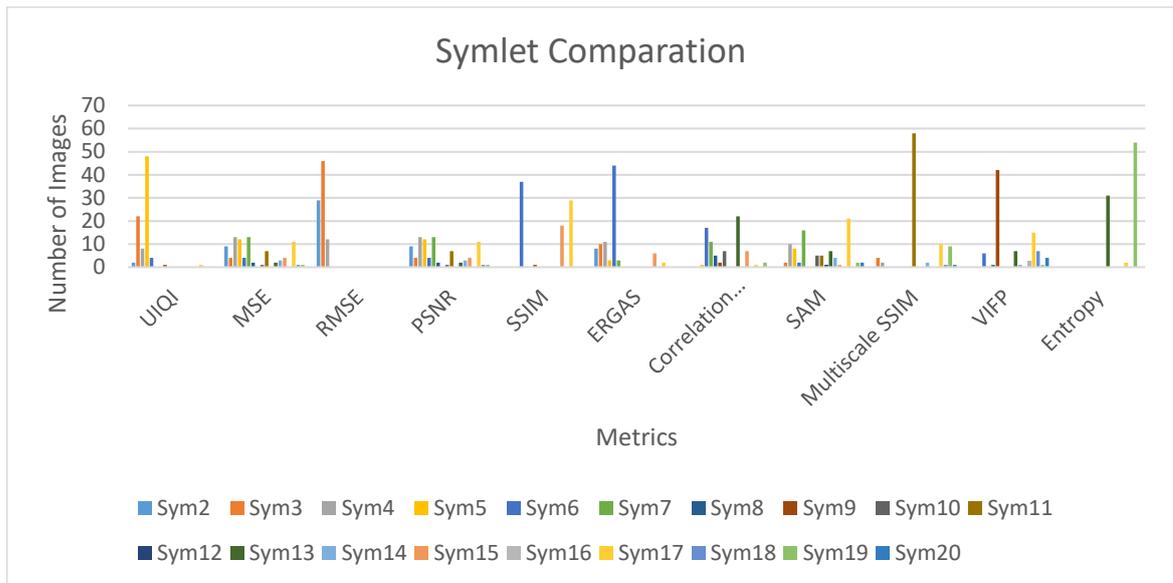
Gambar 5.6 Perbandingan mother wavelet Daubechies

Percobaan selanjutnya dilakukan terhadap *mother wavelet* Reverse Biorthogonal dengan skala yang sama seperti *mother wavelet* Biorthogonal. Perbedaan antara kedua *mother wavelet* ini adalah pada *mother wavelet* Biorthogonal, skala dekomposisi ditentukan terlebih dahulu sebelum skala rekonstruksi, sedangkan pada reverse biorthogonal adalah sebaliknya.

Setelah dilakukan percobaan terhadap skala-skala tersebut diperoleh hasil seperti pada Gambar 4.25 sampai 4.29. Selanjutnya dilakukan pengujian kuantitatif menggunakan metrik dan diperoleh grafik perbandingan seperti pada Gambar 5.6. Dari grafik ini dapat dilihat bahwa *mother wavelet* reverse biorthogonal dengan skala 5.5 memiliki nilai MSE, RMSE, PSNR, SSIM, SAM, dan Multiscale SSIM tertinggi pada mayoritas citra.



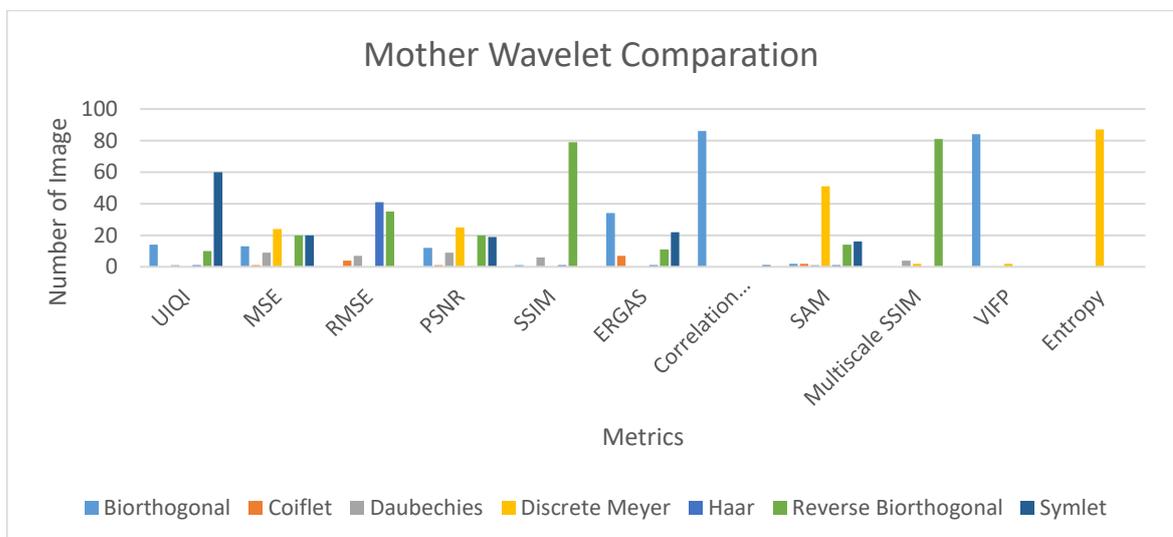
Gambar 5.5 Perbandingan mother wavelet reverse biorthogonal



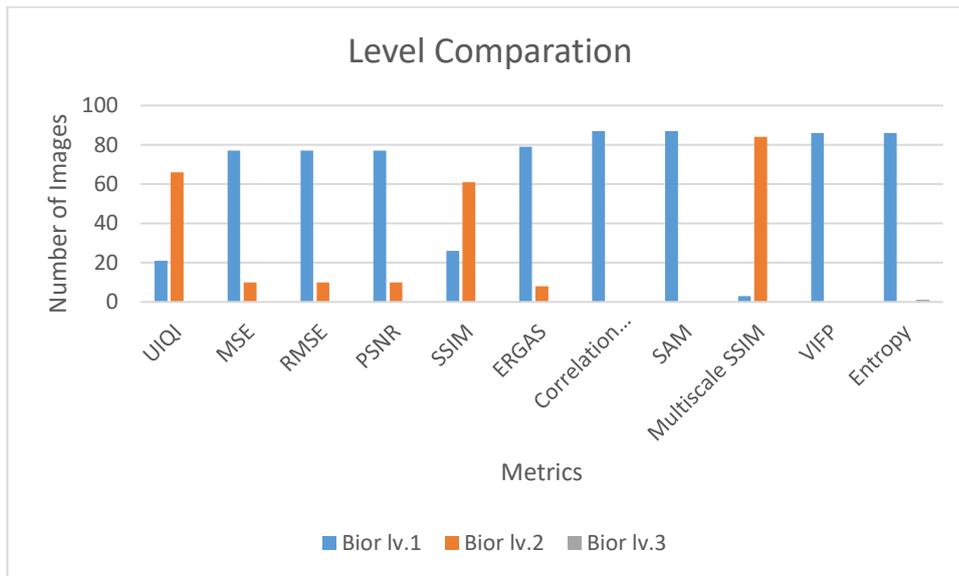
Gambar 5.7 Perbandingan mother wavelet Symlet

Jenis *mother wavelet* terakhir yang diuji adalah *mother wavelet* Symlet dengan skala 1 hingga 20. Wavelet yang dihasilkan oleh setiap skala dapat dilihat pada Gambar 2.15. Setelah dilakukan percobaan terhadap skala-skala tersebut diperoleh hasil seperti pada Gambar 4.30 sampai 4.34. Selanjutnya dilakukan pengujian kuantitatif menggunakan metrik dan diperoleh grafik perbandingan seperti pada Gambar 5.7. Dari grafik ini dapat dilihat bahwa *mother wavelet* Symlet memiliki performansi yang hampir sama pada setiap skala.

Setelah diperoleh skala terbaik pada setiap *mother wavelet*, dilakukan pengujian terhadap semua *mother wavelet* tersebut untuk memperoleh *mother wavelet* dengan performansi terbaik pada citra otak. Berdasarkan pengujian sebelumnya dilakukan perbandingan terhadap *mother wavelet* Biorthogonal 5.5, Coiflet 4, Daubechies 4, Discrete meyer, Haar, reverse biorthogonal 5.5, dan symlet 5. Hasil komparasi antar *mother wavelet* ini dapat dilihat pada Gambar 5.8.



Gambar 5.8 Perbandingan antar *mother wavelet*



Gambar 5.9 Perbandingan antar level DWT

Dari grafik perbandingan tersebut dapat dilihat bahwa *mother wavelet* biorthogonal memiliki performansi terbaik berdasarkan metrik *correlation coefficient* dan VIFP. Oleh sebab itu, pada penelitian ini, *mother wavelet* yang akhirnya diaplikasikan adalah *mother wavelet* Biorthogonal 5.5. Setelah memperoleh *mother wavelet* terbaik, dilakukan pengujian untuk menentukan level dekomposisi DWT terbaik dengan membandingkan antara DWT level 1, 2, dan 3. Hasil perbandingan antara ketiga level ini dapat dilihat pada Gambar 5.9.

Dari hasil perbandingan tersebut dapat dilihat bahwa DWT level 1 memiliki performansi terbaik pada 8 dari 11 metrik yang digunakan. Sehingga pada penelitian ini dilakukan DWT level 1 menggunakan *mother wavelet* Biorthogonal 5.5.

5.2. Pelatihan CNN

Metode CNN dapat digunakan untuk melakukan segmentasi semantik pada citra dengan melakukan klasifikasi dalam skala piksel. U-Net merupakan salah satu arsitektur dari CNN untuk melakukan segmentasi pada kasus gambar biomedis. Untuk melakukan segmentasi, dibutuhkan masukan data pelatihan berupa gambar dan label untuk menghasilkan keluaran gambar hasil prediksi segmentasi. Label yang digunakan berupa data *groundtruth* dari kasus yang digunakan untuk pelatihan. Proses pelatihan CNN dengan arsitektur U-Net akan mengambil informasi berupa fitur yang ada pada gambar serta lokasi dari objek. Fitur pada data akan direkonstruksi untuk menghasilkan keluaran gambar dengan ukuran yang sama. Tabel 4.1 menunjukkan arsitektur yang digunakan untuk melakukan pelatihan dengan tujuan untuk mendapatkan *trained model*. *Trained model* nantinya akan dapat melakukan segmentasi secara otomatis pada data uji.

Berbeda dari arsitektur U-Net yang diajukan oleh Olaf Ronneberger pada 2015, pada penelitian ini digunakan hasil pengembangan dari arsitektur U-Net dimana pada proses ekstraksi fitur digunakan *backbone* Resnet34. Resnet34 merupakan arsitektur CNN yang dikembangkan untuk mengatasi permasalahan *vanishing moment* yang sering muncul pada CNN dengan jumlah layer yang banyak, yaitu fenomena dimana tidak terjadi pengoptimalan *weight* pada model sehingga menyebabkan tidak terjadinya proses *learning*. Hal ini dapat

diatasi dengan menambahkan *skip connections* pada setiap *stage* arsitektur seperti yang ditunjukkan oleh Gambar 3.3.

Arsitektur dari UNET terdiri dari dua bagian yaitu *contracting path* dan *expansive path* dengan detail seperti pada Tabel 4.1. Pada masing-masing bagian terdapat 4 *stage* dan setiap *stage* terdiri dari *batch normalization*, *zero padding*, konvolusi, *activation*, dan *adder*.

Batch normalization adalah metode yang digunakan untuk membuat CNN lebih cepat dan stabil dengan melakukan normalisasi terhadap *layer* melalui proses *recentering* dan *rescaling* pada setiap *batch*. *Zero padding* adalah proses yang dilakukan dengan menambahkan 0 disekitar data input untuk mempertahankan jumlah data pada proses konvolusi.

Konvolusi digunakan dengan cara melakukan perkalian dengan suatu kernel untuk menghasilkan *feature map* atau peta dari fitur gambar. *Max pooling* digunakan untuk melakukan *subsampling* untuk mereduksi ukuran data gambar sehingga dapat memfokuskan *layer* konvolusi untuk mengekstraksi fitur. *Max pooling* akan mengambil nilai data paling besar dari kernel yang dijalankan dengan *stride* tertentu. *Stride* merupakan nilai untuk menjalankan kernel sehingga dapat memenuhi seluruh ukuran data gambar.

Adder digunakan untuk menjumlahkan tensor *output* dari konvolusi dengan tensor *skip connections* yang kemudian akan menjadi input pada layer selanjutnya.

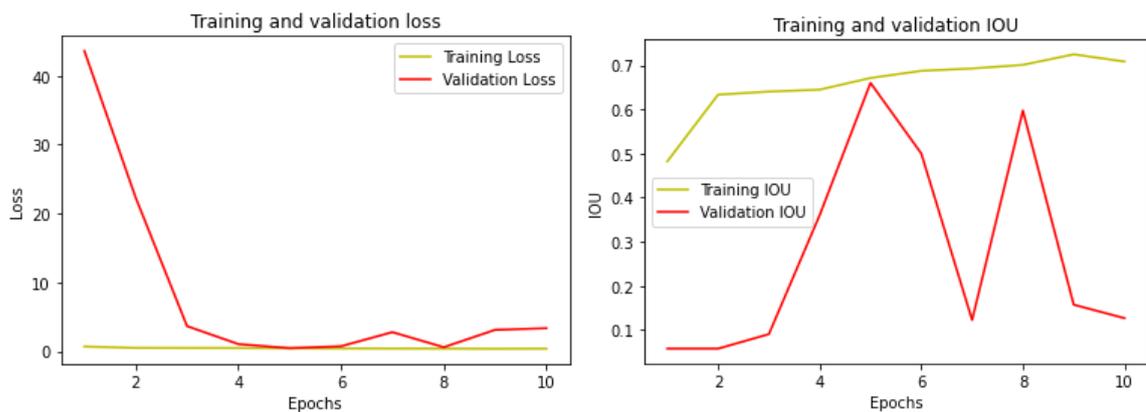
Selanjutnya adalah bagian *expansive part*. Pada bagian ini dimanfaatkan *layer upsampling* untuk mengembalikan dimensi gambar yang diakibatkan oleh proses *max pooling* untuk pencarian fitur. Jumlah perulangan proses pada *expansive path* sama dengan pada bagian *contractive path*. Dengan penambahan *layer concatenate* yang berfungsi untuk menggabungkan rangkaian dengan memanfaatkan informasi pada *layer* cerminan sehingga dihasilkan lokasi yang lebih presisi. Karena arsitektur U-Net memiliki bentuk U sebagai pencerminan proses satu sama lain untuk merekonstruksi dimensi dari gambar sehingga didapatkan informasi lokasi dari ROI.

Pelatihan dilakukan dengan melakukan *backpropagation* untuk memperbarui nilai *weight* untuk memperoleh *weight* maksimum yang dapat melakukan segmentasi terhadap citra. Nilai *weight* diperbarui dengan memperhatikan *loss* dan IoU. Nilai *loss* dan IoU pada setiap *epoch* dapat di plot kedalam sebuah kurva yang disebut *learning curve*. *Learning curve* merupakan plot performansi model pembelajaran terhadap waktu. *Learning curve* dilakukan pengamatan pada *loss* dan *intersection of union* yang berfungsi untuk mengevaluasi *dataset* pelatihan dengan cara diberikan data validasi untuk pengujian pada setiap iterasi atau *epoch*. Nilai *loss* terbaik pada sebuah *learning* adalah mendekati 0 dan IoU terbaik adalah 1. Dengan mengamati pola grafik yang dihasilkan dapat digunakan untuk mendiagnosa masalah yang terjadi saat proses pembelajaran. Apakah model yang digunakan *overfitting*, *underfitting* atau telah dapat menunjukkan performansi yang cukup baik serta untuk mengetahui apakah data pelatihan dan data validasi yang digunakan telah representatif. *Learning curve* untuk setiap *epoch* dapat dilihat pada Gambar 5.10 hingga 5.12.

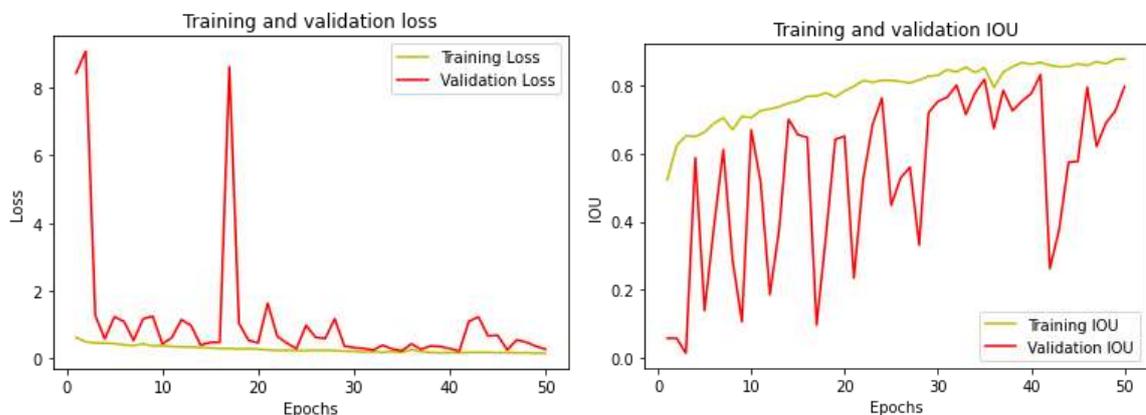
Dari *learning curve* masing-masing percobaan pada *epoch* 10, 50, dan 100 terlihat bahwa *epoch* 10 tidak cukup baik dalam pelatihan dan validasi data. Ini dapat dilihat dari kedua kurva *loss* dan IoU yang masih naik turun dan tidak memiliki kecenderungan yang naik. Ini juga dapat dilihat dari nilai akhir *loss* dan IoU yang dicapai pada pelatihan dengan 10 *epoch* yaitu *loss* 3.13 dan IoU 0.72. Ketika *weight* dari pelatihan ini digunakan untuk melakukan segmentasi terhadap data uji, diperoleh hasil seperti pada Tabel 4.2. Dari tabel dapat dilihat bahwa hasil segmentasi

tidak cukup akurat dimana masih ada daerah yang bukan *hypodense* namun terdeteksi sebagai *hypodense* seperti pada gambar di kolom kedua dan ketiga pada tabel. Selain itu juga terdapat daerah *hypodense* yang tidak tersegmentasi seperti pada gambar di kolom keempat pada tabel. Kesalahan segmentasi ini disebabkan karena *epoch* yang terlalu sedikit yang menyebabkan proses *learning* belum maksimal atau disebut juga dengan *underfitting*. Setelah dilakukan perhitungan nilai DSC dari hasil segmentasi dengan *ground truth* diperoleh nilai rata-rata DSC pada 250 data uji sebanyak 0.803.

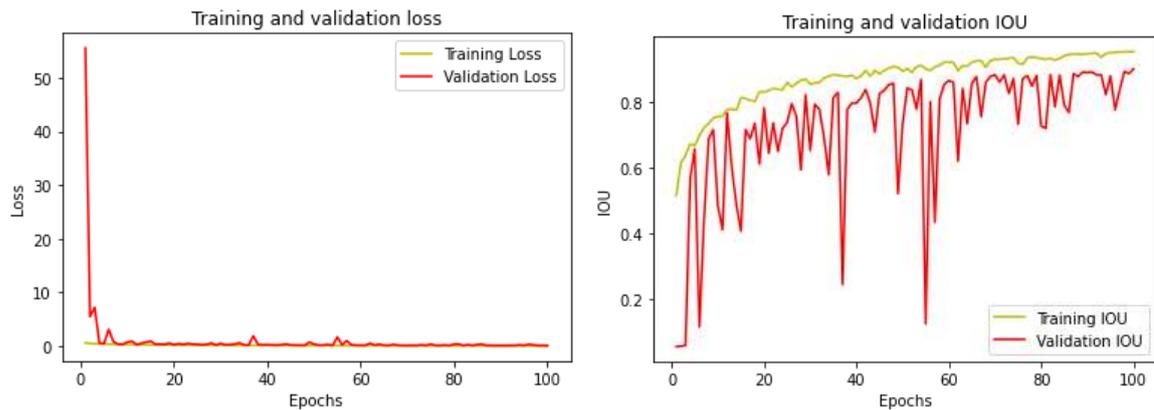
Untuk mengatasi permasalahan *underfitting* tersebut dilakukan peningkatan jumlah *epoch* menjadi 50. Dari *learning curve* dapat dilihat bahwa proses *learning* sudah semakin membaik yang dapat dilihat dari kurva loss yang cenderung turun dan kurva IoU yang semakin mendekati 1. Namun kurva ini masih sangat fluktuatif baik pada *loss* maupun IoU. Ketika *weight* dari pelatihan ini digunakan untuk melakukan segmentasi terhadap data uji, diperoleh hasil seperti pada Tabel 4.3. Dari tabel dapat dilihat bahwa pada beberapa citra hasil segmentasi sudah cukup baik daripada *epoch* 10 dan ketika dilakukan perhitungan nilai rata-rata DSC dari 250 data uji diperoleh nilai DSC sebanyak 0.825. Namun pada beberapa citra lain seperti pada gambar di kolom keempat pada tabel hasil segmentasi masih jauh dari yang seharusnya. Ini menunjukkan bahwa *weight* masih belum stabil dan masih perlu proses *learning* lebih lanjut lagi.



Gambar 5.11 Learning curve CNN dengan epoch 10



Gambar 5.10 Learning curve CNN dengan epoch 50



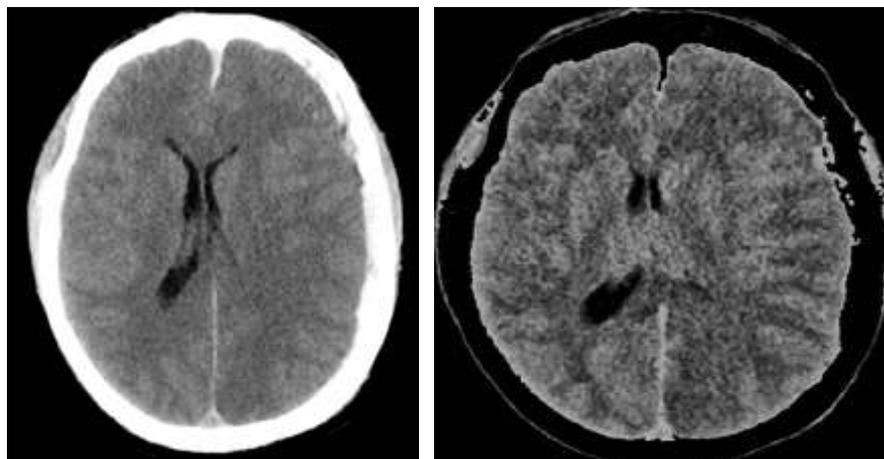
Gambar 5.12 Learning curve CNN dengan epoch 100

Oleh sebab itu *epoch* ditingkatkan lagi menjadi 100. Dengan menggunakan *epoch* 100 terlihat bahwa kurva *loss* dan IoU semakin membaik dengan nilai *loss* cenderung turun dan mendekati 0 sedangkan kurva IoU cenderung naik dan mendekati 1. Pada nilai akhir kurva ini diperoleh nilai *loss* 0.057 dan IoU 0.9. Ketika *weight* dari pelatihan ini diterapkan pada data uji, diperoleh hasil seperti pada Tabel 4.4. Dari tabel dapat dilihat bahwa hasil segmentasi sudah sangat baik dan sesuai dengan *ground truth*. Ini juga dapat dilihat dari nilai DSC yang mencapai 0.932 yang menunjukkan bahwa hasil segmentasi menggunakan *weight* dari pelatihan tersebut telah 93,2% sama dengan *ground truth*.

5.3. Hasil Performansi

5.3.1. Hasil Peningkatan Citra

Peningkatan citra yang dilakukan dengan metode DWT-CLAHE menggunakan *mother wavelet* Biorthogonal 5.5 menunjukkan hasil yang cukup baik. Ini ditunjukkan oleh hasil lesi *hypodense* yang jauh lebih jelas daripada sebelum dilakukan peningkatan, khususnya pada citra kasus *posterior cerebral artery territory infarct*. Pada kasus ini, kontras antara jaringan otak normal dan lesi *hypodense* sangat mirip sehingga apabila citra tanpa peningkatan ini diinputkan pada CNN, akan sulit untuk menemukan fiturnya pada proses ekstraksi fitur. Hasil peningkatan citra pada kasus ini dapat dilihat pada Gambar 5.13.



Gambar 5.13 Hasil peningkatan citra pada kasus *posterior cerebral artery territory infarct*

Dari Gambar 5.13 terlihat bahwa peningkatan citra menggunakan metode yang diajukan memiliki performansi yang baik dan dapat membuat lesi *hypodense* terlihat semakin jelas.

Setelah dilakukan pengujian menggunakan metrik terhadap hasil dari peningkatan citra dengan metode yang diajukan, diperoleh rata-rata nilai UIQI 0.684452004, PSNR 24.9892, dan multiscale SSIM 0.89916. Nilai UIQI yang mendekati 1 menunjukkan kualitas citra keseluruhan yang baik, sedangkan nilai PSNR terbaik dari sebuah adalah pada 30-50 dB. Nilai PSNR yang mendekati 30 menunjukkan bahwa rasio antara sinyal dan *noise* sudah cukup rendah. *Multiscale* SSIM menunjukkan kemiripan antara citra original dan citra yang telah ditingkatkan. Nilai SSIM yang mendekati 0.89916 menunjukkan bahwa kedua citra 89% mirip sehingga dapat disimpulkan bahwa secara umum metode peningkatan citra yang diajukan telah mampu meningkatkan citra dengan baik namun tidak berlebihan sehingga citra masih mirip dengan citra original dan tidak ada informasi yang hilang.

Untuk menguji pengaruh peningkatan citra terhadap proses pelatihan CNN, dilakukan percobaan dengan melatih CNN menggunakan data yang belum dilakukan peningkatan. Hasil yang diperoleh dari percobaan ini dapat dilihat pada Tabel 4.5 hingga 4.7. Percobaan dilakukan pada CNN dengan *epoch* 10, 50, dan 100. Dapat dilihat bahwa hasil yang diperoleh pada CNN dengan *input* berupa citra yang ditingkatkan menghasilkan segmentasi yang lebih akurat daripada CNN dengan *input* berupa citra yang tidak ditingkatkan. Ini dapat dilihat dari nilai rata-rata DSC dari hasil segmentasi CNN dengan *input* citra tanpa peningkatan. Rata-rata DSC yang diperoleh dari segmentasi CNN dengan *input* citra tanpa peningkatan dengan *epoch* 10, 50, dan 100 secara berturut-turut adalah 0.682, 0.839, dan 0.896. Pada *epoch* 10 dan 100 nilai DSC dari CNN dengan *input* citra tanpa peningkatan lebih rendah daripada CNN dengan *input* berupa citra yang ditingkatkan. Hal ini disebabkan karena pada citra yang tidak ditingkatkan intensitas piksel ROI dan *background* sangat mirip sehingga batasnya menjadi tidak jelas dan menyebabkan proses ekstraksi fitur menjadi lebih sulit. Sedangkan pada *epoch* 50 CNN dengan *input* berupa citra tanpa peningkatan memiliki nilai yang lebih tinggi. Hal ini dapat disebabkan karena ketidakstabilan model seperti yang sebelumnya dibahas pada subbab 5.2.

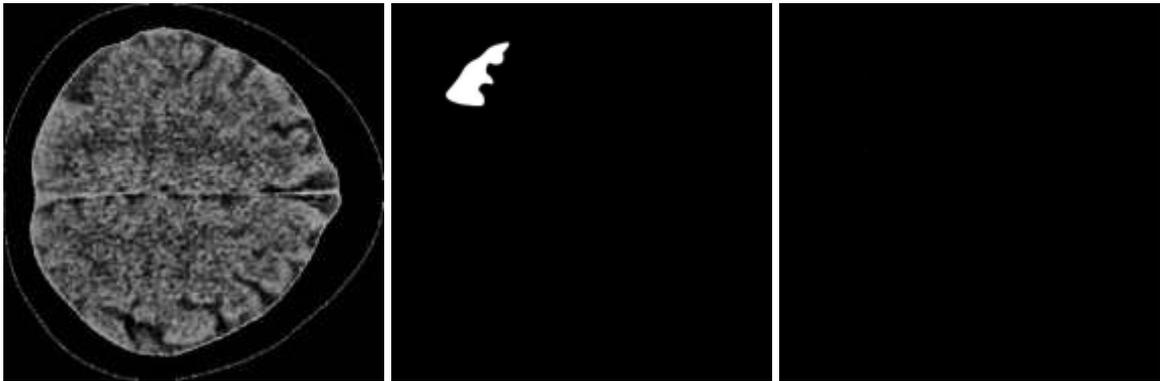
5.3.2. Hasil Segmentasi Citra

Proses pelatihan untuk mendapatkan *weight* menggunakan 750 data citra. *Learning Curve* yang ditunjukkan pada Gambar 5.12 untuk performansi *loss* dan *IoU* pada *epoch* 100 menunjukkan bahwa data validasi mencapai nilai dengan rentang perbedaan yang cukup besar dengan data pelatihan. Hal ini disebabkan karena data validasi yang digunakan pada *epoch* masih kurang representatif untuk diaplikasikan pada model atau dapat juga dikarenakan data validasi yang digunakan cukup besar sehingga letak ROI pada gambar pun variatif, dikarenakan pembagian dilakukan secara otomatis tanpa dilakukan pengecekan pada lebih lanjut pada gambar. Pada *learning curve* yang dihasilkan ini digunakan data validasi sebesar 250 untuk menguji model pada setiap iterasi atau *epoch*.

Proses evaluasi dengan memanfaatkan *loss* dan *IoU* menunjukkan nilai 0,05 dan 0,9. Hal ini menunjukkan bahwa performansi dari model telah cukup baik. Namun, pada saat diamati lebih lanjut gambar prediksi yang dihasilkan terdapat gambar yang tidak membawa informasi dari objek ROI. Seperti kasus pada Gambar 5.14. Hal ini terjadi diakibatkan karena evaluasi dari model CNN dilakukan dengan cara melakukan perbandingan dari nilai data pada setiap *pixel*. Sehingga jika diamati, bagian dari *background* dengan warna hitam menempati bagian

yang cukup besar dibandingkan dengan ROI. Sedangkan hasil segmentasi menunjukkan gambar dengan daerah penuh hitam. Hal ini tidak akan menyebabkan penurunan dari performansi karena ROI hanya mendapatkan presentasi yang kecil pada piksel gambar. Sehingga pada kasus *imbalance data* atau ketidakseimbangan antara kedua kelas data yang diklasifikasi (ROI dan *background*) tidak dapat digunakan untuk menyimpulkan performansi dari gambar hasil segmentasi.

Pengujian dari *trained model* yang telah dihasilkan menggunakan data sebanyak 250 buah data dengan memanfaatkan *Dice Similarity Coefficient* untuk menguji hasil performansi pada setiap gambar prediksi yang dihasilkan. Nilai DSC dari pengujian adalah 0.932 yang menunjukkan bahwa hasil segmentasi menggunakan weight dari pelatihan tersebut telah 93,2% sama dengan ground truth. Hasil segmentasi secara lengkap dapat dilihat pada lampiran.



Gambar 5.14 Hasil segmentasi yang tidak mengandung ROI

BAB 6

KESIMPULAN DAN SARAN

6.1. Kesimpulan

Pada penelitian ini diajukan sebuah sistem yang dapat melakukan segmentasi terhadap lesi *hypodense* untuk deteksi awal stroke iskemia. Tahapan yang dilakukan pada sistem ini ialah *grayscale*, *filtering*, *autocropping*, segmentasi otak, peningkatan citra, pelatihan CNN untuk segmentasi, dan pengujian. *Grayscale* berfungsi untuk mengkonversi citra masukan menjadi citra *grayscale*. Selanjutnya dilakukan *filtering* untuk mengurangi *noise* yang terdapat pada citra. Kemudian, *autocropping* digunakan untuk memfokuskan citra masukan pada objek yang akan diolah. Tahap berikutnya dalam sistem ini adalah peningkatan citra yang terdiri merupakan kombinasi antara metode *Discrete Wavelet Transform* (DWT) dengan *Contrast Limited Adaptive Histogram Equalization* (CLAHE) untuk *denoising*, memperjelas detail dan meningkatkan kontras pada citra. Hasil peningkatan citra ini kemudian diaugmentasi dan menjadi input pada model CNN U-Net untuk proses segmentasi citra.

Berdasarkan hasil percobaan yang dilakukan dapat diambil beberapa kesimpulan, yaitu metode CNN yang diajukan telah menunjukkan performansi yang sangat baik dalam melakukan segmentasi otomatis terhadap citra CT scan stroke iskemia yang ditunjukkan oleh nilai DSC yang mencapai 0.932. Performansi ini dipengaruhi oleh *hyperparameter* yang mengontrol proses *learning* dan data yang digunakan sebagai input pada proses *learning*.

Percobaan dilakukan terhadap *hyperparameter epoch*, yaitu digunakan *epoch* 10, 50, dan 100 untuk menentukan jumlah *epoch* yang tepat untuk proses *learning model*. Dari percobaan tersebut diperoleh nilai DSC untuk *epoch* 10, 50, dan 100 secara berturut-turut sebanyak 0,803, 0,825 dan 0,932 sehingga dapat disimpulkan bahwa peningkatan jumlah *epoch* berbanding lurus dengan peningkatan nilai DSC yang dibuktikan dari DSC *epoch* 100 yang memiliki nilai tertinggi dari *epoch* lainnya.

Untuk mengoptimalkan proses *learning*, pada data input dilakukan *preprocessing* yang membuat citra menjadi lebih jelas dan kontras antara lesi *hypodense* dan jaringan lain disekitarnya menjadi meningkat. Penentuan parameter jenis *mother wavelet* dan level dekomposisi pada *Discrete Wavelet Transform* (DWT) sangat berpengaruh pada detail serta kekontrasan citra. Ini dapat dilihat dari nilai DSC yang diperoleh dari segmentasi CNN dengan input citra tanpa peningkatan dengan *epoch* 10, 50, dan 100 secara berturut-turut adalah 0.682, 0.839, dan 0.896 yang nilainya lebih kecil daripada CNN dengan input citra dengan peningkatan.

Berdasarkan perhitungan kuantitatif terhadap metrik diperoleh citra dengan penggunaan *mother wavelet* jenis Biorthogonal 5.5 serta dekomposisi level 1 pada DWT menghasilkan citra yang paling baik kontrasnya secara visual karena tidak mengalami kekurangan atau peningkatan kontras yang berlebihan. Sehingga dapat disimpulkan, metode DWT-CLAHE mampu menghasilkan kontras yang tidak berlebihan namun memiliki kemampuan diferensiasi yang lebih baik antara jaringan lunak normal pada otak dan infark *hypodense* pada otak, serta terbukti dapat meningkatkan kualitas citra CT scan asli.

Pengujian terhadap hasil akhir sistem diperoleh DSC dengan nilai 0.932 yang menunjukkan bahwa tingkat kemiripan (*similarity*) dari citra hasil prediksi dan ground truth

93.2%.

6.2. Saran

Untuk meningkatkan performansi CNN dapat dilakukan penambahan jumlah dataset. Selain itu juga dapat dilakukan pengembangan dengan menggunakan data stroke iskemia dengan onset 1-2 jam, karena waktu ini merupakan waktu yang krusial dalam proses diagnosis stroke iskemia. Pengembangan juga dapat dilakukan untuk mengklasifikasi stroke iskemia dan stroke hemoragik dan pengembangan menjadi aplikasi yang lebih interaktif.

6.3. Ucapan Terima Kasih

Ucapan terima kasih penulis sampaikan kepada pihak *radiopaedia.org* yang telah menyediakan data berupa citra CT scan otak pasien stroke iskemia sehingga penelitian pada Tugas Akhir ini dapat terlaksana.

DAFTAR PUSTAKA

- [1] WHO, “The top 10 causes of death,” 2020. <https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death> (accessed Dec. 22, 2020).
- [2] Kementerian Kesehatan RI, “Laporan Nasional Riset Kesehatan Dasar 2018,” 2018. [Online]. Available: http://labdata.litbang.kemkes.go.id/images/download/laporan/RKD/2018/Laporan_Nasional_RKD2018_FINAL.pdf.
- [3] Centers for Disease Control and Prevention, “Stroke Facts,” 2020. <https://www.cdc.gov/stroke/facts.htm>.
- [4] A. E. Urdaneta and P. Bhalla, “Cutting Edge Acute Ischemic Stroke Management,” *Emerg. Med. Clin. North Am.*, vol. 37, no. 3, pp. 365–379, 2019, doi: 10.1016/j.emc.2019.03.001.
- [5] J. M. Wardlaw, “Radiology of stroke,” *J. Neurol. Neurosurg. Psychiatry*, vol. 70, no. SUPPL. 1, pp. 17–111, 2001, doi: 10.1136/jnnp.70.suppl_1.i7.
- [6] A. Diya *et al.*, “Peningkatan Kualitas Citra CT Scan Otak Menggunakan Metode DWT-CLAHE Untuk Deteksi Awal Stroke Iskemia,” pp. 1–12, 2020.
- [7] T. L. Tan, K. S. Sim, and A. K. Chong, “Contrast enhancement of CT brain images for detection of ischemic stroke,” *2012 Int. Conf. Biomed. Eng. ICoBE 2012*, no. February, pp. 385–388, 2012, doi: 10.1109/ICoBE.2012.6179043.
- [8] T. L. Tan, K. S. Sim, C. P. Tso, and A. K. Chong, “Contrast enhancement of computed tomography images by adaptive histogram equalization-application for improved ischemic stroke detection,” *Int. J. Imaging Syst. Technol.*, vol. 22, no. 3, pp. 153–160, 2012, doi: 10.1002/ima.22016.
- [9] P. D. Boraste and P. K. P. N., “Image Enhancement using DWT,” *Int. J. Eng. Comput. Sci. ISSN2319-7242*, vol. 4, no. 2, pp. 10509–10515, 2015.
- [10] H. Lidong, Z. Wei, W. Jun, and S. Zebin, “Combination of contrast limited adaptive histogram equalisation and discrete wavelet transform for image enhancement,” *IET Image Process.*, vol. 9, no. 10, pp. 908–915, 2015, doi: 10.1049/iet-ipr.2015.0150.
- [11] F. H. Martini, J. L. Nath, and E. F. Bartholomew, *Fundamentals of Anatomy & Physiology. 9th edition*. 2013.
- [12] L. R. Caplan, “Caplan’s Stroke: A Clinical Approach,” in *Caplan’s Stroke: A Clinical Approach*, Fifth., Cambridge: Cambridge University Press, 2016.
- [13] N. Hema Rajini and R. Bhavani, “Computer aided detection of ischemic stroke using segmentation and texture features,” *Meas. J. Int. Meas. Confed.*, vol. 46, no. 6, pp. 1865–1874, 2013, doi: 10.1016/j.measurement.2013.01.010.
- [14] Y. Zhu and C. Huang, “An Improved Median Filtering Algorithm for Image Noise Reduction,” *Phys. Procedia*, vol. 25, pp. 609–616, 2012, doi: 10.1016/j.phpro.2012.03.133.
- [15] O. Zahraa, “Edge Detection and Image Sharpening.”
- [16] A. Dhamija, “A Brief Study of Various wavelet families and compression techniques,” *J. Glob. Res. Comput. Sci.*, vol. 4, no. 4, pp. 43–49, 2013.
- [17] J. W. Woods, *Image Enhancement and Analysis*. 2012.
- [18] “Chapter 3: Wavelets for Biomedical Image Denoising,” Chennai: Anna University, pp. 52–92.
- [19] L. Gabralla, “Denoising CT Images using wavelet transform,” vol. 6, no. 5, pp. 25–29, 2015.
- [20] A. Dixit and S. Majumdar, “Comparative Analysis of Coiflet and Daubechies Wavelet

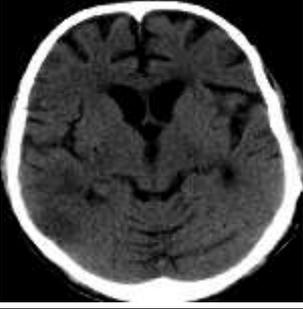
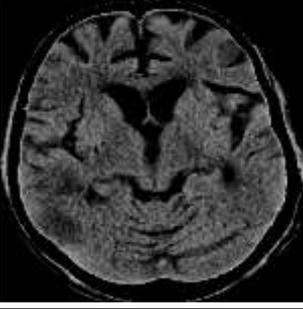
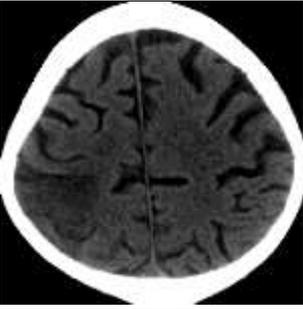
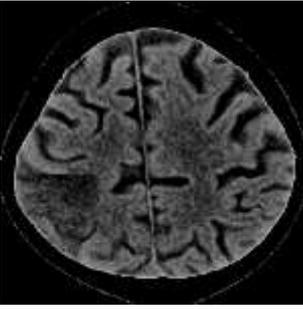
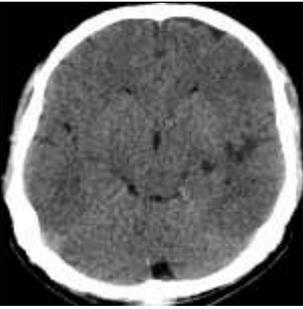
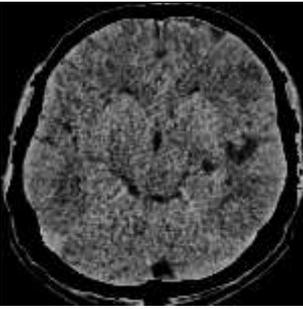
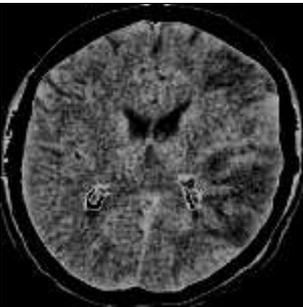
- using Global TRhreshold for Image De-Noising,” *Int. J. Adv. Eng. Technol.*, vol. 6, no. 5, pp. 2247–2252, 2013.
- [21] K. O’Shea and R. Nash, “An Introduction to Convolutional Neural Networks,” pp. 1–11, 2015, [Online]. Available: <http://arxiv.org/abs/1511.08458>.
- [22] “Convolutional Neural Network for Visual Recognition.” <https://cs231n.github.io/convolutional-networks/> (accessed Jun. 05, 2021).
- [23] A. S. A. Amidi, “Convolutional Neural Networks cheatsheet.” <https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-convolutional-neural-networks> (accessed Jun. 06, 2021).
- [24] O. Ronneberger, P. Fischer, and T. B. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation,” 2015.
- [25] D. F. Deng and F. Gaillard, “Ischaemic stroke.” <https://radiopaedia.org/articles/ischaemic-stroke>.
- [26] Z. Wang and A. C. Bovik, “A universal image quality index,” *IEEE Signal Process. Lett.*, vol. 9, no. 3, pp. 81–84, 2002, doi: 10.1109/97.995823.
- [27] Y. A. Y. Al-najjar and D. C. Soong, “Comparison of Image Quality Assessment :,” *Int. J. Sci. Eng. Res.*, vol. 3, no. 8, pp. 1–5, 2012, [Online]. Available: <http://www.ijser.org/researchpaper%5CComparison-of-Image-Quality-Assessment-PSNR-HVS-SSIM-UIQI.pdf>.
- [28] N. Chervyakov, P. Lyakhov, and N. Nagornov, “Analysis of the quantization noise in discrete wavelet transform filters for 3D medical imaging,” *Appl. Sci.*, vol. 10, no. 4, 2020, doi: 10.3390/app10041223.

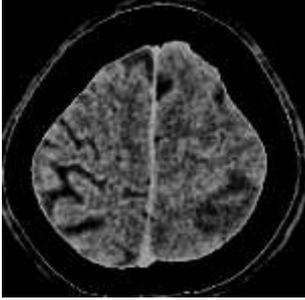
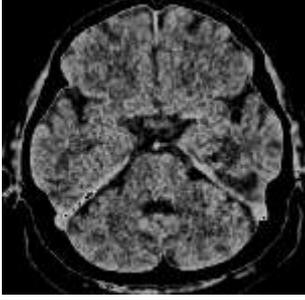
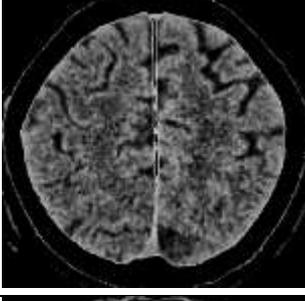
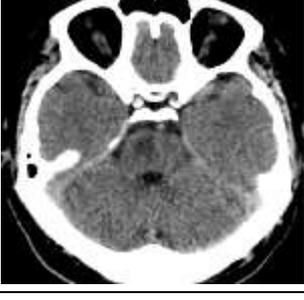
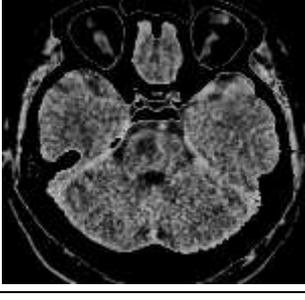
LAMPIRAN

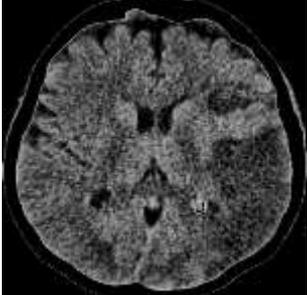
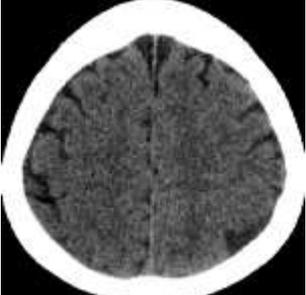
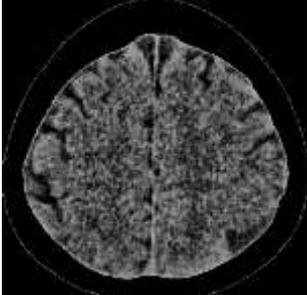
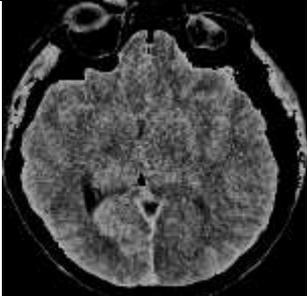
A. Daftar sumber data

Case	Link	Case information	Case description
Acute infarction	https://radiopaedia.org/cases/acute-infarction-1	Case contributed by Dr Hani Makky Al Salam rID: 13087 Published: 27th Feb 2011	Loss of grey white matter differentiation with diffuse hypoattenuation involving right post central gyrus, right superior parietal lobule and down into right superior temporal gyrus.
Ischemic stroke	https://radiopaedia.org/cases/ischemic-stroke-2	Case contributed by Dr David Cuete rID: 26882 Published: 12th Jan 2014	Computed tomography shows two regions of ischemic stroke in the territory of the left middle cerebral artery, involving the regions supplied by both the anterior and posterior branches.
PCA territory infarct	https://radiopaedia.org/cases/pca-territory-infarct-1	Case contributed by Dr David Cuete rID: 30225 Published: 27th Jul 2014	Computed tomography revealed a hypodensity in left occipital with effacement of the sulci. No midline shift is observed.
Pontine infarct	https://radiopaedia.org/cases/pontine-infarct-2	Case contributed by Dr David Cuete rID: 30226 Published: 27th Jul 2014	Computed tomography shows a subtle hypodensity in the right pontine region.
Subacute left middle cerebral artery infarct	https://radiopaedia.org/cases/subacute-left-middle-cerebral-artery-infarct	Case contributed by Dr David Cuete rID: 35732 Published: 20th Apr 2015	Extensive infarction of subacute aspect, in the territory of the left middle cerebral artery without hemorrhagic transformation and no significant mass effect.
Posterior cerebral artery territory infarct	https://radiopaedia.org/cases/posterior-cerebral-artery-territory-infarct-due-to-subdural-haematoma-and-uncal-herniation	Case contributed by Assoc Prof Craig Hacking rID: 40106 Published: 8th Oct 2015	There is a 6mm left convexity acute subdural hematoma with extension along the tentorium cerebelli and falx. Evidence of raised intracranial pressure, with diffuse loss of hemispheric sulci bilaterally, compression of the left lateral and third ventricles, 5mm rightward midline shift and obliteration of the basal cisterns with transtentorial herniation and compression of the midbrain. There is loss of grey-white matter differentiation within the left occipital and temporal lobes, consistent with acute infarction of the left posterior cerebral artery (PCA).

B. Hasil peningkatan citra dengan mother wavelet Biorthogonal 5.5 level 1

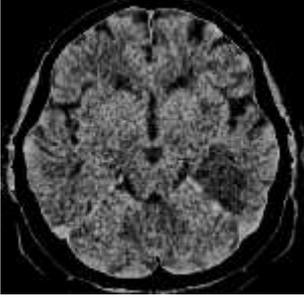
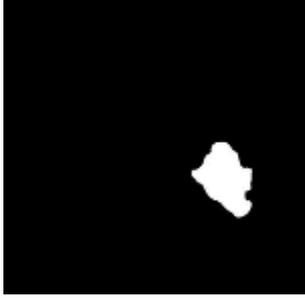
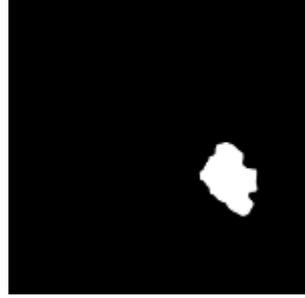
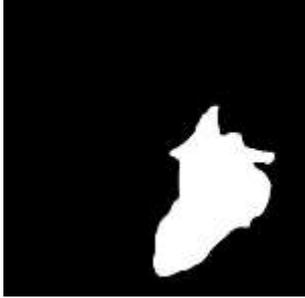
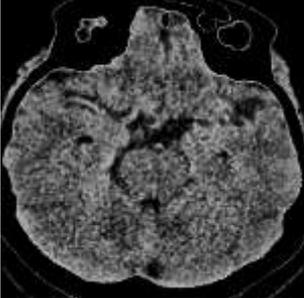
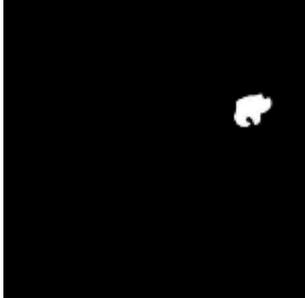
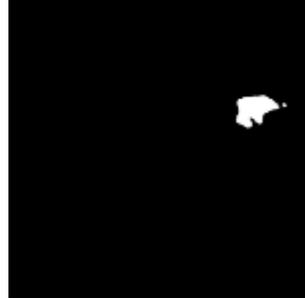
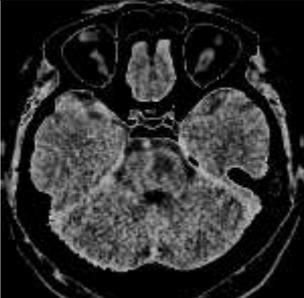
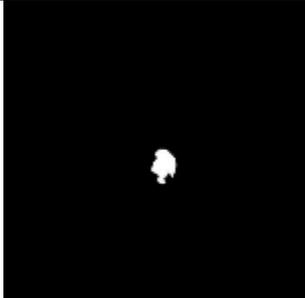
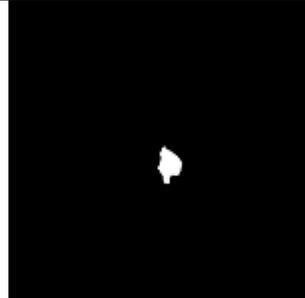
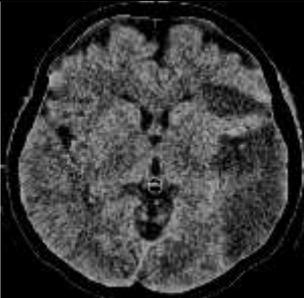
Citra Original	Citra yang telah ditingkatkan
	
	
	
	
	

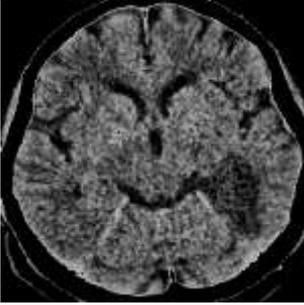
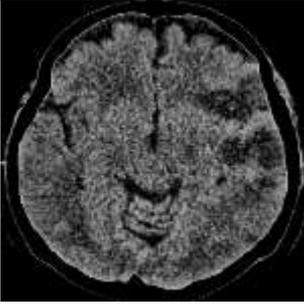
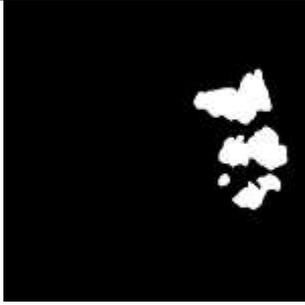
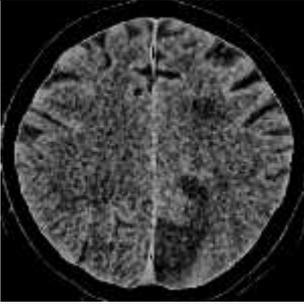
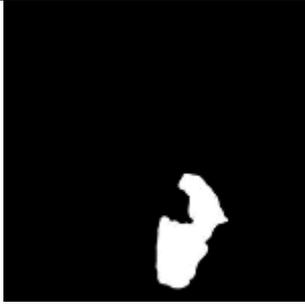
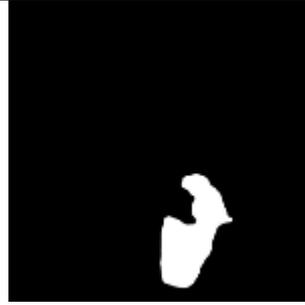
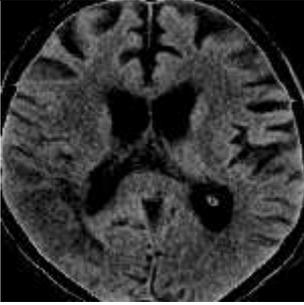
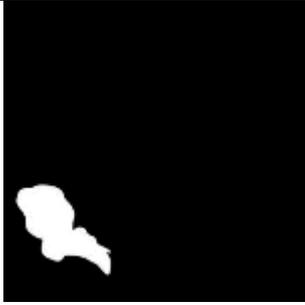
Citra Original	Citra yang telah ditingkatkan
	
	
	
	
	

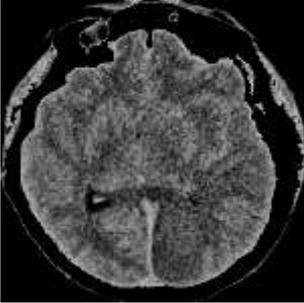
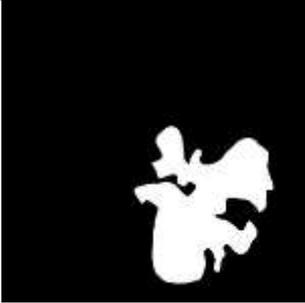
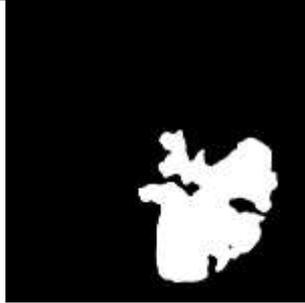
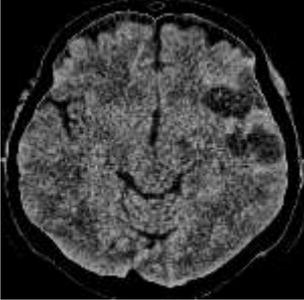
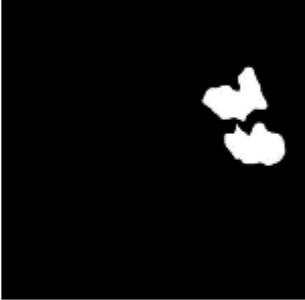
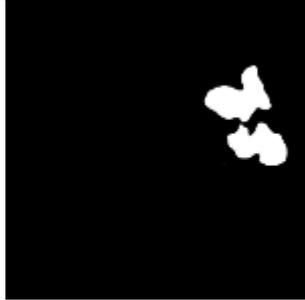
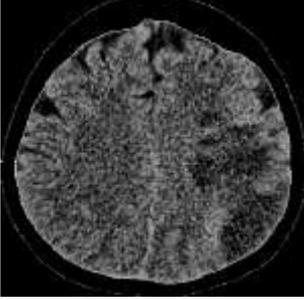
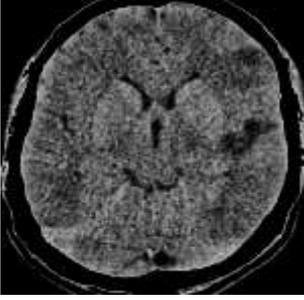
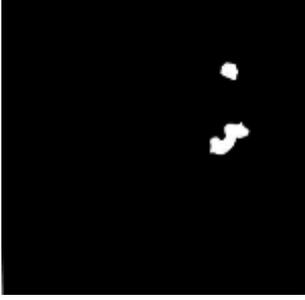
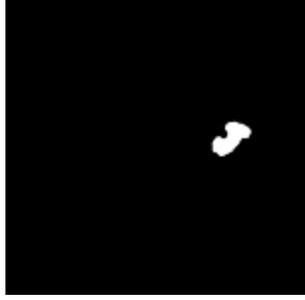
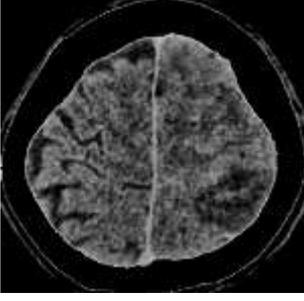
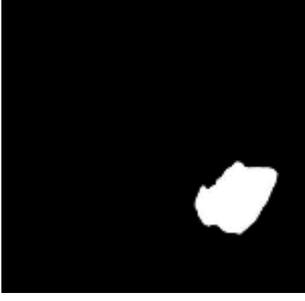
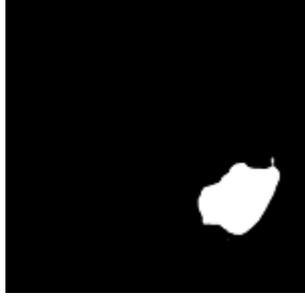
Citra Original	Citra yang telah ditingkatkan
	
	
	
	
	

Hasil selengkapnya dapat dilihat pada *repository* Github pada link berikut:
<https://github.com/bhetrisonia/Hypodense-Segmentation-using-CNN>

C. Hasil segmentasi

<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		
		

<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		
		

<i>Test Case</i>	<i>Ground Truth</i>	<i>Predicted Image</i>
		
		
		
		
		

Hasil selengkapnya dapat dilihat pada *repository* Github pada link berikut:
<https://github.com/bhetrisonia/Hypodense-Segmentation-using-CNN>

BIODATA PENULIS



Penulis dilahirkan di Kabupaten Solok pada tanggal 22 November 1998. Penulis merupakan anak keenam dari tujuh bersaudara. Penulis menempuh pendidikan sekolah dasar di SD Negeri 20 Koto Gaek Guguk dan meneruskan ke jenjang sekolah menengah pertama di SMP Negeri 5 Gunung Talang dan melanjutkan ke sekolah menengah atas SMA Negeri 1 Padang Panjang. Kemudian penulis meneruskan ke jenjang pendidikan yang lebih tinggi ke Departemen Teknik Biomedik, Institut Teknologi Sepuluh Nopember Surabaya, dengan fokus studi pencitraan dan pengolahan citra medika. Selama menempuh pendidikan tinggi, penulis aktif dalam organisasi kemahasiswaan tingkat departemen sebagai kepala divisi kesehatan departemen Keprofesian dan Kesejahteraan Mahasiswa Himpunan Mahasiswa Teknik Elektro (HIMATEKTRO ITS) dan Tim Formatur Himpunan Mahasiswa Teknik Biomedik(HMTB).

E-mail : bhetri.17073@mhs.its.ac.id



**BIOMEDICAL ENGINEERING ITS
2021**