

**TUGAS AKHIR - EB184803**

**OTOMASI SKORING KALSIMUM ARTERI KORONER DARI  
CITRA CT-SCAN MENGGUNAKAN CNN UNTUK  
ASESMEN RISIKO PENYAKIT JANTUNG**

**Aditya Wardianto**  
NRP 0731194000001

Dosen Pembimbing  
**Dr. Tri Arief Sardjono, S.T., M.T**  
NIP 197002121995121001  
**Nada Fitriyatul Hikmah, S.T., M.T**  
NIP 199001072018032001

**Program Studi Sarjana Teknik Biomedik**  
Departemen Teknik Biomedik  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2023



**TUGAS AKHIR - EB184803**

**OTOMASI SKORING KALSIUM ARTERI KORONER DARI  
CITRA CT-SCAN MENGGUNAKAN CNN UNTUK  
ASESMEN RISIKO PENYAKIT JANTUNG**

**Aditya Wardianto**  
NRP 07311940000001

Dosen Pembimbing  
**Dr. Tri Arief Sardjono, S.T., M.T**  
NIP 197002121995121001  
**Nada Fitriyatul Hikmah, S.T., M.T**  
NIP 199001072018032001

**Program Studi Sarjana Teknik Biomedik**  
Departemen Teknik Biomedik  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2023



**FINAL PROJECT - EB184803**

# **AUTOMATION OF CORONARY ARTERY CALCIUM SCORING FROM CT-SCAN IMAGES USING CNN FOR HEART DISEASE RISK ASSESSMENT**

**Aditya Wardianto**  
NRP 07311940000001

Supervisor  
**Dr. Tri Arief Sardjono, S.T., M.T**  
NIP 197002121995121001  
**Nada Fitriyatul Hikmah, S.T., M.T**  
NIP 199001072018032001

**Study Program Bachelor Degree of Biomedical Engineering**  
Department of Biomedical Engineering  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya  
2023

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

Oleh  
Aditya Wardianto  
NRP. 07311940000001

Tanggal Ujian : 07 Juli 2023  
Periode Wisuda : September 2023

Disetujui Oleh :  
Dosen Penguji :



1. Muhammad Yazid, B.Eng., M.Eng.  
NIP. 198004202015041001

Dosen Pembimbing :



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



2. Rezki El Arif, S.T., M.T., Ph.D  
NIP. 2023199211060



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

Kepala Departemen Teknik Biomedik,



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

## PERNYATAAN ORISINALITAS

Yang bertanda tangan di bawah ini:

Nama Mahasiswa / NRP : Aditya Wardianto / 07311940000001  
Program Studi : S1-Teknik Biomedik  
Dosen Pembimbing 1 / NIP : Dr. Tri Arief Sardjono, S.T., M.T /  
197002121995121001  
Dosen Pembimbing 2 / NIP : Nada Fitriyatul Hikmah, S.T., M.T /  
199001072018032001

dengan ini menyatakan bahwa Tugas Akhir dengan judul “Otomasi Skoring Kalsium Arteri Koroner Dari Citra CT-Scan Menggunakan CNN Untuk Asesmen Risiko Penyakit Jantung” adalah hasil karya sendiri, bersifat orisinal, dan ditulis dengan mengikuti kaidah penulisan ilmiah. Bilamana di kemudian hari ditemukan ketidaksesuaian dengan pernyataan ini, maka saya bersedia menerima sanksi sesuai dengan ketentuan yang berlaku di Institut Teknologi Sepuluh Nopember.

Surabaya, 21 Juli 2023

Mahasiswa



Aditya Wardianto  
NRP. 07311940000001

# OTOMASI SKORING KALSIMUM ARTERI KORONER DARI CITRA CT-SCAN MENGGUNAKAN CNN UNTUK ASESMEN RISIKO PENYAKIT JANTUNG

Nama : Aditya Wardianto  
NRP : 07311940000001  
Pembimbing : 1. Dr. Tri Arief Sardjono, S.T., M.T  
2. Nada Fitriyatul Hikmah, S.T., M.T

## ABSTRAK

Penyakit jantung merupakan masalah kesehatan yang cukup serius di Indonesia dengan prevalensinya yang mencapai 1,5% menurut Laporan Nasional Riskesdas tahun 2018. *Coronary Artery Calcium scoring* (skoring CAC) merupakan metode yang dapat digunakan untuk mengukur penumpukan kalsium pada pembuluh darah jantung, yang terkait erat dengan risiko penyakit jantung. Namun, proses skoring CAC terbatas oleh keterbatasan sumber daya, terutama sumber daya ahli dan laboratorium khusus. Dengan perkembangan teknologi *deep learning*, otomasi analisis citra medis dapat mengoptimalkan proses serta membantu tenaga medis dalam melakukan analisis. Penelitian sebelumnya yang menerapkan *deep learning* dalam otomasi skoring CAC belum membahas terkait efisiensi model yang digunakan. Penelitian ini mengembangkan sistem dengan model CNN berbasis UNet++ dengan *pretrained encoder* MobileNetV2 untuk melakukan skoring CAC pada citra 2D CT-Scan *gated*. Model dikembangkan menggunakan dataset COCA dari Stanford AIMI yang berisikan data CT-Scan *gated* 789 pasien. Varian model terbesar (DS4) memiliki 4,1 juta parameter, *dice coefficient* bernilai 0,9378; *Kappa* bernilai 0,813; *precision* bernilai 0,9354; *recall* bernilai 0,6203; dan waktu rata-rata prediksi pasien bernilai  $2,59 \pm 0,4$  detik menggunakan 1 GPU RTX 3090. Turunan dari model terbesar, khususnya varian DS3 memiliki parameter dan waktu rata-rata prediksi lebih rendah dibandingkan model DS4 dengan sedikit penurunan pada performa. Model pada penelitian ini memiliki performa *precision* dan waktu rata-rata prediksi pasien yang lebih baik dibandingkan dengan penelitian sebelumnya dengan penurunan pada performa *dice coefficient*, *recall* dan *Kappa* yang masih berada pada rentang kategori klasifikasi yang hampir sempurna serta bersaing dengan model UNet++ biasa. Arsitektur model CNN yang diusulkan pada penelitian ini dapat melakukan stratifikasi risiko penyakit jantung pasien.

Kata kunci: CAC, penyakit jantung, *deep learning*

# ***AUTOMATION OF CORONARY ARTERY CALCIUM SCORING FROM CT-SCAN IMAGES USING CNN FOR HEART DISEASE RISK ASSESSMENT***

Name : Aditya Wardianto  
NRP : 07311940000001  
Supervisor : 1. Dr. Tri Arief Sardjono, S.T., M.T  
2. Nada Fitriyatul Hikmah, S.T., M.T

## **ABSTRACT**

*Heart disease is a significant health issue in Indonesia with a prevalence rate of 1.5% according to the National Riskesdas Report in 2018. Coronary Artery Calcium scoring (CAC scoring) is a method used to measure calcium buildup in the hearts blood vessels, which is closely related to the risk of heart disease. However, the CAC scoring process is limited by resource constraints, particularly expert resources, and specialized laboratories. With the development of deep learning technology, automated medical image analysis can optimize the process and assist healthcare professionals in analysis. Previous research applying deep learning to automate CAC scoring has not discussed the efficiency of the models used. This study developed a system using a UNet++-based CNN model with a pretrained MobileNetV2 encoder for CAC scoring on 2D gated CT-Scan images. The model was developed using the COCA dataset from Stanford AIMI, which includes gated CT-Scan data from 789 patients. The largest model variant (DS4) had 4.1 million parameters, a dice coefficient of 0.9378, a Kappa value of 0.813, a precision value of 0.9354, a recall value of 0.6203, and an average patient prediction time of  $2.59 \pm 0.4$  second using 1 GPU RTX 3090. Derivatives of the largest model, particularly the DS3 variant, had lower parameters and average prediction times compared to the DS4 model, with a slight decrease in performance. The model in this study demonstrated better performance in precision, and average patient prediction time compared to previous research, with a decrease in dice coefficient, recall and Kappa performance that still falls within the range of nearly perfect classification while still competitive with the regular UNet++ model. The proposed CNN model architecture in this study can stratify the risk of heart disease in patients.*

*Keywords: CAC scoring, heart disease, deep learning*

## KATA PENGANTAR

Penulis mengucapkan puji dan syukur kepada Tuhan Yang Maha Esa, karena hanya dengan kasih sayang-Nya penulis berhasil menyelesaikan Tugas Akhir ini dengan lancar dan tepat waktu. Penulis juga ingin mengucapkan terima kasih kepada berbagai pihak, antara lain:

1. Allah SWT dengan rahmatnya memberikan ketenangan selama proses pengerjaan Tugas Akhir ini
2. Orang tua dan keluarga penulis, almarhum bapak, mama dan kakak yang selalu menyemangati dan menasehati penulis selama ini
3. Bapak Dr. Tri Arief Sardjono, S.T., M.T selaku dosen pembimbing satu yang selalu sabar dalam proses bimbingan selama Tugas Akhir ini
4. Ibu Nada Fitriyatul Hikmah, S.T., M.T. selaku dosen pembimbing dua yang selalu memberikan motivasi selama bimbingan dan mempercayakan penulis dengan Tugas Akhir ini
5. Bapak Ibu Dosen dan Tendik yang telah membantu penulis selama berkuliah di ITS
6. Rekan-rekan BME5 yang saling menyemangati selama Tugas Akhir ini
7. Nura yang selalu membantu di kala penulis kesulitan dalam proses pengerjaan Tugas Akhir ini.
8. Dan seluruh pihak yang tidak bisa penulis cantumkan seluruhnya di lembar kata pengantar ini, yang telah membantu penulis selama perkuliahan ini

Melalui ucapan terima kasih ini, penulis juga menyadari bahwa masih terdapat kekurangan dalam berbagai aspek dan masih banyak hal yang perlu dan dapat penulis perbaiki. Oleh karena itu, penulis terbuka dengan kritik dan saran dari pembaca untuk memperbaiki Tugas Akhir ini. Semoga penelitian ini dapat memberikan manfaat bagi banyak orang dan dapat dikembangkan lebih baik pada penelitian selanjutnya.

Surabaya, Juli 2023

Penulis



## DAFTAR ISI

HALAMAN PENGESAHAN.....	iii
PERNYATAAN ORISINALITAS .....	iv
ABSTRAK .....	v
ABSTRACT.....	vi
KATA PENGANTAR .....	vii
DAFTAR ISI.....	viii
DAFTAR GAMBAR .....	x
DAFTAR TABEL.....	xi
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	2
1.3. Batasan Masalah.....	2
1.4. Tujuan dan Manfaat Penelitian .....	2
1.5. Kontribusi.....	3
BAB II TINJAUAN PUSTAKA.....	4
2.1. Korelasi <i>Coronary Artery Calcium Score</i> dan Penyakit Jantung.....	4
2.2. Otomasi <i>Coronary Artery Calcium Scoring</i> .....	5
2.3. Model CNN Enkoder-Dekoder UNet dan UNet++.....	6
2.4. Optimasi Konvolusi .....	7
2.5. Rasional Penelitian.....	8
BAB III METODE PENELITIAN.....	10
3.1. Desain Sistem.....	10
3.2. Data Citra <i>CT-Scan Gated</i> .....	10
3.3. <i>Preprocessing</i> Data .....	11
3.3.1. <i>Preprocessing</i> Data Segmentasi.....	11
3.3.2. <i>Preprocessing</i> Citra.....	14
3.4. Model <i>CNN</i> .....	14
3.5. <i>Postprocessing</i> Data.....	19
3.5.1. <i>Connected Component Labelling</i> .....	19
3.5.2. Skoring <i>Agatston</i> .....	20
3.6. <i>Graphical User Interface</i> .....	21
3.7. Uji Coba Sistem .....	23
BAB IV PENGUJIAN SISTEM .....	25
4.1. Pengujian <i>Preprocessing</i> .....	25
4.1.1. Pengujian Seleksi Data dan <i>Random Undersampling</i> .....	25
4.1.2. Pengujian <i>Preprocessing</i> Segmentasi .....	27
4.1.3. Pengujian <i>Preprocessing</i> Citra.....	27
4.2. Pengujian Model .....	27
4.3. Pengujian Sistem.....	32
BAB V PEMBAHASAN .....	35
5.1. Pembahasan Hasil Pengujian <i>Preprocessing</i> .....	35

5.1.1.	Pembahasan Hasil Pengujian Seleksi Data dan <i>Random Undersampling</i> .....	35
5.1.2.	Pembahasan Hasil Pengujian <i>Preprocessing</i> Segmentasi.....	35
5.1.3.	Pembahasan Hasil Pengujian <i>Preprocessing</i> Citra .....	36
5.2.	Pembahasan Hasil Pengujian Model .....	36
5.3.	Pembahasan Hasil Pengujian Sistem .....	37
5.4.	Perbandingan Dengan Penelitian Sebelumnya.....	38
BAB VI KESIMPULAN .....		40
6.1.	Kesimpulan .....	40
6.2.	Saran.....	40
DAFTAR PUSTAKA .....		41
BIODATA PENULIS .....		44

## DAFTAR GAMBAR

Gambar 2.1 Citra CT dan Nilai Agatston [10].....	4
Gambar 2.2 Perbandingan Model Pada Penelitian Eng et al dan Hong et al [7] [6]5	
Gambar 2.3 Perbandingan Model UNet dan UNet++ [11][12] .....	7
Gambar 2.4 Perbandingan <i>Depthwise Seperable Convolution</i> dan <i>Inverted Residual Block</i> [8][9] .....	8
Gambar 2.5 Diagram <i>Fishbone</i> .....	9
Gambar 3.1 Diagram Blok Sistem Otomasi Skoring CAC .....	10
Gambar 3.2 Diagram Blok <i>Preprocessing</i> .....	12
Gambar 3.3 Arsitektur UNet++ [12] dengan <i>Pretrained MobileNetV2 Encoder</i> [9] .....	17
Gambar 3.4 Blok Diagram <i>Postprocessing Pipeline</i> .....	19
Gambar 3.5 Tampilan Depan GUI.....	21
Gambar 3.6 Tampilan GUI Pada Mode <i>Breakdown</i> .....	22
Gambar 3.7 Tampilan GUI Pada Mode <i>Whole Scan</i> .....	22
Gambar 3.8 Tampilan Hasil Skoring Pada Mode <i>Whole Scan</i> .....	22
Gambar 3.9 Tampilan GUI Dengan Hasil Skoring Pada Mode <i>Breakdown</i> .....	23
Gambar 4.1 Jumlah Citra Data <i>Training</i> Sebelum dan Sesudah <i>Undersampling</i> . 26	
Gambar 4.2 Jumlah Piksel Data <i>Training</i> Sebelum dan Sesudah <i>Undersampling</i> 26	
Gambar 4.3 Jumlah Piksel dan Citra Data Validasi dan <i>Testing</i> .....	26
Gambar 4.4 Perbandingan vektor koordinat <i>floating point</i> .....	27
Gambar 4.5 Perbandingan Rentang Nilai Citra Selama Proses <i>Preprocessing</i> ....	28
Gambar 4.6 Contoh Hasil <i>True Positive Model</i> .....	29
Gambar 4.7 Contoh Hasil <i>True Negative Model</i> .....	29
Gambar 4.8 Contoh Hasil <i>False Positive Model</i> .....	30
Gambar 4.9 Contoh Hasil <i>False Negative Model</i> .....	30
Gambar 4.10 Contoh 1 Hasil Model Yang Terdegradasi.....	31
Gambar 4.11 Contoh 2 Hasil Model Yang Terdegradasi.....	31
Gambar 4.12 <i>Confusion Matrix Pruning Model DS1</i> .....	32
Gambar 4.13 <i>Confusion Matrix Pruning Model DS2</i> .....	33
Gambar 4.14 <i>Confusion Matrix Pruning Model DS3</i> .....	33
Gambar 4.15 <i>Confusion Matrix Pruning Model DS4</i> .....	34

## DAFTAR TABEL

Tabel 3.1 Arsitektur UNet++ dengan <i>Pretrained MobileNetV2 Encoder</i> .....	18
Tabel 3.2 Tabel Konversi <i>Weighting Factor</i> terhadap <i>Hounsfield Unit</i> [6] .....	20
Tabel 3.3 Tingkat Keparahan Kalsifikasi Arteri Koroner [26] .....	21
Tabel 4.1 Perbandingan Metrik Performa <i>Pruning</i> Model Pada Data <i>Testing</i> .....	28
Tabel 4.2 Perbandingan Performa <i>Pruning</i> Model Pada Sistem .....	32
Tabel 5.1 Perbandingan Performa Model Dan Sistem Penelitian Ini Terhadap Penelitian Dengan Pendekatan <i>One Shot</i> Sebelumnya .....	39

# BAB I

## PENDAHULUAN

### 1.1. Latar Belakang

Melalui data yang dihimpun WHO pada tahun 2019, penyakit kardiovaskular merupakan penyebab kematian utama di dunia dengan angka kematian 17,9 juta jiwa atau sebesar 32%. Dari jumlah kematian tersebut, 85% diantaranya disebabkan oleh serangan jantung dan *stroke* [1]. Berdasarkan Laporan Nasional Risesdas 2018, prevalensi penyakit jantung di Indonesia mencapai 1,5% yang berarti 15 dari 1000 orang Indonesia menderita penyakit jantung [2]. Dengan tingginya angka kematian di dunia, penyakit kardiovaskular merupakan penyakit yang perlu dideteksi sedini mungkin. Penyakit jantung memiliki jangka asimtomatik yang panjang sehingga terdapat jendela waktu dalam intervensi preventif dini melalui asesmen risiko [3]. Berdasarkan *American College of Cardiology* dan *American Heart Association*, asesmen risiko direkomendasikan untuk dilakukan menggunakan *Pooled Cohort Equation* yang dikembangkan pada tahun 2013 sebagai pengembangan dari metode asesmen risiko sebelumnya. Namun, metode ini memiliki kelemahan karena praktisi klinis harus mengaplikasikan *Pooled Cohort Equation* yang dikembangkan berdasarkan sampel pasien, sehingga metode ini tidak efektif jika pasien memiliki etnis dan ras yang tidak ada dalam sampel pasien yang digunakan [4].

Skoring *CAC* atau *Coronary Artery Calcification scoring* merupakan metode kuantifikasi penumpukan kalsium pada pembuluh darah jantung yang dapat digunakan sebagai penanda yang memberikan informasi dan memiliki korelasi tinggi terhadap risiko penyakit jantung dibandingkan dengan faktor risiko tradisional penyakit jantung [5]. Skoring *CAC* terbatas dalam penggunaannya dalam asesmen risiko akibat besarnya sumber daya yang dibutuhkan untuk melakukan asesmen, mulai dari penggunaan *CT scan* yang memerlukan pengamatan khusus, sumber daya ahli yang terbatas hingga terlewatnya kesempatan asesmen risiko bagi pasien dengan *CT scan* berkala [6]. Hal ini mengindikasikan terdapat ruang untuk melakukan optimasi dalam pemrosesan skoring *CAC*.

Riset-riset terkini mulai menerapkan teknik *deep learning* untuk melakukan analisis pada citra medis [6]. Penelitian sebelumnya berhasil melakukan otomatisasi skoring *CAC* seperti penelitian oleh Eng et al [6] dan Hong et al [7], namun belum membahas sepenuhnya mengenai perbandingan kecepatan dan akurasi dari model yang dikembangkan. Model seperti *MobileNetV1* oleh Howard et al [8] dan *MobileNetV2* oleh Sandler et al [9] melakukan efisiensi model melalui metode alternatif dari konvolusi tradisional untuk mengurangi jumlah komputasi. Penerapan model yang lebih efisien dapat mempermudah pengaplikasian model dalam kondisi nyata dan meningkatkan adopsi model secara langsung di bidang medis.

## 1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan sebelumnya, skoring CAC dapat meningkatkan performa asesmen risiko penyakit jantung, namun pengaplikasiannya mengalami hambatan akibat sumber daya yang terbatas. Oleh karena itu, rumusan masalah pada penelitian ini terdiri dari tiga poin. Pertama, perancangan model *deep learning* menggunakan arsitektur UNet++ dengan *pretrained encoder inverted residual block* MobileNetV2 sebagai dasar sistem otomasi skoring CAC. Kedua, pengintegrasian model dalam suatu sistem otomasi yang mengolah data segmentasi menjadi skor CAC (nilai *Agatston*). Terakhir, evaluasi sistem otomasi dan perbandingannya dengan penelitian sebelumnya.

## 1.3. Batasan Masalah

Batasan masalah dalam penelitian ini adalah penggunaan data sekunder yang disediakan oleh laboratorium Stanford AIMI untuk proses *training* pada model *deep learning*. Data yang digunakan berupa pasangan data citra *gated CT scan* dan data segmentasi kalsium pembuluh darah jantung yang dibatasi pada jumlah tertentu. Selain itu, proses evaluasi pada sistem otomasi dilakukan pada data *testing* yang didapatkan dari pembagian data. Selain itu, penelitian ini juga dibatasi pada penggunaan metode *deep learning* berbasis arsitektur UNet++ dengan *pretrained encoder inverted residual block* MobileNetV2, sehingga metode-metode lain tidak dieksplorasi lebih lanjut dalam penelitian ini.

## 1.4. Tujuan dan Manfaat Penelitian

Tugas akhir ini bertujuan untuk mengembangkan sebuah sistem otomasi skoring CAC yang dapat meningkatkan keakuratan dan efisiensi proses diagnosis CAC pada pasien penyakit jantung koroner. Sistem ini akan dirancang dengan menggunakan metode *deep learning* berbasis arsitektur UNet++ dengan *pretrained encoder inverted residual block* MobileNetV2, dengan tujuan menghasilkan model yang mampu melakukan klasifikasi skor CAC (nilai *Agatston*) secara akurat dan efisien.

Tugas akhir ini diharapkan dapat memberikan manfaat bagi tenaga medis dan pasien dalam proses diagnosis dini penyakit jantung koroner. Dengan adanya sistem otomasi skoring CAC yang dirancang menggunakan metode *deep learning* berbasis arsitektur CNN UNet++ dan MobilNetV2, diharapkan proses identifikasi nilai CAC (nilai *Agatston*) pada pasien dapat dipermudah dan dipercepat. Dalam jangka panjang, sistem ini diharapkan dapat menambah penelitian baru yang membantu radiolog dan dokter dalam membuat keputusan yang lebih akurat dan tepat mengenai penanganan pasien berdasarkan asesmen risiko penyakit jantung.

### **1.5. Kontribusi**

Kontribusi ilmiah dari penelitian ini adalah, melakukan pengembangan dari penelitian sebelumnya terhadap otomasi proses skoring *CAC* dan memberikan kesempatan pengembangan topik lebih lanjut dari penelitian ini.

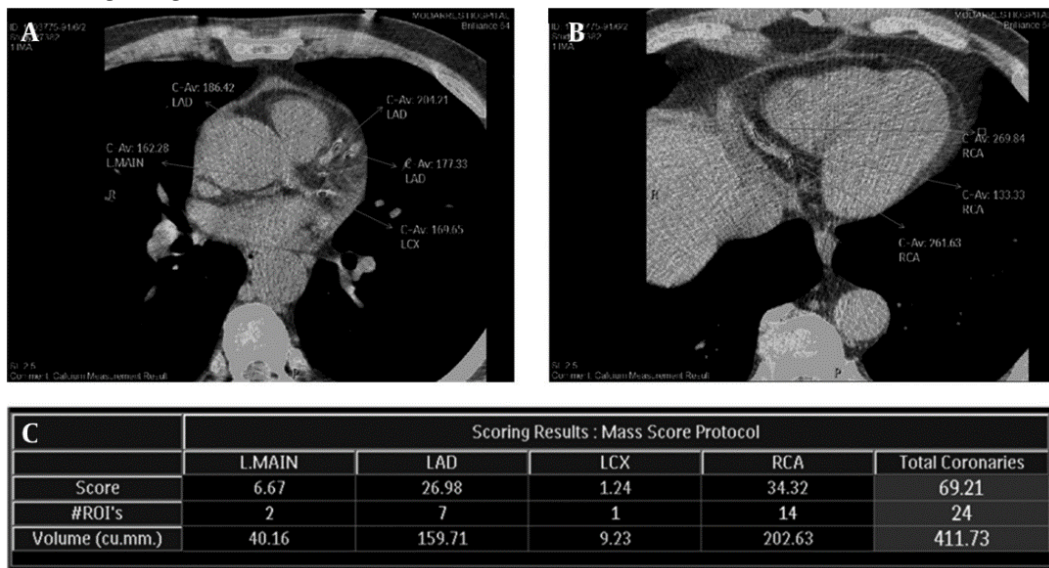
Kontribusi praktis dari penelitian ini adalah, memberikan alternatif metode otomasi skoring *CAC* untuk mempermudah dokter dalam mengambil keputusan pasca asesmen risiko penyakit jantung terkait intervensi preventif dini yang mampu meningkatkan kualitas dan harapan hidup pasien.

## BAB II TINJAUAN PUSTAKA

### 2.1. Korelasi *Coronary Artery Calcium Score* dan Penyakit Jantung

*Coronary artery calcium (CAC)* merupakan fitur yang sangat spesifik yang terdapat pada kondisi aterosklerosis jantung. Berbagai penelitian baik jangka panjang maupun pendek telah menunjukkan bahwa *CAC* selalu muncul sebagai fitur yang konsisten dan dapat direproduksi dalam asesmen risiko penyakit jantung. Fitur ini sangat berguna dalam perencanaan intervensi preventif pada kasus pasien asimtomatik, seperti pemberian statin dan aspirin [5].

Pada awalnya, deteksi *CAC* dilakukan menggunakan metode tradisional seperti radiografi dada, fluoroskopi, atau fluoroskopi pengurangan digital. Namun, dengan kemajuan teknologi dalam bidang pencitraan jantung, *CAC* telah diakui sebagai fitur yang mampu memprediksi keberadaan penyakit pembuluh darah jantung. Metode *cardiac gating*, yang pertama kali diperkenalkan pada teknik *electron-beam computed tomography (EBCT)*, memungkinkan deteksi, lokalisasi, dan pengukuran *CAC* dengan tingkat sensitivitas yang lebih tinggi. Penerapan metode EBCT ini memberikan keunggulan dalam resolusi temporal jantung yang berdetak. Namun, dalam penggunaannya secara umum, metode EBCT kurang optimal dalam prosedur CT *scan* secara umum, sehingga perlahan digantikan oleh metode *multi-detector computed tomography (MDCT)*. MDCT memberikan keunggulan dalam fleksibilitas, memungkinkan pengukuran *CAC* tanpa memerlukan prosedur *cardiac gating* yang kompleks. Pengukuran *CAC* pada metode MDCT yang tidak menggunakan *cardiac gating* dapat memberikan nilai kuantitatif dari nilai *CAC* dengan korelasi yang cukup tinggi dengan metode *cardiac gating* [5].



Gambar 2.1 Citra CT dan Nilai Agatston [10]

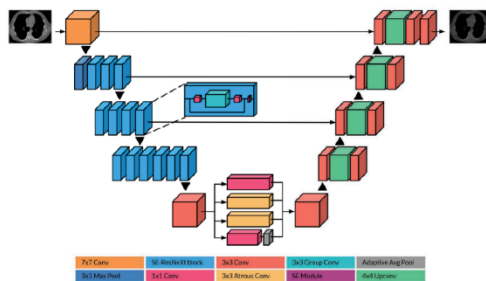


Metode kuantifikasi CAC pada citra CT scan pertama kali dikembangkan oleh Agatston et al. pada tahun 1990 [10]. Metode ini melibatkan perhitungan nilai berdasarkan jumlah, luasan, dan intensitas piksel pada setiap lesi yang terdeteksi di setiap *slice* citra CT. Dalam prosesnya, setiap lesi diberi nilai Agatston yang mencerminkan tingkat kalsifikasi pada lesi tersebut. Nilai Agatston total dari seorang pasien diperoleh dengan menjumlahkan nilai Agatston dari setiap lesi yang terdeteksi pada masing-masing pembuluh darah jantung, seperti *Left Main Coronary Artery*, *Left Anterior Descending*, *Left Circumflex*, dan *Right Coronary Artery*, di setiap *slice* citra CT. Pada Gambar 2.1, dapat dilihat visualisasi citra CT yang telah melalui proses segmentasi dan perhitungan nilai Agatston untuk memberikan gambaran yang lebih jelas mengenai keberadaan dan distribusi CAC pada arteri koroner pasien.

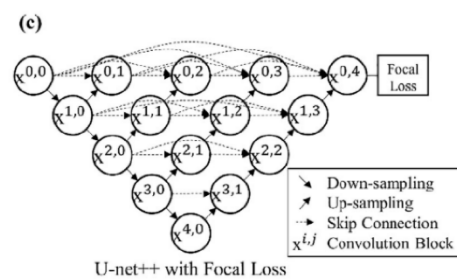
### 2.2. Otomasi Coronary Artery Calcium Scoring

Asesmen risiko merupakan salah satu metode primer dalam pencegahan penyakit jantung, di mana skoring CAC menggunakan CT scan menjadi salah satu metode non-invasif yang umum digunakan. Meskipun secara klinis terbukti memiliki korelasi yang tinggi dengan risiko penyakit jantung, penggunaan metode ini masih memiliki beberapa batasan. Batasan-batasan tersebut antara lain terkait dengan keterbatasan sumber daya dan ahli yang belum mencukupi, serta pasien-pasien yang menjalani CT scan rutin namun memiliki penyakit jantung yang tidak terdeteksi karena tidak dilakukan skoring CAC [6].

Dalam rangka mengatasi batasan tersebut, beberapa penelitian telah menggunakan CNN untuk mengotomasi proses skoring CAC. Sebagai contoh, penelitian yang dilakukan oleh Eng et al. menghasilkan dua model otomatisasi skoring CAC yang menggunakan input dari CT *scan non-gated* dan *gated*, dengan output berupa Nilai Agatston untuk empat arteri koroner, yaitu LCA, LAD, LCX, dan RCA. Pendekatan ini dipilih karena tidak memerlukan segmentasi manual dan *cardiac atlas* untuk lokalisasi arteri koroner, sehingga tidak diperlukan penggunaan kontras saat melakukan CT *scan* [6].



Eng et al Model Architecture



Hong et al Model Architecture

Gambar 2.2 Perbandingan Model Pada Penelitian Eng et al dan Hong et al [6] [7]

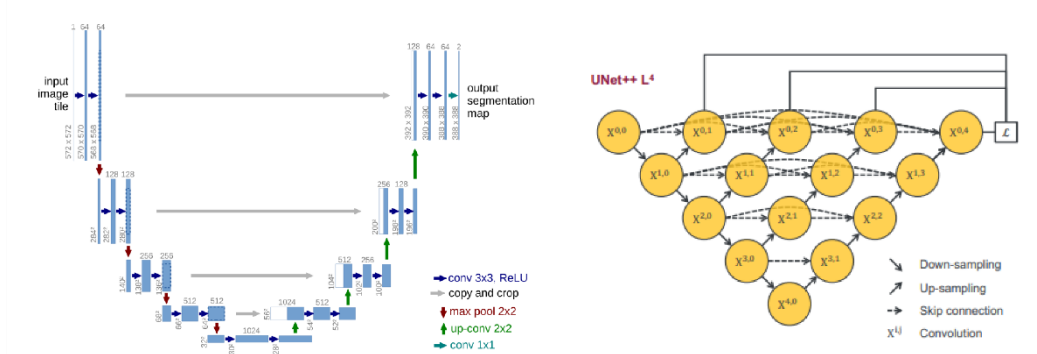
Data yang digunakan dalam penelitian Eng et al. terdiri dari data CT *scan gated* dan *non-gated* yang diperoleh dari *Stanford Hospital and Clinics*, serta data eksternal dari berbagai universitas, perusahaan, dan pusat akademik. Data tersebut kemudian disegmentasi secara manual oleh radiolog dan diubah skala menjadi 512x512 piksel untuk digunakan dalam proses pelatihan model [6].

Metode algoritma yang digunakan dalam kedua model terdiri dari tiga tahap. Tahap pertama melibatkan penggunaan model CNN dengan arsitektur *decoder-encoder* seperti pada UNet, yang berfungsi untuk menghasilkan probabilitas kalsium serta lokasi kalsium pada citra. Selanjutnya, pada tahap kedua dilakukan analisis *connected components* yang mengolah probabilitas tersebut menjadi segmentasi area kalsium, yang kemudian dapat digunakan untuk menghitung Nilai Agatston. Pada tahap ketiga, Nilai Agatston digunakan untuk melakukan klasifikasi risiko penyakit jantung pasien menggunakan *gradient boosted decision tree classifier* [6].

Penelitian lain yang relevan dilakukan oleh Hong et al. menggunakan model *Convolutional Neural Network* (CNN) dengan fungsi *loss* yang berbeda, yaitu menggunakan UNet++ dengan tambahan focal loss sebagai fungsi *loss*. Pendekatan ini memiliki beberapa perbedaan mendasar dengan pendekatan yang digunakan oleh Eng et al. Salah satu perbedaan utama adalah pada metode *post-processing* yang digunakan untuk menghitung Nilai Agatston. Pada penelitian Hong et al., metode *post-processing* melibatkan penggunaan *connected component labelling* pada voxel citra dan mengabaikan lesi dengan luasan lebih kecil dari 1 mm<sup>2</sup> [7]. Perbandingan antara model yang digunakan oleh Eng et al dan Hong et al dapat dilihat pada Gambar 2.2.

### 2.3. Model CNN Enkoder-Dekoder UNet dan UNet++

Model dengan arsitektur *encoder-decoder* sering digunakan untuk melakukan segmentasi semantik dan *instance* dalam berbagai penelitian. Salah satu model yang diusulkan oleh Ronneberger et al. adalah UNet, yang dikembangkan untuk mengatasi kendala ketersediaan data yang pada saat itu membatasi efektivitas model CNN [11]. Tingkat akurasi yang tinggi dari model ini dapat diatribusikan pada penggunaan *skip connection* yang menggabungkan *feature map* kasar pada bagian decoder dengan *feature map* halus pada bagian *encoder*. Strategi ini terbukti efektif dalam memulihkan detail halus pada objek yang tersegmentasi, meskipun latar belakang citra kompleks. Namun, arsitektur ini memiliki dua batasan penting. Pertama, kedalaman antara satu model dengan model lainnya harus disesuaikan dengan kompleksitas masalah dan jumlah data yang digunakan, sehingga proses pelatihan dan penggunaan menjadi sulit. Kedua, penggunaan *skip connection* memaksa *feature map* pada *decoder* dan *encoder* memiliki skala yang sama, sedangkan secara semantik *feature map* keduanya berbeda, dan belum ada teori konkret yang memastikan bahwa penggabungan keduanya adalah metode terbaik [12].



UNet by Ronneberg et al

UNet++ by Zhou et al

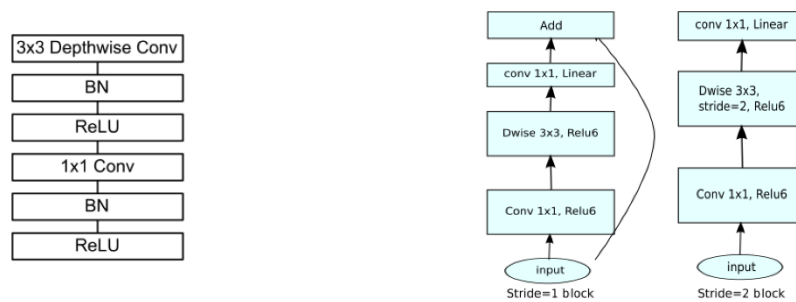
Gambar 2.3 Perbandingan Model UNet dan UNet++ [11][12]

Penelitian selanjutnya oleh Zhou et al. mengembangkan arsitektur UNet++ sebagai pengembangan lanjutan dari UNet. Model UNet++ dirancang untuk mengatasi batasan-batasan yang ada pada arsitektur sebelumnya. Perbedaan utama arsitektur ini terletak pada dua keunggulannya. Pertama, kedalaman model tidak lagi menjadi kendala karena UNet++ telah menyematkan model UNet dengan berbagai kedalaman di dalamnya. Hal ini memungkinkan pelatihan UNet yang telah disematkan dilakukan secara bersamaan, sehingga proses pelatihan dan *pruning* model menjadi lebih mudah. Kedua, arsitektur UNet++ tidak dibatasi dalam skala *feature map* karena penggunaan skip connection pada UNet++ memberikan kebebasan dalam menentukan skala *feature map* pada bagian *decoder*, sehingga model dapat mengatur aliran *feature map* dari *encoder* ke *decoder* sesuai kebutuhan [12].

Proses eksperimen dilakukan pada model UNet++ menggunakan enam set data biologis, kemudian performanya dibandingkan dengan arsitektur UNet. Hasil eksperimen menunjukkan bahwa arsitektur UNet++ mampu memberikan segmentasi citra yang lebih akurat pada keenam set data, serta memberikan keunggulan performa yang konsisten dibandingkan dengan model-model lain yang digunakan pada citra medis, seperti UNet, *wide UNet*, dan UNet+ [12]. Perbandingan arsitektur antara model UNet dan UNet++ dapat dilihat pada Gambar 2.3.

## 2.4. Optimasi Konvolusi

Optimasi konvolusi adalah salah satu aspek penting dalam pengembangan jaringan saraf konvolusional (CNN) untuk meningkatkan efisiensi dan kinerja komputasi. Salah satu pendekatan yang berhasil dalam mengoptimalkan konvolusi adalah dengan menggunakan metode *depthwise separable convolution* yang diperkenalkan pada penelitian MobileNet oleh Howard et al. Metode ini telah terbukti efektif dalam mengurangi beban komputasi dengan menggabungkan dua jenis konvolusi: *depthwise convolution* dan *pointwise convolution* [8].



**Depthwise Seperable Convolution** by Howard et al      **Inverted Residual Block** by Sandler et al

Gambar 2.4 Perbandingan *Depthwise Seperable Convolution* dan *Inverted Residual Block* [8][9]

*Depthwise convolution* dilakukan pada setiap saluran input secara terpisah, menghasilkan fitur yang lebih ringan secara komputasi. Setelah itu, *pointwise convolution* dilakukan untuk menggabungkan fitur-fitur tersebut dengan menggunakan konvolusi linier. Pendekatan ini membantu mengurangi kompleksitas perhitungan dengan mempertahankan informasi fitur yang penting.

Namun, terdapat teknik lain yang digunakan untuk mengoptimasi konvolusi, seperti penggunaan kernel berukuran kecil yang dapat mengurangi jumlah parameter dan kompleksitas komputasi seperti halnya pada *pointwise convolution*. Pendekatan ini memungkinkan model CNN untuk menjalankan operasi konvolusi dengan lebih cepat dan lebih efisien [8].

Dalam penelitian MobileNetV2 oleh Sandler et al yang mengembangkan teori dari penelitian Howard et al, proses optimia konvolusi dilakukan melau implementasi *inverted residual block*. Struktur ini memanfaatkan *linear bottleneck* untuk mengekstraksi informasi pada dimensi yang rendah, serta *inverted residuals* untuk meningkatkan propagasi sinyal antar lapisan. MobileNetV2 berhasil memberikan hasil yang lebih akurat dengan kompleksitas yang lebih rendah dibandingkan dengan beberapa arsitektur model lain dalam berbagai tugas pengenalan objek [9]. Perbandingan *depthwise seperable convolution* dan *inverted residual block* dapat dilihat pada Gambar 2.4.

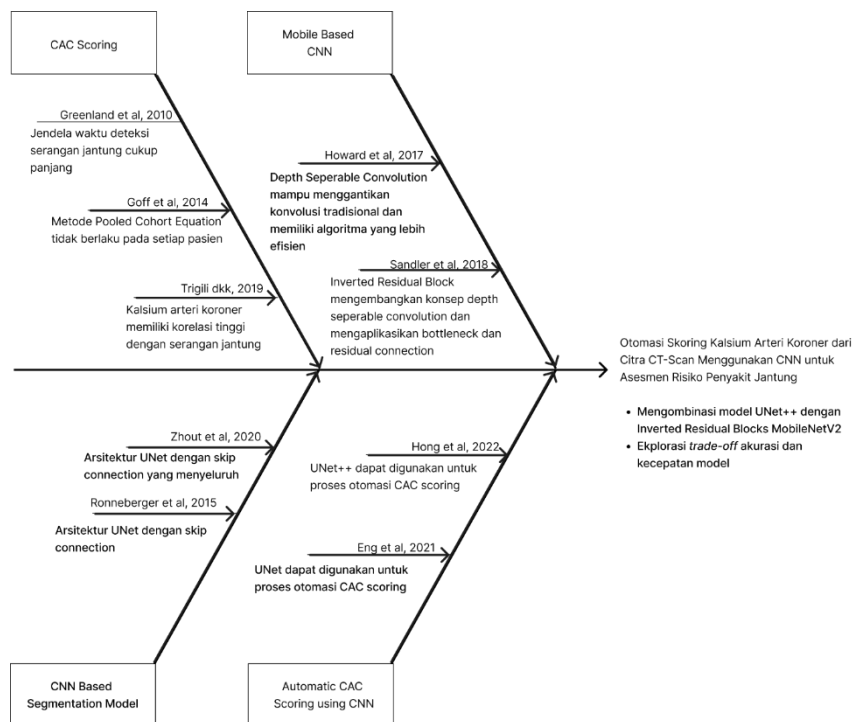
## 2.5. Rasional Penelitian

Penelitian ini bertujuan untuk mengembangkan sebuah sistem otomatisasi skoring kalsium arteri koroner dari citra CT-Scan menggunakan CNN untuk asesmen risiko penyakit jantung. Model CNN yang digunakan dalam penelitian ini merupakan kombinasi antara UNet++, yang dikembangkan oleh Zhou et al. [12], dengan *inverted residual block* yang dikembangkan oleh Sandler et al. [9].

Pemilihan kedua model tersebut dilakukan sebagai pengembangan dari model-model sebelumnya yang telah diteliti oleh Eng et al. [6], yang menggunakan arsitektur UNet yang dikembangkan oleh Ronneberger et al. [14], dan model yang dikembangkan oleh Hong et al. [7] dengan menggunakan arsitektur UNet++ yang juga dikembangkan oleh Zhou et al. [12]. Pada penelitian ini, fokus diberikan pada pengoptimalan model untuk memudahkan aksesibilitas, dan untuk itu digunakanlah inverted residual block yang berasal dari MobileNetV2 yang dikembangkan oleh Sandler et al. [9]. MobileNetV2 sendiri merupakan pengembangan dari *depthwise separable convolution* yang diperkenalkan oleh MobileNetV1 yang dikembangkan oleh Howard et al. [8].

Selain itu, dalam penelitian ini juga dilakukan tinjauan terhadap penggunaan skor kalsium arteri koroner dalam asesmen risiko serangan jantung. Konsep ini sebelumnya telah dibahas secara mendalam oleh Greenland pada tahun 2018 [5] dan 2010 [3], serta Goff pada tahun 2014 [4]. Penelitian sebelumnya tersebut memberikan dasar teoritis yang kuat dalam penerapan kalsium arteri koroner sebagai indikator risiko penyakit jantung.

Dengan menggabungkan kontribusi penelitian sebelumnya, penelitian ini berupaya untuk mengoptimalkan model CNN dalam sistem otomatisasi skor kalsium arteri koroner. Melalui pengembangan arsitektur menggunakan UNet++ dan *inverted residual block*, diharapkan sistem yang dikembangkan dapat memberikan hasil yang lebih akurat dan efisien dalam asesmen risiko penyakit jantung berdasarkan citra CT-Scan.



## BAB III METODE PENELITIAN

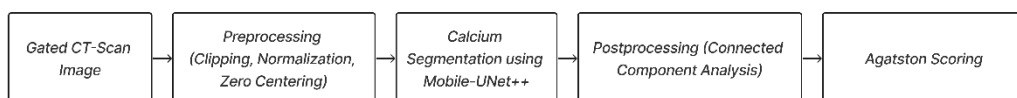
### 3.1. Desain Sistem

Sistem otomatis yang dirancang terdiri dari beberapa tahap yaitu, pengambilan citra, segmentasi gambar menggunakan model arsitektur UNet++ dengan *pretrained encoder inverted residual block* MobileNetV2, *post-processing* menggunakan *connected component labelling*, dan diakhiri dengan kalkulasi nilai *Agatston*. Pengambilan citra tidak dilakukan secara langsung (data primer), melainkan menggunakan data sekunder berupa citra CT-Scan *gated* yang didapatkan dari *Stanford University COCA Dataset* berbentuk DICOM. Citra ini kemudian diteruskan ke model dan proses inferensi dilakukan per lapisan dari data citra pasien. Hasil segmentasi dari model ini, kemudian diteruskan ke *post-processing* menggunakan *connected-component labelling* yang mengelompokkan hasil segmentasi pada citra. Hasil segmentasi yang sudah *dipost-processing* kemudian dapat digunakan untuk melakukan perhitungan nilai *Agatston* yang digunakan untuk asesmen risiko penyakit jantung. Sistem otomatis skoring kalsium arteri koroner yang diajukan dapat dilihat pada Gambar 3.1.

### 3.2. Data Citra CT-Scan Gated

Data yang digunakan pada penelitian ini merupakan data sekunder yang didapatkan melalui *Stanford Center for Artificial Intelligence in Medicine and Imaging (AIMI)*. Data ini merupakan salah satu data utama yang digunakan pada penelitian mengenai otomatisasi skoring CAC menggunakan *deep learning* oleh Eng et al. Proses pengambilan data sudah disetujui oleh *Review Board* dari institusi terkait dan *form consent* sudah didapatkan untuk penggunaan data tersebut [6]. Penggunaan data COCA memerlukan persetujuan terhadap *Research User Agreement* dan *Term of Use* dari *Stanford University School of Medicine* yang membatasi penggunaan data untuk keperluan riset non-komersil.

Dataset COCA merupakan kumpulan dari dua jenis data yang berbeda. Tipe data pertama merupakan data *ECG-gated CT* koroner dalam bentuk DICOM dengan segmentasi dan nilai kalsium koroner dalam bentuk xml. Tipe data kedua merupakan data *non-gated CT* dada dalam bentuk DICOM dan nilai kalsium koroner dalam bentuk csv. Penelitian ini akan menggunakan data *ECG-gated CT* koroner sebagai data utama pada proses *training* dan validasi model yang akan dikembangkan.



Gambar 3.1 Diagram Blok Sistem Otomasi Skoring CAC

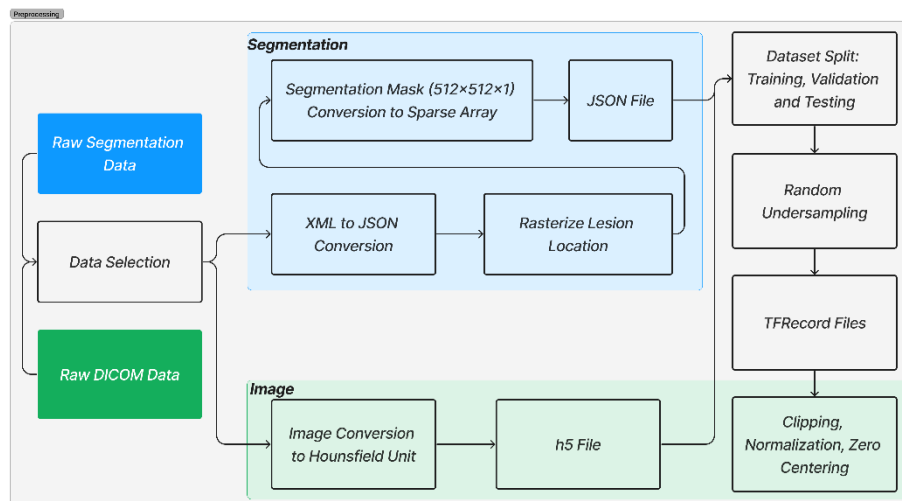
Penggunaan data *ECG-gated CT* pada penelitian ini didasari oleh dua faktor. Faktor pertama, *ECG-gated CT* merupakan referensi standar dalam pendeteksian dan kuantifikasi *CAC* [13]. Meta analisis oleh Kim et al pada 16 penelitian terhadap reliabilitas *non-gated CT* pada asesmen *CAC* menunjukkan terdapat korelasi yang tinggi antara asesmen risiko *gated CT* dan *non-gated CT* [14]. Namun, pedoman *Society of Cardiovascular Computed Tomography* bersama *Society of Thoracic Radiology* pada tahun 2016, merekomendasikan evaluasi dan pelaporan hasil *CAC* pada citra *non-gated CT*, khususnya pada pasien dengan umur lebih dari 40 tahun, dikelompokkan secara kualitatif menjadi, *none*, *mild*, *moderate*, atau *severe* [15], sehingga tidak ada pelaporan dari hasil *CAC* dalam bentuk segmentasi lesi pada citra. Faktor kedua, sesuai dengan standar pelaporan kualitatif *CAC* pada *non-gated CT*, *dataset COCA* tidak memiliki data segmentasi kalsium. Pengembangan model menggunakan *non-gated CT* oleh Eng et al memerlukan tambahan data dari *dataset MESA* yang berpasangan (*gated* dan *non-gated*) dengan tetap menggunakan *gated CT* sebagai *ground truth* [6].

### 3.3. Preprocessing Data

Data citra *gated* berbentuk DICOM dan xml yang didapatkan dari dataset *COCA* merupakan data mentah yang didapatkan langsung dari proses segmentasi oleh radiolog. Fitur yang diperlukan selama proses *training* model terdiri dari citra *slice* dan segmentasi kalsium untuk model *CNN*. Data pasien terlebih dahulu diseleksi untuk mencegah data-data yang salah atau rusak masuk ke model. Data pasien dengan nilai *CAC* dibagi menjadi data pasien untuk *training*, *validation*, dan *testing* dengan perbandingan 70%:20%:10%. Proses pembagian dilakukan untuk masing-masing pasien dengan *CAC* dan tanpa *CAC* untuk memastikan masing-masing pembagian data merepresentasikan pasien dengan *CAC* dan tanpa *CAC*. Untuk menanggulangi ketidakseimbangan antara jumlah citra dengan *CAC* dan tanpa *CAC* pada data *training*, dilakukan *random undersampling* dimana citra tanpa *CAC* dipilih secara acak agar memiliki jumlah yang sama dengan citra dengan *CAC*. Proses *preprocessing* akan menghasilkan dua data, yaitu data citra, data segmentasi yang akan digunakan oleh model *CNN*. Untuk mempercepat dan memudahkan model *CNN* untuk mengolah data, data citra h5 dan data segmentasi json dikonversi terlebih dahulu menjadi *trecord*, sejenis *file* berbasis *proto buffer* yang berfungsi untuk menserialkan data sehingga beban masukan dan keluaran data oleh model *CNN* berkurang [16]. Keseluruhan proses *preprocessing* dirangkum pada blok diagram yang tertera pada Gambar 3.2.

#### 3.3.1. Preprocessing Data Segmentasi

Untuk mengubah data mentah segmentasi menjadi data yang dapat digunakan model, diperlukan 3 proses proses, proses tersebut meliputi konversi data *hashmap xml* menjadi *json*, rasterisasi data koordinat lesi untuk menghasilkan *segmentation mask*, dan konversi data *segmentation mask* menjadi *sparse array*.



Gambar 3.2 Diagram Blok *Preprocessing*

Proses pengonversian data xml mentah menjadi *json* terbagi atas dua tahap, yaitu tahap pengonversian awal data mentah xml langsung menjadi struktur data *json* dan tahap pembersihan sekaligus pemilahan data penting yang diperlukan untuk tahap selanjutnya. Pada tahap pertama, pengonversian data langsung dilakukan tanpa melakukan konversi khusus pada masing-masing fitur yang ada pada data mentah. Pada tahap kedua, dilakukan pemilahan fitur dengan mengambil data indeks pasien, indeks citra, jenis lesi serta lokasi lesi dalam satuan piksel. Terdapat kesalahan yang tertera pada data mentah COCA dimana indeks citra pada data segmentasi memiliki urutan terbalik jika dibandingkan pada indeks citra pada pasien, sehingga diperlukan tambahan penyesuaian agar indeks citra selaras.

Proses rasterisasi dilakukan pada koordinat keliling lesi dalam sumbu x dan y. Proses rasterisasi ini dilakukan untuk melakukan diskritisasi lokasi lesi agar tepat berada satu lokasi piksel, mengonversi data keliling menjadi data luasan, dan mengonversi data koordinat x dan y menjadi baris dan kolom untuk mempermudah penggunaannya nanti dalam program. Algoritma proses rasterisasi pertama yang dikembangkan merupakan algoritma Bresenham pada tahun 1965. Algoritma Bresenham mengonversi koordinat pembentuk garis ke koordinat diskrit yang merepresentasikan garis tersebut. Algoritma Bresenham memanfaatkan persamaan garis dengan gradien untuk menentukan lokasi piksel terdekat dalam nilai diskrit ke garis. Algoritma Bresenham menentukan kapan harus menaikkan nilai y pada koordinat diskrit dengan membandingkan polaritas dari hasil kalkulasi tanpa perlu melihat magnitudo. Algoritma ini juga dioptimasi kembali menggunakan *incremental computation* dimana hasil komputasi algoritma untuk iterasi  $n + 1$  tidak memerlukan komputasi gradien ulang, melainkan melanjutkan nilai kalkulasi pada iterasi ke  $n$ . Optimasi tersebut memungkinkan proses rasterisasi dilakukan tanpa menggunakan operasi perkalian dan pembagian [17]. Algoritma Bresenham dapat direpresentasikan pada Persamaan 3.1 sampai Persamaan 3.5.



$$\Delta x = x_1 - x_0 \quad (3.1)$$

$$\Delta y = y_1 - y_0 \quad (3.2)$$

$$D_{n+1} = \begin{cases} D_n + 2\Delta y, & D_n < 0 \\ D_n + 2(\Delta y - \Delta x), & D_n \geq 0 \end{cases} \quad (3.3)$$

$$x_{n+1} = x_n + 1 \quad (3.4)$$

$$y_{n+1} = \begin{cases} y_n, & D_n < 0 \\ y_n + 1, & D_n \geq 0 \end{cases} \quad (3.5)$$

dengan  $x_0$  dan  $y_0$  adalah koordinat titik awal dari garis dan  $x_1$  dan  $y_1$  adalah koordinat titik akhir dari garis. Persamaan 3.1 sampai 3.5 merepresentasikan algoritma yang digunakan pada konteks garis dengan gradien positif dengan sudut 0 sampai 45°. Representasi algoritma pada gradien dan rentang sudut yang berbeda tetap menerapkan pendekatan yang sama dengan sedikit modifikasi. Untuk mempercepat proses rasterisasi, penelitian ini menggunakan *library geo-rasterize*. *Library* tersebut merupakan implementasi ulang algoritma rasterisasi GDAL (*Geospatial Data Abstraction Library*) dalam bahasa Rust. *Library* ini menerapkan algoritma Bresenham dan mengoptimasinya pada bahasa pemrograman terkompilasi, dalam hal ini Rust, sehingga proses rasterisasi akan lebih cepat dibandingkan implementasi ulang pada bahasa Python. Hasil rasterisasi ini akan menghasilkan data *segmentation mask* dua dimensi yang akan diproses lebih lanjut.

Proses terakhir merupakan proses konversi data *segmentation mask* menjadi data *sparse array*, yaitu format data yang menyimpan koordinat *region of interest*. Proses ini dilakukan karena format data *segmentation mask* memiliki banyak nilai 0 sehingga penggunaan penyimpanan *file* dapat berkurang dengan hanya menyimpan lokasi dari piksel-piksel dari lesi kalsium.

Bentuk *segmentation mask* yang digunakan oleh model sebagai *ground truth* memiliki dimensi 512x512x1 untuk menyesuaikan keluaran dari model yang menggunakan fungsi aktivasi *sigmoid*. *Segmentation mask* yang digunakan berisikan probabilitas ada tidaknya kalsium dengan kelas negatif sebagai *null* (bukan pembuluh darah) dan kelas positif merepresentasikan kalsium yang terletak pada keempat lokasi pembuluh darah yang berpengaruh pada nilai *Agatston* yaitu LCA (*Left Coronary Artery*), LCX (*Left Circumflex Artery*), LAD (*Left Anterior Descending Artery*), dan RCA (*Right Coronary Artery*).

*Segmentation mask* yang digunakan oleh Eng et al terdiri 2 jenis segmentasi dengan dimensi 512x512x4 (*multiclass*) dan 512x512x1 (*binary*) [6]. Modifikasi ini dilakukan untuk menyederhanakan keluaran dari model ketika menggunakan mode *deep supervision*. *Deep supervision* memerlukan keluaran dari tiap kedalaman *layer*, sehingga jika model memiliki keluaran *binary* dan *multiclass*, keluaran dari model akan menjadi dua kali lipat dari kedalaman model. Proses *preprocessing* segmentasi akan menghasilkan suatu file json yang memiliki informasi lesi untuk tiap pasien beserta indeks gambar, koordinat dari lesi kalsium.

### 3.3.2. Preprocessing Citra

Proses *preprocessing* citra dilakukan pada dua waktu yang berbeda yaitu proses *preprocessing* sebelum dan ketika *training* model. Proses *preprocessing* sebelum *training* adalah proses ekstraksi data *Hounsfield Unit* dari citra. Proses *preprocessing* ketika *training* meliputi proses yang paling penting dari proses *preprocessing* citra yaitu proses *clipping*, normalisasi dan *zero-centering*.

Proses ekstraksi data *Hounsfield Unit* didasari oleh Persamaan 3.6 yang tertera pada standar DICOM PS3.3 terkait *Modality LUT Module*

$$\text{Output units} = m \times SV + b \quad (3.6)$$

dimana  $m$  adalah *Rescale Slope* (0028, 1053),  $SV$  adalah *stored value* atau data yang tersimpan pada DICOM, dan  $b$  adalah *Rescale Intercept* (0028, 1052) [18]. Jenis dari keluaran Persamaan 3.2 menjadi *Hounsfield Unit* jika *Image Type* (0008, 0008) bernilai “ORIGINAL” tanpa “LOCALIZER” dan *Multi-energy CT Acquisition* (0018, 9361) bernilai “NO”. Jika kondisi tersebut tidak sesuai maka tipe keluaran bergantung pada *Rescale Type* (0028, 1054) [18].

Seluruh citra dan data citra seperti ketebalan *slice* dan besar pixel dalam milimeter disimpan dalam bentuk h5 untuk mempermudah akses data dan mengoptimasi penggunaan penyimpanan pada komputer.

### 3.4. Model CNN

Pada penelitian ini model CNN yang dikembangkan memiliki arsitektur dasar UNet++ dengan *encoder* yang berisikan *bottleneck inverted residual layer* dari MobileNetV2. *Bottleneck inverted residual layer* berisikan *depthwise convolution* sebagai pengganti *layer* konvolusi tradisional untuk mengurangi jumlah komputasi [9]. Model UNet++ biasa dengan parameter yang mengikuti penelitian Zhou et al juga *ditrain* sebagai perbandingan dengan model yang diajukan. Visualisasi arsitektur model CNN dapat dilihat pada Gambar 3.3, informasi dari tiap *layer* tercantum pada Tabel 3.1, dan pemetaan hubungan antar *node* dapat direpresentasikan pada Persamaan 3.7.

Pemetaan fitur pada struktur model CNN yang digunakan didasari oleh persamaan UNet++ pada Persamaan 3.7.

$$x^{i,j} = \begin{cases} H(D(x^{i-1,j})), & j = 0 \\ H\left(\left[[x^{i,k}]_{k=0}^{j-1}, U(x^{i+1,j-1})\right]\right), & j > 0 \end{cases} \quad (1)$$

dengan  $x$  merupakan *node* dari model,  $i$  merupakan indeks dari *node* secara vertikal,  $j$  merupakan indeks dari *node* secara horizontal,  $H$  adalah operasi konvolusi dengan fungsi aktivasi,  $D$  adalah operasi *downsampling*,  $U$  adalah operasi *upsampling* dan  $[$  adalah operasi konkatenasi [12]. Operasi  $H$  dilakukan oleh blok yang berisikan dua konvolusi dua dimensi dengan setiap operasi  $H$  diberikan *batch normalization* dan *dropout* dengan probabilitas 0.2 untuk mengatasi *overfitting*. Penggunaan konvolusi dua dimensi biasa pada operasi  $H$  yang merupakan dasaran dari semua blok dari model dikarenakan dari batasan yang dimiliki oleh *depthwise convolution*,

operasi pembentuk *inverse residual block*. *Depthwise convolution* pada pengaplikasiannya pada tingkat *hardware* memiliki intensitas aritmetika yang rendah dimana operasi ini melakukan lebih sedikit operasi aritmetika dibandingkan operasi memori sehingga operasi tersebut memakan waktu lebih banyak melakukan *read* dan *write* ke memori dibandingkan proses operasi aritmetik [19]. Walau demikian, *depthwise convolution* memiliki penggunaan memori yang lebih rendah dibandingkan dengan konvolusi biasa sehingga penggunaannya pada *encoder* masih dapat dilakukan tanpa harus sepenuhnya mengorbankan kecepatan model dalam proses inferensi. *Depthwise convolution* dapat menjadi lebih cepat jika menggunakan *batch* yang besar, akan tetapi dalam pengaplikasiannya pada penelitian ini penggunaan *batch* yang besar tidak mungkin dilakukan karena ukuran dimensi citra yang sudah cukup besar. Operasi *D* dilakukan oleh bagian *downsampling inverse residual block* pada MobileNetV2, dimana *inverse residual block* memiliki parameter *stride* yang dapat digunakan untuk *downsampling*. Untuk melakukan *upsampling*, model CNN ini menggunakan *Conv2DTranspose* atau menggunakan konvolusi untuk memberikan tambahan cara bagi model untuk mempelajari data yang diterima.

Model memiliki kedalaman lima lapisan secara vertikal atau empat tingkatan *pruning*. Parameter jumlah filter pada blok H mengikuti pola 16, 32, 64, 128, 256 berdasarkan kedalaman *i* atau *layer* secara vertikal. Pola tersebut merupakan pola dasar yang juga ditemukan pada model sebelumnya seperti UNet dimana jumlah filter akan semakin besar dengan semakin dalamnya lapisan. Pola tersebut muncul karena ide dasar dari *encoder* adalah melakukan penurunan dimensi masukan menggunakan proses *downsampling* dan diikuti dengan peningkatan kedalaman atau *channel* melalui peningkatan filter.

Parameter *pretrained inverse residual block* MobileNetV2 yang digunakan pada bagian enkoder mengikuti jumlah yang tertera pada penelitian awal oleh Sandler et al. Untuk mengimbangi jumlah *downsampling* pada UNet++, lokasi pengambilan lokasi *node* untuk dijadikan *skip connection* dilakukan pada *node input model*, *node bottleneck 1*, *node bottleneck 2*, *node bottleneck 3*, dan *node bottleneck 5* pada ujung enkoder. Blok yang digunakan pada *encoder* diambil pada *inverted residual block* MobileNetV2 yang telah melalui proses *training* sebelumnya pada data *ImageNet*. Penggunaan *pretrained* model pada *encoder* dilakukan untuk mempercepat proses konvergensi dari model dibandingkan memulai proses *training* dari nol. MobileNetV2 memiliki penelitian lanjutan yaitu MobileNetV3 yang memiliki pembaharuan seperti *self-attention* dan *h-swish activation function* [20], akan tetapi pembaharuan ini hanya terletak pada *layer* akhir sehingga penggunaannya dalam *encoder* yang memerlukan *layer* yang lebih awal tidak memaksimalkan penggunaan pembaharuan tersebut. *Encoder pretrained* MobileNetV2 memiliki 3 *channel*, sedangkan masukan model memiliki 1 *channel*. Untuk mengatasi hal tersebut diberikan operasi konkatenasi yang menduplikasi masukan model dari 1 *channel* menjadi 3 *channels*. Model ini memiliki total

parameter 4.143.972 parameter dengan parameter yang dapat *ditrain* sebanyak 4.122.276 parameter dan parameter yang tidak dapat *ditrain* sebanyak 21.696 parameter.

Parameter dan blok yang digunakan pada model ini dijelaskan lebih detail pada Tabel 3.2 dimana pada kolom tipe *block*, *IRB* adalah *inverse residual block*, *C2T* adalah *Conv2dTranspose*, dan *C2* adalah konvolusi dua dimensi. Pada kolom parameter, *f* adalah jumlah filter pada blok, *t* adalah parameter faktor ekspansi yang tersedia pada *inverse residual block*, *n* adalah parameter iterasi yang tersedia pada *inverse residual block*, *s* adalah *stride* yang digunakan pada konvolusi dan *k* adalah besar *kernel* yang digunakan pada konvolusi.

Fungsi *loss* yang digunakan merupakan fungsi *Log-cosh Dice loss*, pemilihan fungsi *loss* ini didasari oleh kemampuannya yang baik pada data yang tidak seimbang dibandingkan *dice loss* biasa, sehingga sesuai dengan ketidakseimbangan yang dihadapi pada data COCA. Fungsi *Log-cosh Dice loss* dapat dilihat pada Persamaan 3.8.

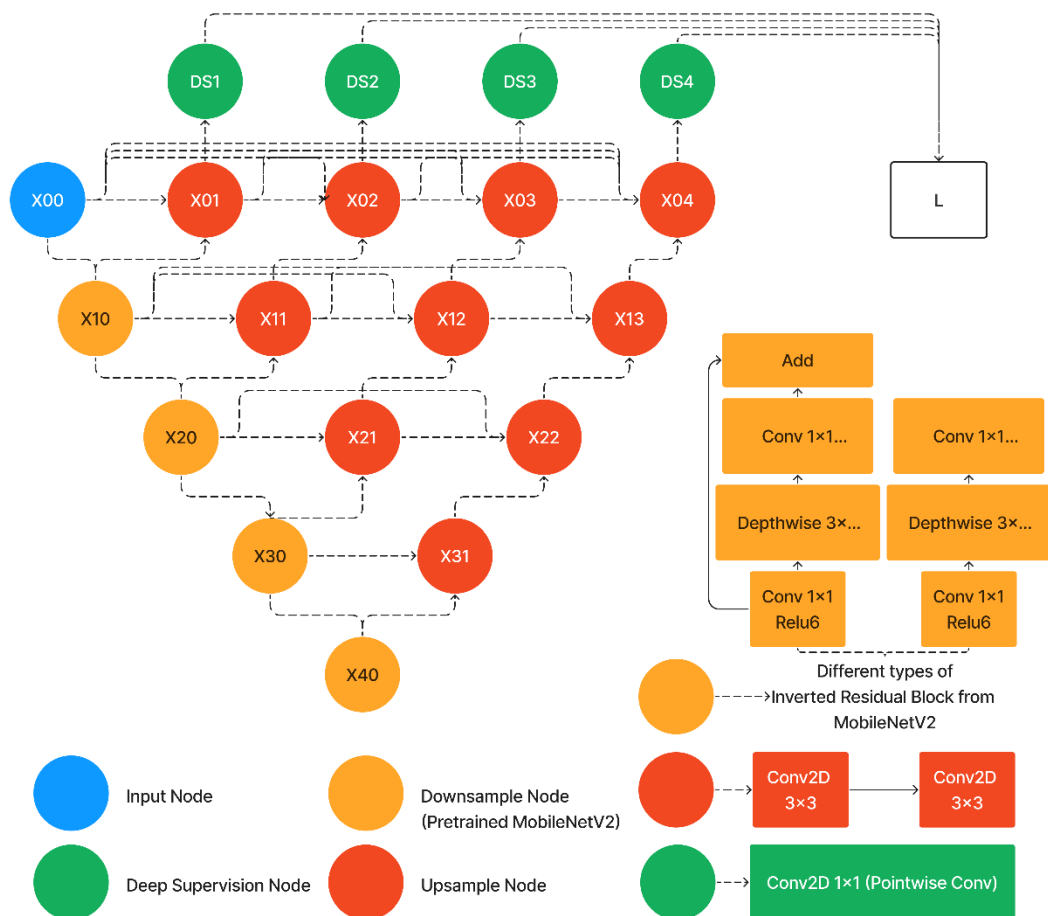
$$\text{Log - Cosh Dice Loss} = \log \left( \cosh \left( 1 - \frac{2\hat{y}y + \epsilon}{y^2 + \hat{y}^2 + \epsilon} \right) \right) \quad (3.8)$$

dimana *y* adalah target prediksi,  $\hat{y}$  adalah hasil prediksi, dan  $\epsilon$  merupakan koefisien untuk mencegah terjadinya pembagian terhadap nol [21]. Terdapat kustomisasi pada rumus *dice coefficient* yang digunakan, yaitu melalui pemberian kuadrat pada penyebut. Pemberian kuadrat ini dilakukan untuk mempercepat konvergensi model berdasarkan penelitian yang dilakukan oleh Milletari et al [22]. Fungsi optimasi yang digunakan merupakan fungsi optimasi Adam dengan *learning rate* 0,001.

Proses *training* dari model dilakukan dalam mode *mixed precision* yang dikembangkan oleh Narang et al. Mode ini memungkinkan pengurangan penggunaan memori pada model melalui penyimpanan *weight*, aktivasi dan gradien dalam *half precision floating point* [23]. Biasanya model *machine learning* menggunakan *single precision floating point* yang menggunakan 32bit untuk menyimpan data *floating point*, sehingga penggunaan *half precision floating point* yang menggunakan 16 bit dapat mengurangi penggunaan memori. Pengurangan jumlah bit untuk merepresentasikan nilai-nilai pembentuk model juga mengurangi tingkat kepresisian dari kalkulasi model. *Single precision floating point* memiliki 24 bit untuk menyimpan kepresisian nilai *floating point* sedangkan *half precision floating point* hanya memiliki 11 bit untuk menyimpan kepresisian nilai *floating point* [24]. Untuk mengatasi penurunan kepresisian ini, Narang et al mengusulkan tiga metode saat melakukan *training* dalam mode *mixed precision*. Metode pertama adalah penyimpanan duplikat parameter *weight* dalam *single precision* selama proses pengumpulan gradien pada tahap optimasi parameter. Parameter *weight* ini kemudian dibulatkan ke tingkat *half precision* ketika melalui proses *forward* dan *backward propagation*. Penyimpanan duplikat ini akan meningkatkan penggunaan memori model sebanyak 50% pada parameter *weight* akan tetapi karena mayoritas memori digunakan oleh fungsi aktivasi yang tetap berada pada *half precision*,

penggunaan memori saat *training* akan tetap berkurang 50%. Metode kedua adalah *loss scaling* yang digunakan untuk mempertahankan gradien dengan nilai kecil yang memerlukan presisi yang lebih tinggi. Proses *loss scaling* terdiri dari multiplikasi *loss* pada proses *forward propagation* dengan suatu faktor skala, dalam hal ini biasanya memiliki nilai dengan kelipatan kedua. Pada proses *backward propagation* gradien dari fungsi *loss* dimultiplikasi dengan invers faktor skala yang digunakan pada *forward propagation*. Metode ketiga adalah melakukan *forward propagation* menggunakan *half precision*, mengakumulasi hasil kalkulasi dalam *single precision*, dan menyimpan hasil kalkulasi ke memori dalam *half precision*.

Model CNN akan dilatih dalam mode *deep supervision*. Mode ini secara langsung melakukan *training* untuk keluaran tiap *layer training* untuk memacu *ensemble learning*. Penelitian mengenai otomasi skoring arteri kalsium sebelumnya yang menggunakan UNet++ oleh Hong et al tidak memiliki pembahasan mengenai *deep supervision*. Mode *deep supervision* memungkinkan model CNN untuk melalui proses *pruning*, yaitu proses pelepasan *layer* secara berurutan untuk mengurangi jumlah *layer* [12]. Model menggunakan fungsi aktivasi *sigmoid* pada keluaran *deep supervision* yang berisikan konvolusi dengan kernel 1x1.



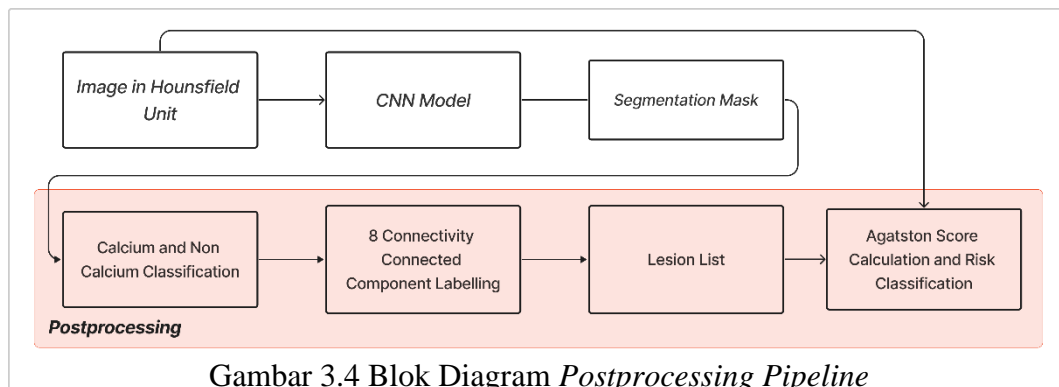
Gambar 3.3 Arsitektur UNet++ [12] dengan *Pretrained MobileNetV2 Encoder* [9]

Tabel 3.1 Arsitektur UNet++ dengan *Pretrained MobileNetV2 Encoder*

Layer	Node	Type Node	Output	Type block	Parameter				
					f	t	n	s	k
0	X00	Input	512x512x1	Input	-	-	-	-	-
			512x512x3	Concatenate	-	-	-	-	-
1	X10	Downsample	256x256x32	C2	32	-	-	2	-
			256x256x16	IRB	16	1	1	1	-
			256x256x32	C2	32	-	2	1	-
	X01	Upsample	512x512x16	C2T	16	-	-	2	2
			512x512x16	C2	16	-	2	1	-
DS1	Deep Supervision	512x512x1	C2	1	-	-	1	1	
2	X20	Downsample	128x128x24	IRB	24	6	2	2	-
			128x128x64	C2	64	-	2	1	-
	X11	Upsample	256x256x32	C2T	32	-	-	2	2
			256x256x32	C2	32	-	2	1	-
	X02	Upsample	512x512x16	C2T	16	-	-	2	2
			512x512x16	C2	16	-	2	1	-
	DS2	Deep Supervision	512x512x1	C2	1	-	-	1	1
3	X30	Downsample	64x64x32	IRB	32	6	3	2	-
			64x64x128	C2	128	6	2	1	-
	X21	Upsample	128x128x64	C2T	64	-	-	2	2
			128x128x64	C2	64	-	2	1	-
	X12	Upsample	256x256x32	C2T	32	-	-	2	2
			256x256x32	C2	32	-	2	1	-
	X03	Upsample	512x512x16	C2T	16	-	-	2	2
			512x512x16	C2	16	-	2	1	-
DS3	Deep Supervision	512x512x1	C2	1	-	-	1	1	
4	X40	Downsample	32x32x64	IRB	64	6	4	2	-
			32x32x96	IRB	96	6	3	1	-
			32x32x256	C2	256	-	2	1	-
	X31	Upsample	64x64x32	C2T	32	-	-	2	2
			64x64x32	C2	32	-	2	1	-
	X22	Upsample	128x128x64	C2T	64	-	-	2	2
			128x128x64	C2	64	-	2	1	-
	X13	Upsample	256x256x32	C2T	32	-	-	2	2
			256x256x32	C2	32	-	2	1	-
	X04	Upsample	512x512x16	C2T	16	-	-	2	2
			512x512x16	C2	16	-	2	1	-
DS4	Deep Supervision	512x512x1	C2	1	-	-	1	1	

### 3.5. Postprocessing Data

Proses algoritma *postprocessing* pada penelitian ini mengacu pada metode *postprocessing* yang digunakan oleh Eng et al [6]. Terdapat tiga tahapan dalam proses *postprocessing* ini. Tahap pertama adalah proses klasifikasi ada tidaknya kalsium untuk tiap piksel dari hasil keluaran model melalui pengaplikasian *threshold*  $> 0,5$ . Tahap kedua adalah proses *8-connectivity connected component labelling* dimana hasil segmentasi akan dikelompokkan menjadi lesi-lesi atau luasan. Tahap terakhir, proses kalkulasi nilai *Agatston* (per citra) dapat mulai dikalkulasikan menggunakan citra masukan awal dalam bentuk *Hounsfield Unit*, dan kelompok lesi-lesi yang didapatkan melalui tahap nilai *Agatston* tiap *slice* citra dikalkulasi menggunakan citra masukan yang sudah dikonversi dalam satuan *Hounsfield Unit* dan lokasi lesi kalsium. Diagram blok keseluruhan proses *postprocessing* tertera pada Gambar 3.4.



Gambar 3.4 Blok Diagram *Postprocessing Pipeline*

#### 3.5.1. Connected Component Labelling

*Connected component labelling* (CCL) digunakan untuk mengelompokkan hasil prediksi dengan kalsium menjadi kumpulan lesi-lesi. Pengelompokan ini dilakukan karena metode penilaian *Agatston* memerlukan luasan dari lesi pada proses kalkulasinya. Mode CCL yang digunakan merupakan mode *8-connected* dimana algoritma mengelompokkan piksel berdasarkan kelompok pada 8 piksel di sekitar piksel utama. Salah satu algoritma CCL pertama yang dikembangkan merupakan algoritma yang dikembangkan oleh Rosenfeld et al pada tahun 1966. Algoritma Rosenfeld et al terdiri dari dua tahap, pada tahap pertama dilakukan iterasi untuk tiap piksel pada citra. Untuk tiap piksel *foreground*, algoritma melakukan pemeriksaan pada piksel di sekitar piksel *foreground*. Jika semua piksel disekitar piksel utama bernilai nol, maka piksel tersebut diberikan label baru. Jika terdapat piksel dengan label di sekitar piksel utama, piksel tersebut diberikan label dengan nilai minimum. Selama melakukan iterasi pada citra di tahap pertama, algoritma menyimpan hubungan antar label yang berada pada lokasi yang sama. Pada tahap kedua, dengan menggunakan hubungan antar label dengan lokasi yang berdekatan dari tahap pertama, tiap pengelompokan piksel akan diberikan label

dengan nilai minimum dan tidak akan ada lagi pengelompokan piksel dengan nilai label yang berbeda [25].

Sejak tahun 1966, metode CCL telah mengalami optimasi dan pengembangan untuk mengurangi penggunaan memori dan mempercepat proses CCL. Untuk memastikan sistem menggunakan metode tercepat dalam implementasi CCL, sistem yang dikembangkan dengan bahasa pemrograman Python menggunakan *library* OpenCV. Pengimplementasian algoritma CCL yang digunakan oleh OpenCV adalah algoritma *Spagheti* oleh Bolelli et al. Algoritma ini melakukan operasi CCL melalui konversi struktur data citra yang awalnya dua dimensi menjadi struktur data *Directed Rooted Acyclic Graph* (DRAG). DRAG yang dihasilkan digunakan untuk menggenerasi kode *if else* dari representasi *graph*. Proses *labelling* dimulai dari fitur terkuat pada *graph* yang digenerasi dan dari fitur tersebut algoritma perlahan menyebar berdasarkan kriteria yang ditentukan untuk menemukan pasangan lesi yang sesuai [26].

### 3.5.2. Skoring Agatston

Nilai *Agatston* tidak diberikan dalam *dataset CT-Scan gated* melainkan pada *dataset CT-Scan non-gated*. Namun, pasien pada *dataset CT-Scan non-gated* tidak sama dengan pasien yang ada pada *dataset CT-Scan gated*. Untuk mendapatkan nilai tersebut diperlukan proses penghitungan menggunakan data citra dan data *segmentation mask*. Proses perhitungan dilakukan menggunakan Persamaan 3.9

$$\text{Agatston Score} = W \times \text{Pixels} \times \text{Pixel Size} \quad (3.9)$$

dimana  $W$  merupakan *weighting factor* yang didasari oleh atenuasi maksimal pada piksel citra di atas  $>130$  *Hounsfield Unit* seperti yang dijelaskan pada Tabel 3.2, *Pixels* merupakan jumlah piksel yang teridentifikasi sebagai kalsium pada pembuluh koroner, dan *Pixel Size* adalah besar 1 piksel dalam satuan  $\text{mm}^2$  [7]. Pada sistem untuk mencegah *noise* seperti halnya yang dilakukan Hong et al, lesi dengan area lebih kecil dari  $1 \text{ mm}^2$  tidak dihitung. Hasil dari nilai *Agatston* kemudian dapat digunakan untuk mengelompokkan pasien berdasarkan tingkat keparahan kalsifikasi arteri koroner seperti yang tertera pada Tabel 3.3.

Tabel 3.2 Tabel Konversi *Weighting Factor* terhadap *Hounsfield Unit* [7]

W	HU ( <i>Hounsfield Unit</i> )
1	130-199
2	200-299
3	300-399
4	$\geq 400$

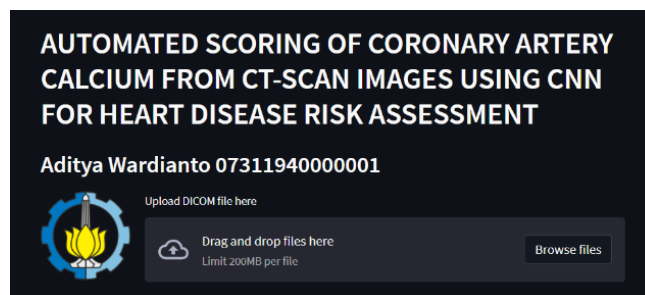


Tabel 3.3 Tingkat Keparahan Kalsifikasi Arteri Koroner [27]

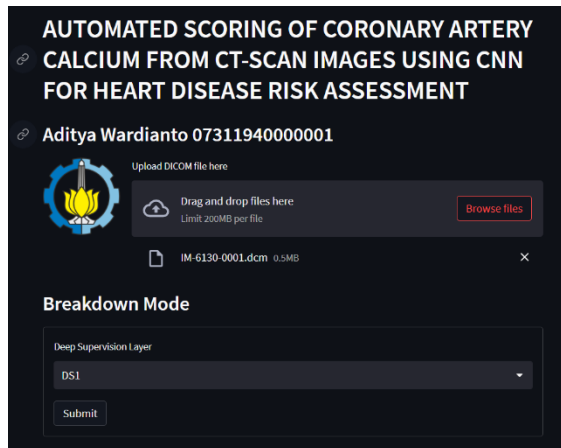
Risiko	Nilai Agatston	Korelasi Klinis
<i>Absent</i>	0	Risiko rendah penyakit jantung di masa depan
<i>Minimal</i>	1-10	Jumlah aterosklerosis rendah, risiko rendah penyakit jantung di masa depan
<i>Mild</i>	11-100	Jumlah aterosklerosis ringan, risiko ringan penyakit jantung di masa depan
<i>Moderate</i>	101-400	Jumlah aterosklerosis sedang, risiko sedang penyakit jantung di masa depan
<i>High</i>	>400	Jumlah aterosklerosis tinggi, risiko tinggi penyakit jantung di masa depan

### 3.6. Graphical User Interface

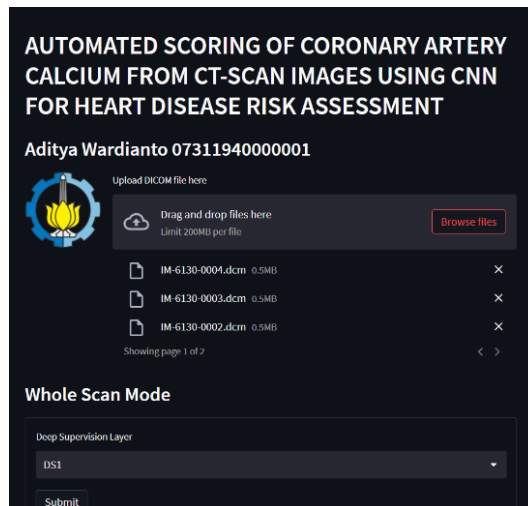
Program GUI dibuat menggunakan *library* Python bernama Streamlit. GUI ini berbentuk *web app* yang dapat *deploy* pada jaringan lokal atau melalui jaringan internet. Pada penelitian ini GUI dikembangkan untuk mempermudah penggunaan model secara umum khususnya untuk mempermudah untuk melakukan *pruning* pada model yang sudah melalui proses *training*. GUI memiliki dua mode yaitu mode *breakdown* dan *whole scan* yang secara otomatis berubah berdasarkan jumlah *file* DICOM yang diunggah. Ketika pengguna mengunggah satu *file* DICOM, GUI akan berada pada mode *breakdown* dimana setiap proses yang dilalui oleh citra hingga menjadi segmentasi biner akan diperlihatkan melalui gambar pada GUI. Ketika pengguna mengunggah lebih dari satu *file* DICOM, GUI akan berubah menjadi *mode whole scan*. Mode ini bekerja dengan melakukan skoring CAC pada seluruh *file* citra yang diberikan tanpa memberikan gambar dari proses untuk masing-masing citra, melainkan nilai total CAC untuk keseluruhan citra. Mode ini dibuat sedemikian rupa untuk alasan performa, dimana untuk satu pasien, terdapat kurang lebih 50 citra yang perlu melalui proses skoring CAC. Jika seluruh gambar citra ditampilkan secara bersamaan, program GUI akan lebih banyak menggunakan kemampuan komputasi komputer untuk menampilkan gambar, bukan dalam inferensi model CNN. Tampilan GUI dapat diamati pada Gambar 3.5 hingga Gambar 3.8.



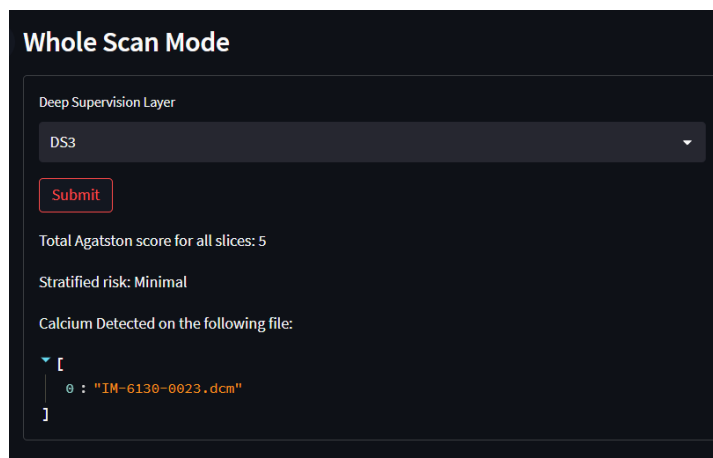
Gambar 3.5 Tampilan Depan GUI



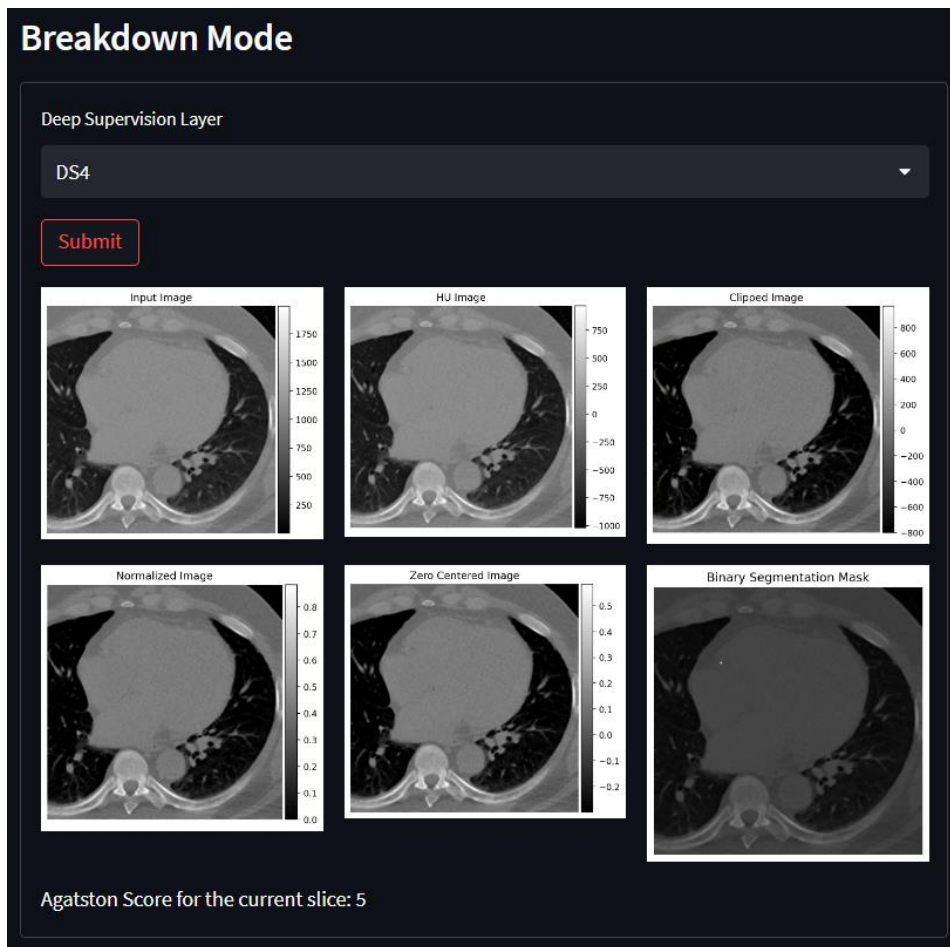
Gambar 3.6 Tampilan GUI Pada Mode *Breakdown*



Gambar 3.7 Tampilan GUI Pada Mode *Whole Scan*



Gambar 3.8 Tampilan Hasil Skoring Pada Mode *Whole Scan*



Gambar 3.9 Tampilan GUI Dengan Hasil Skoring Pada Mode *Breakdown*

### 3.7. Uji Coba Sistem

Pengujian sistem dibagi menjadi 3 tahap, pengujian hasil *preprocessing*, model, dan sistem. Pengujian proses *preprocessing* meliputi pengujian pada proses seleksi data, *random undersampling*, rasterisasi pada segmentasi dan *preprocessing* pada citra.

Pengujian model dilakukan pada tiap tingkatan *pruning* model melalui kalkulasi metrik performa *dice coefficient*, *precision* dan *recall* serta melihat perbandingan prediksi dari model terhadap *ground truth*. Persamaan *dice coefficient* dapat dilihat pada Persamaan 3.10

$$Dice\ Coefficient = \frac{2\hat{y}y + \epsilon}{y + \hat{y} + \epsilon} \quad (3.10)$$

dimana  $y$  adalah target prediksi,  $\hat{y}$  adalah hasil prediksi, dan  $\epsilon$  merupakan koefisien untuk mencegah terjadinya pembagian terhadap nol saat kalkulasi. Persamaan *precision* dan *recall* dapat dilihat pada Persamaan 3.11 dan Persamaan 3.12

$$Precision = \frac{TP}{TP + FP} \quad (3.11)$$

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

dengan  $TP$  adalah jumlah hasil prediksi *true positive*,  $FP$  adalah jumlah hasil prediksi *false positive*  $FN$  adalah jumlah hasil prediksi *false negative*.

Pengujian sistem dilakukan pada tiap tingkatan *pruning* model dengan melihat nilai *Agatston* dan label klasifikasi dari pasien. Untuk nilai *Agatston* metrik yang digunakan merupakan *mean absolute error* yang dirumuskan melalui Persamaan 3.11

$$MAE = \frac{\sum_{i=1}^N |y - \hat{y}|}{N} \quad (3.11)$$

dengan  $y$  adalah target prediksi,  $\hat{y}$  adalah hasil prediksi dan  $N$  adalah jumlah data.

Untuk label klasifikasi tingkat keparahan kalsium arteri coroner, metrik yang digunakan berupa *Cohen's Kappa*, kecepatan rata-rata inferensi per pasien serta *confusion matrix* dari hasil prediksi. Persamaan *Cohen's Kappa* dirumuskan pada Persamaan 3.12

$$k = \frac{p_o - p_e}{1 - p_e} \quad (3.12)$$

dengan  $p_o$  adalah persentase jumlah pengukuran yang konsisten antar *rater* dan  $p_e$  adalah persentase jumlah perubahan pengukuran antar *rater*.

## BAB IV PENGUJIAN SISTEM

### 4.1. Pengujian *Preprocessing*

Pengujian yang dilakukan pada proses *processing* dilakukan secara berurutan mengikuti alur dari proses *preprocessing* mulai dari seleksi data, *preprocessing* segmentasi, *preprocessing* citra, dan *random undersampling* data.

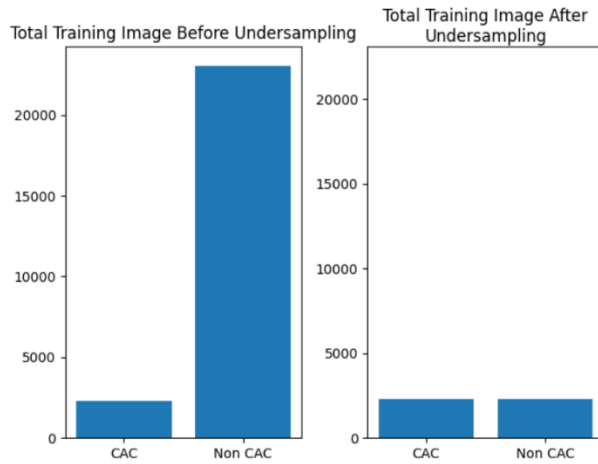
#### 4.1.1. Pengujian Seleksi Data dan *Random Undersampling*

Data CT-Scan *gated* pasien yang tersedia dari data COCA terdiri dari 789 pasien dengan 451 pasien memiliki nilai CAC dan 338 pasien dengan nilai CAC 0. Untuk masing-masing pasien terdapat 50 irisan citra CT-Scan sebesar 512x512 dengan total keseluruhan citra dari tiap pasien sebanyak 36.411 citra. Dari seluruh citra pasien terdapat 3.656 citra dengan kalsifikasi kalsium atau meliputi sekitar 0.04% citra yang tersedia.

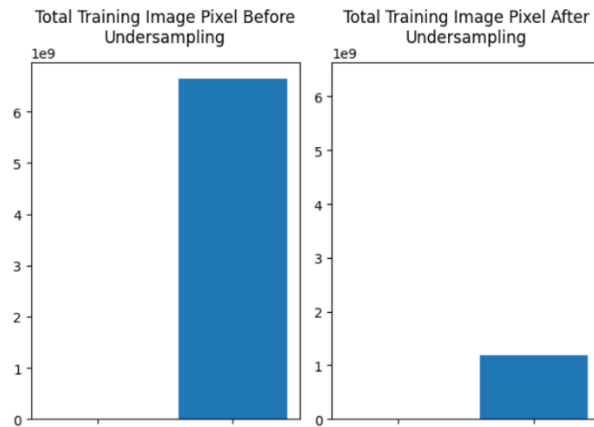
Proses seleksi data pasien dilakukan selama proses konversi data lesi dan data citra. Ketika terjadi *error*, data pasien akan diperiksa dan jika ditemukan ada kesalahan, data pasien tersebut tidak digunakan. Dari proses seleksi ini, terdapat 30 pasien dengan lesi yang tumpang tindih setelah proses rasterisasi, 2 pasien dengan lesi yang tidak dapat digunakan, 14 pasien yang memiliki lebih dari satu *scan* citra, 4 pasien tanpa citra di DICOM dan 1 pasien dengan indeks segmentasi yang tidak sesuai. Proses seleksi mengurangi data pasien sejumlah 51 pasien menyisakan 738 pasien untuk melanjutkan proses *preprocessing*. Dari 738 pasien tersebut, 404 merupakan pasien dengan CAC dan 334 pasien tidak memiliki CAC.

Pembagian data yang digunakan oleh model mengikuti distribusi 70%:20%:10% dengan proses pembagian dilakukan untuk masing-masing pasien dengan CAC dan tanpa CAC. Pembagian tersebut dilakukan untuk mendapatkan distribusi pasien yang sama antar data *training*, validasi, dan *testing*. Data *training* terdiri dari 510 pasien dengan total 2.277 citra (31.7491 piksel) dengan CAC dan 23.065 citra (6.642.935.757 piksel) tanpa CAC sebelum mengalami *random undersampling*. Setelah *random undersampling*, jumlah citra dengan CAC dan tanpa CAC pada data *training* sama yaitu sebanyak 2277 citra untuk masing-masing jenis pasien (1.193.486.285 piksel untuk citra tanpa CAC). Grafik perbandingan pengurangan data *training* sebelum dan sesudah *undersampling* dapat dilihat pada Gambar 4.2 dan Gambar 4.1.

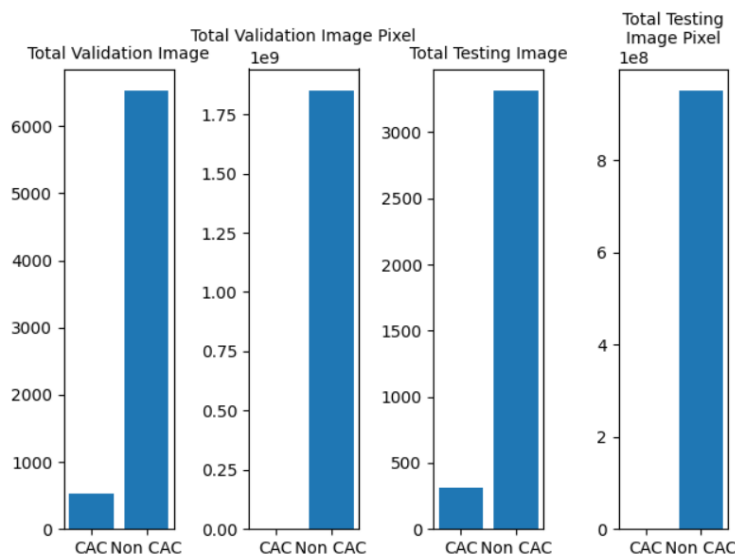
Data validasi merupakan 20% dari keseluruhan data yang terdiri dari 146 pasien dengan komposisi 81 pasien dengan CAC dan 65 pasien tanpa CAC. Data validasi memiliki 528 citra dengan CAC dan 6525 citra tanpa CAC. Data *testing* merupakan 10% dari keseluruhan data dengan total 74 pasien dengan komposisi 41 pasien dengan CAC dan 33 pasien tanpa CAC. Data *testing* memiliki 341 citra tanpa CAC dan 3460 citra tanpa CAC. Grafik distribusi irisan citra dan jumlah piksel pada masing-masing dataset validasi dan *test* dapat dilihat pada Gambar 4.3.



Gambar 4.1 Jumlah Citra Data *Training* Sebelum dan Sesudah *Undersampling*



Gambar 4.2 Jumlah Pixel Data *Training* Sebelum dan Sesudah *Undersampling*



Gambar 4.3 Jumlah Pixel dan Citra Data Validasi dan *Testing*

#### 4.1.2. Pengujian *Preprocessing* Segmentasi

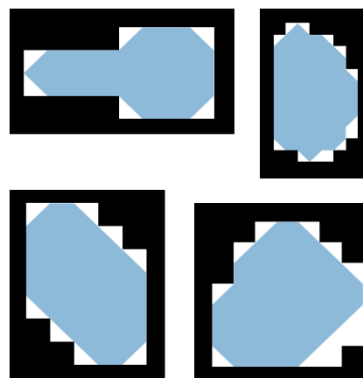
Pengujian *preprocessing* segmentasi dilakukan dengan melakukan perbandingan antara pemetaan vektor koordinat *floating point* dalam bentuk vektor terhadap hasil rasterisasi. Perbandingan tersebut dapat dilihat pada Gambar 4.4

#### 4.1.3. Pengujian *Preprocessing* Citra

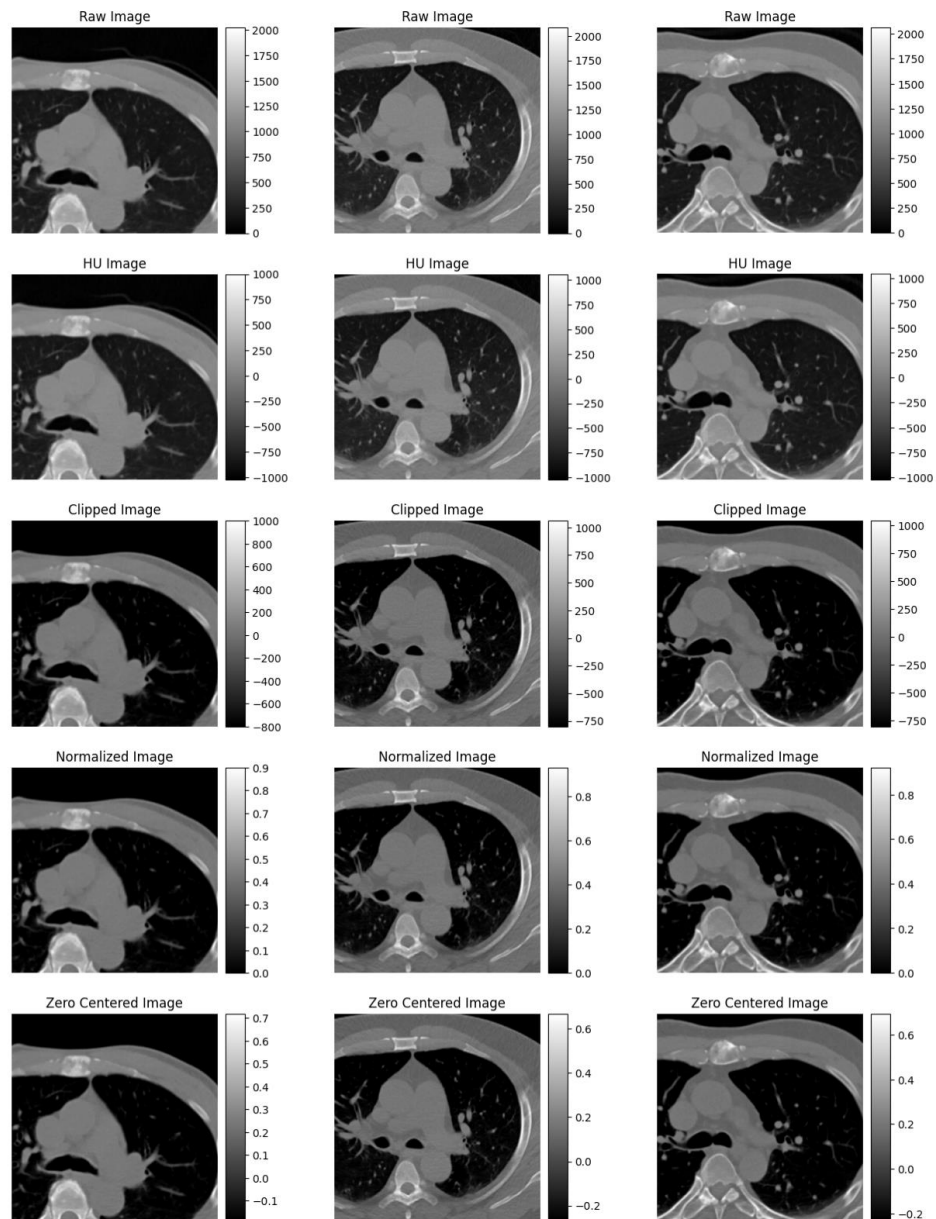
Pengujian *preprocessing* citra dilakukan secara kuantitatif melalui rentang nilai citra selama proses *preprocessing*. Pada pembacaan awal citra mentah data citra mentah dengan tipe *uint16* sehingga nilainya memiliki rentang 0-65535. Pada konversi ke *Hounsfield Unit* akan terjadi perubahan pada rentang yang sebelumnya sepenuhnya positif menjadi negative dan positif. Pada saat *clipping*, rentang dari *Hounsfield Unit* akan membuat nilai minimal citra menjadi -800 dan maksimal 1200. Pada saat normalisasi, rentang nilai seharusnya berada pada 0-1. Pada saat *zero centering* rentang nilai akan berada antara -1 - 1. Proses perubahan rentang nilai citra selama proses *processing* dapat dilihat pada Gambar 4.5.

#### 4.2. Pengujian Model

Proses *training* model dilakukan menggunakan 4 GPU RTX 3090 selama 311 epoch. Model yang disimpan merupakan model dengan *loss* terhadap data validasi terendah, dengan model terbaik merupakan model dari epoch 225. Setelah melalui proses *training*, model melalui proses evaluasi secara kuantitatif dan kualitatif. Proses evaluasi ini dilakukan menggunakan data *testing* untuk memastikan model tidak diuji pada data yang belum pernah sama sekali dilihat oleh model. Secara kuantitatif, model akan dievaluasi berdasarkan metrik *dice coefficient*, *precision*, *recall*. Data kuantitatif terkait metrik performa dapat dilihat lebih lanjut pada Tabel 4.1. Untuk melakukan evaluasi secara kualitatif, hasil prediksi model dibandingkan dengan *ground truth* data *testing*. Perbandingan hasil prediksi dengan *ground truth*, menemukan kasus *true positive*, *true negative*, dan *false negative*, kasus *false positive* dan beberapa kasus hasil segmentasi yang terdegradasi. Kasus-kasus tersebut dapat diamati pada Gambar 4.6 sampai Gambar 4.11.



Gambar 4.4 Perbandingan vektor koordinat *floating point*

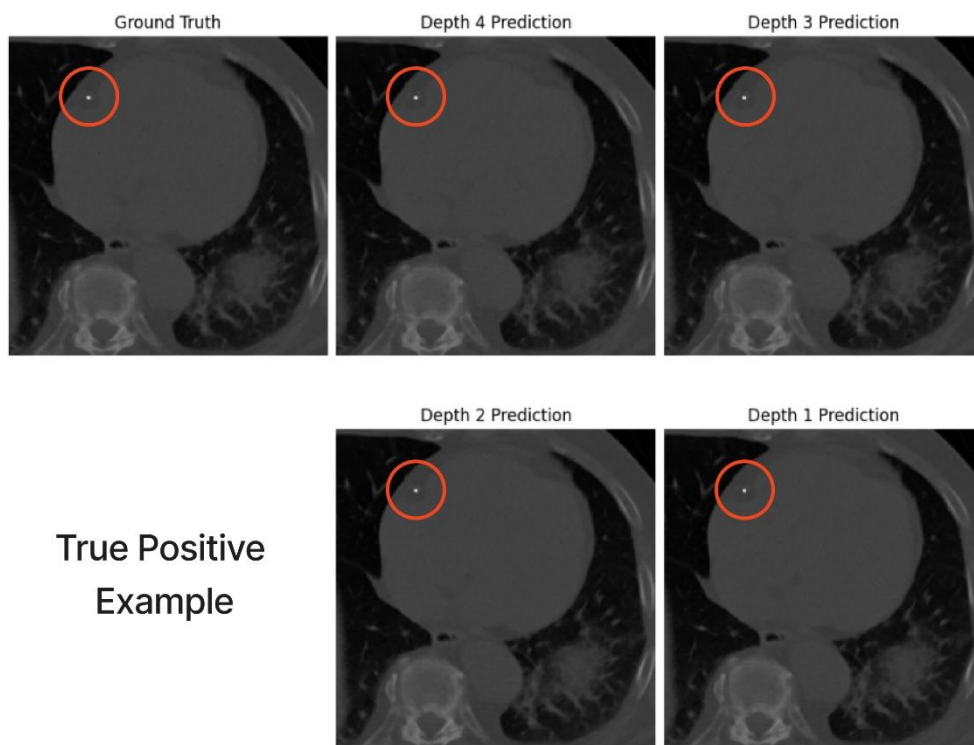


Gambar 4.5 Perbandingan Rentang Nilai Citra Selama Proses *Preprocessing*

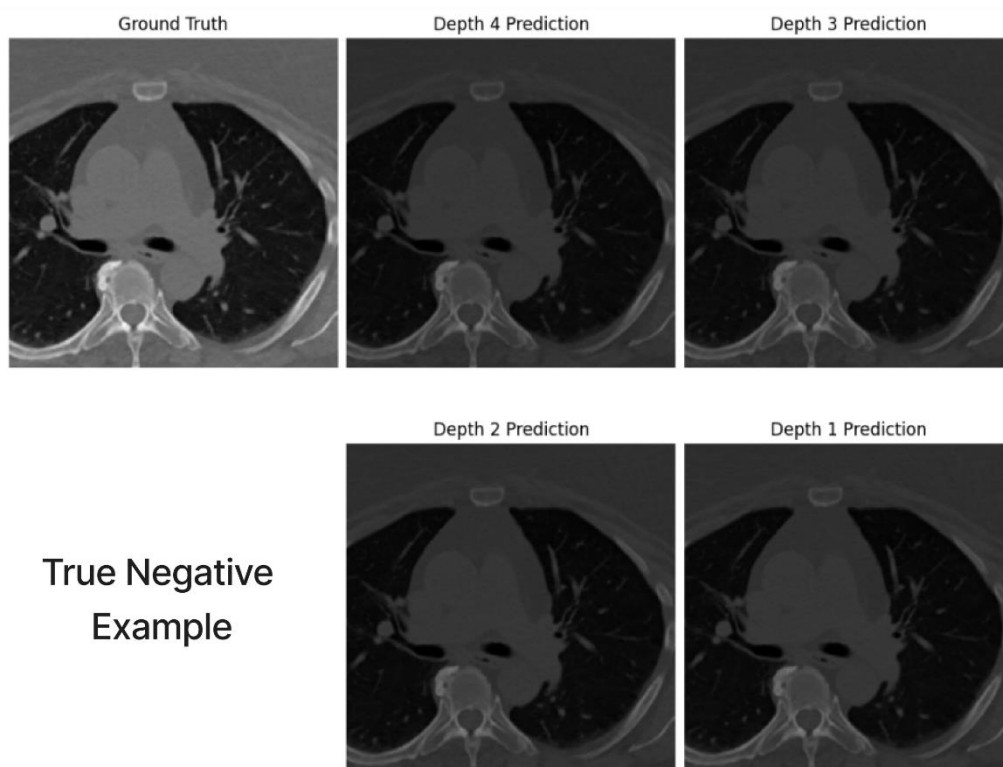
Tabel 4.1 Perbandingan Metrik Performa *Pruning* Model Pada Data *Testing*

<i>Pruning</i>	Parameter	Dice	Precision	Recall
DS4	4.143.921	0,9378	0,9354	0,6203
DS3	848.561	0,9092	0,9121	0,6224
DS2	231.969	0,8018	0,8730	0,3928
DS1	48.433	0,0788	0,2866	0,2049

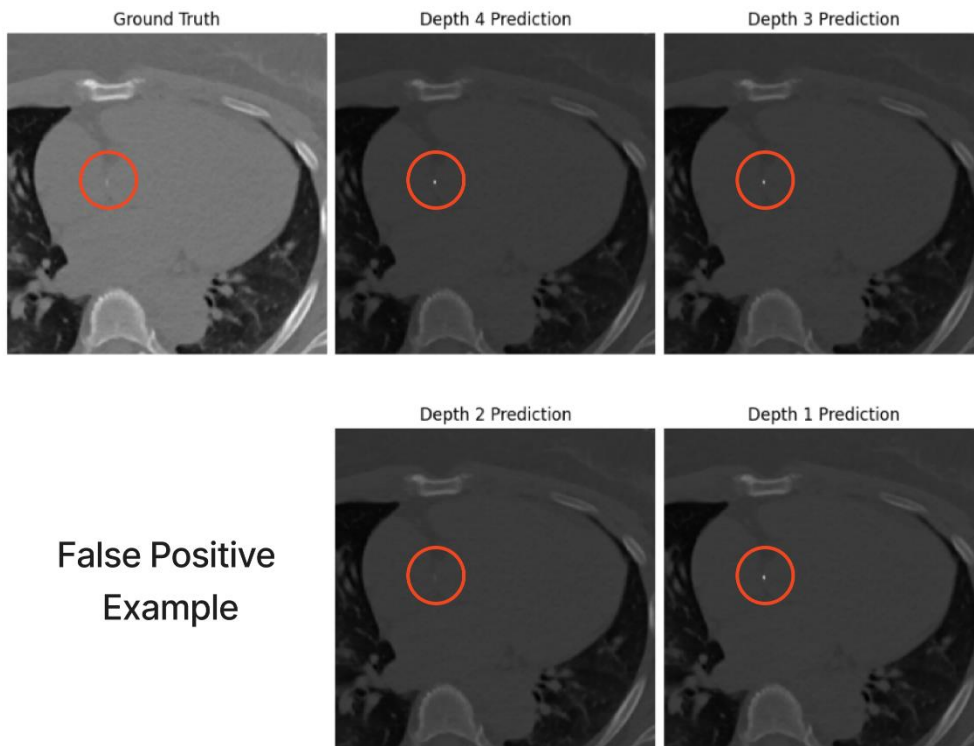




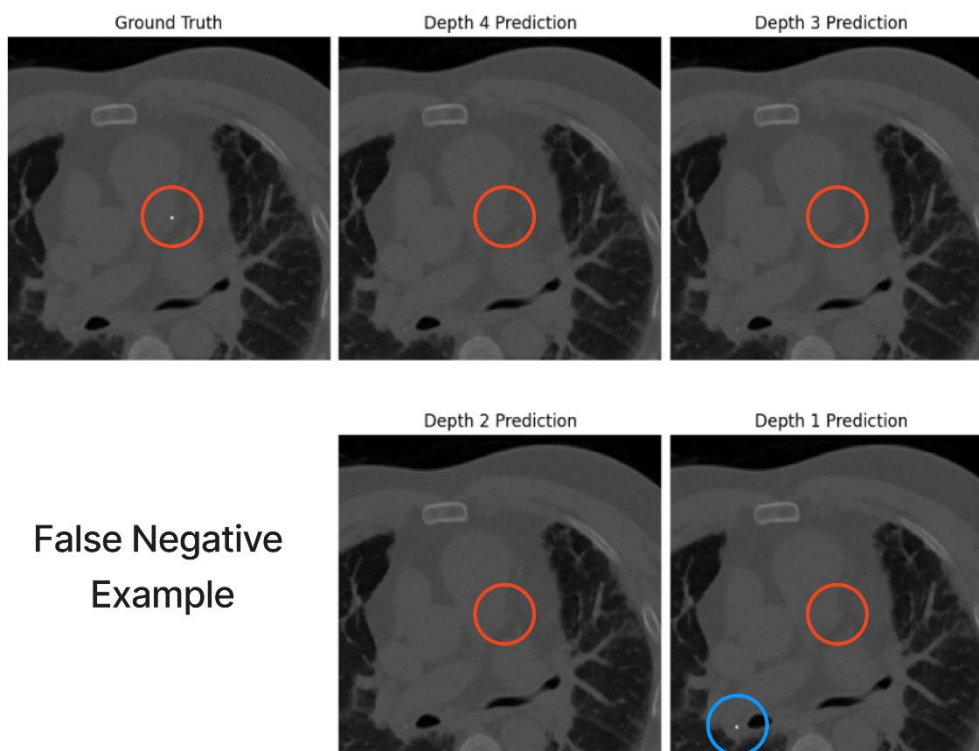
Gambar 4.6 Contoh Hasil *True Positive* Model



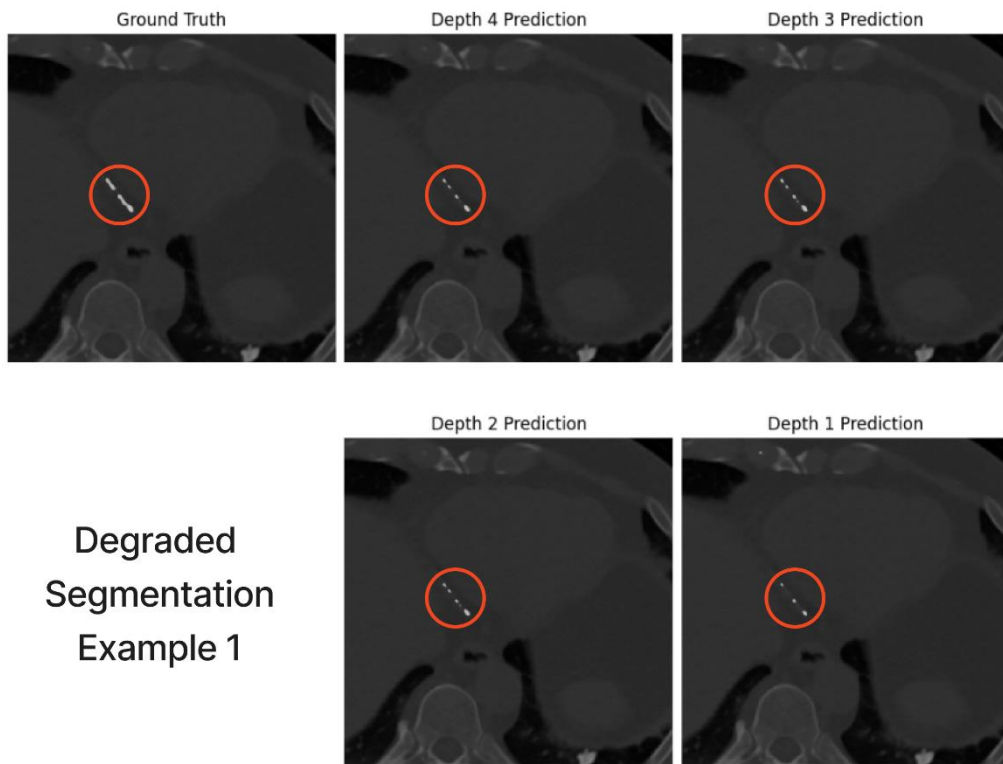
Gambar 4.7 Contoh Hasil *True Negative* Model



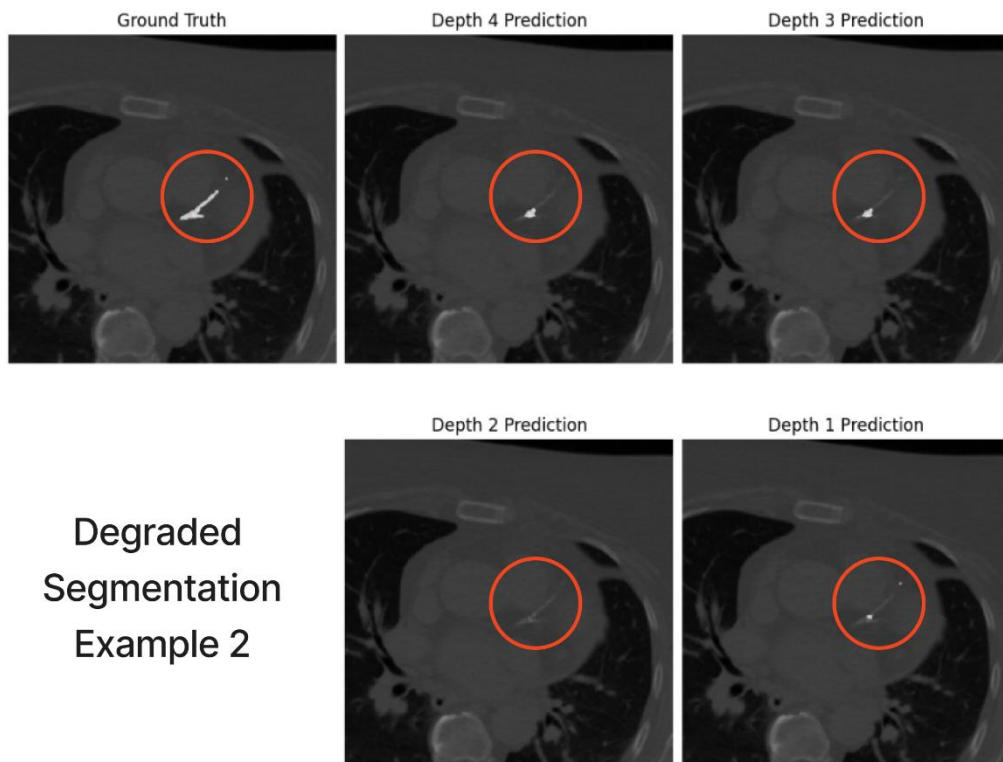
Gambar 4.8 Contoh Hasil *False Positive* Model



Gambar 4.9 Contoh Hasil *False Negative* Model



Gambar 4.10 Contoh 1 Hasil Model Yang Terdegradasi

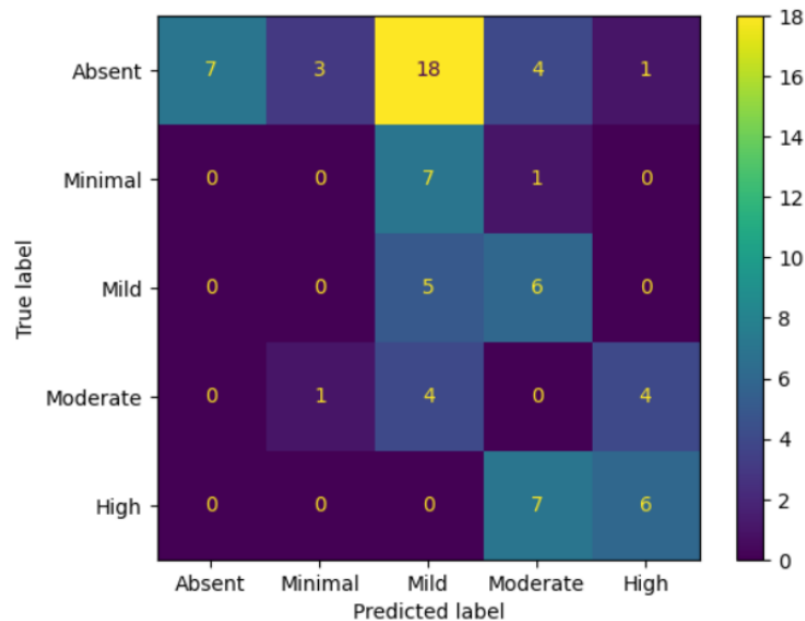


### 4.3. Pengujian Sistem

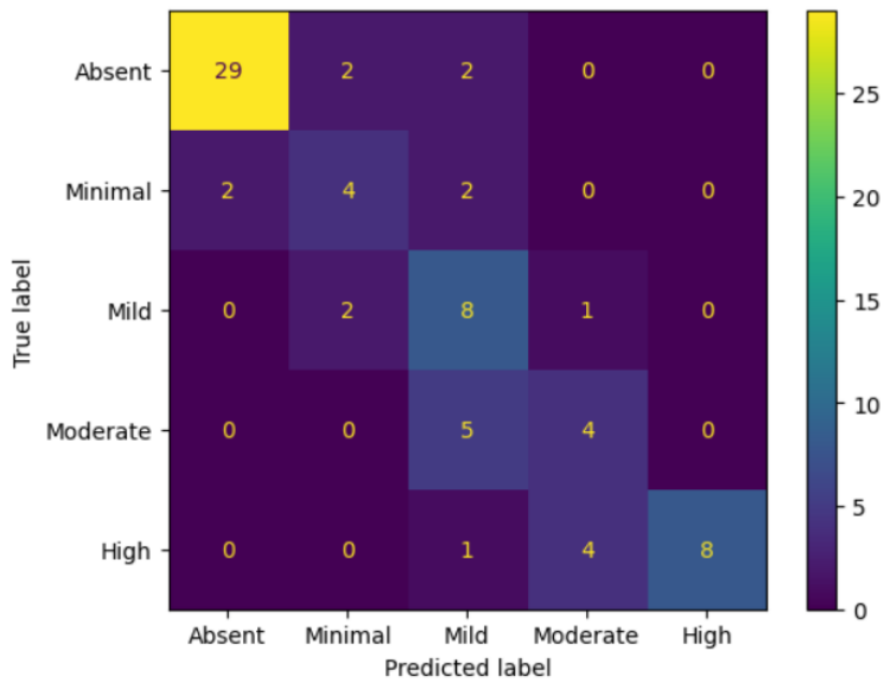
Model yang sudah melalui proses *training* diintegrasikan dengan sistem yang menyambungkan bagian *preprocessing* dan *postprocessing* pada model. Pengujian ini dilakukan melalui perbandingan antara nilai *ground truth* Agatston dari pasien dengan hasil prediksi dari sistem melalui kalkulasi RMSE serta perhitungan rata-rata waktu *inference* sistem untuk mengolah seluruh citra pada pasien. Pengujian rata-rata waktu *inference* dilakukan menggunakan 1 GPU RTX 3090 dengan data 76 pasien dari data *testing*. Proses pengujian dilakukan dalam dua tahap, tahap pertama adalah tahap *warmup* dimana sistem akan melakukan prediksi pada setiap data *testing* tanpa pencatatan waktu, dan diikuti dengan tahap *testing* dimana sistem melakukan prediksi dan waktu inferensi untuk masing-masing pasien dicatat dan diolah untuk didapatkan rata-rata dan standar deviasinya. Proses pengujian dilakukan secara *headless* dimana proses *testing* dilakukan menggunakan *script* dan hasil prediksi tidak disimpan untuk memastikan perhitungan waktu dilakukan tanpa *overhead* akibat penyimpanan hasil prediksi. Perbandingan performa antar *pruning* model terhadap sistem dapat dilihat pada Tabel 4.2 dan *confusion matrix* dari hasil pengujian sistem dapat dilihat melalui Gambar 4.12 sampai Gambar 4.15

Tabel 4.2 Perbandingan Performa *Pruning* Model Pada Sistem

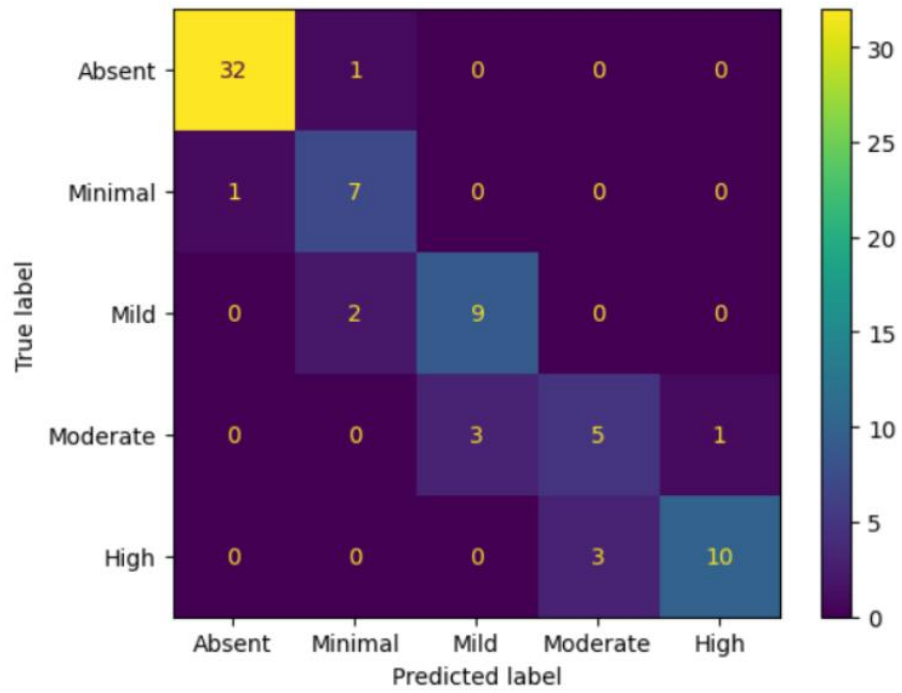
<i>Pruning</i>	Parameter	Kappa	MAE	Rata-Rata Waktu (s/pasien)
DS4	4.143.921	0,813	90,74	2,59 ± 0,4
DS3	848.561	0,794	93,27	2,35 ± 0,33
DS2	231.969	0,612	155,82	2,09 ± 0,30
DS1	48.433	0,086	237,42	1,99 ± 0,28

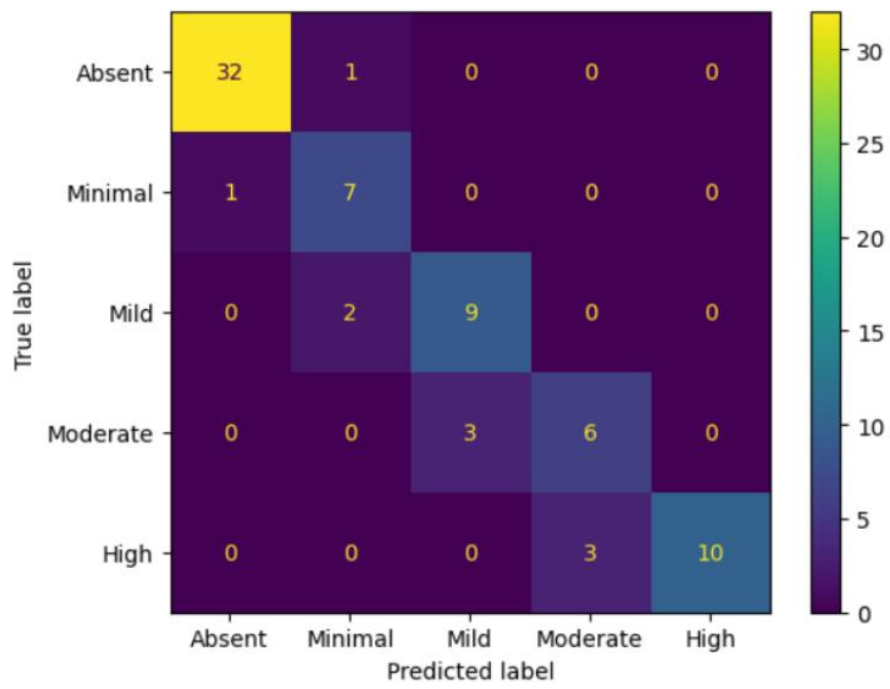


Gambar 4.12 *Confusion Matrix Pruning Model DS1*



Gambar 4.13 *Confusion Matrix Pruning Model DS2*





## **BAB V**

### **PEMBAHASAN**

#### **5.1. Pembahasan Hasil Pengujian *Preprocessing***

Data COCA merupakan data yang berisikan lesi segmentasi kalsium pada pembuluh darah jantung. Bentuk pembuluh darah jantung yang relatif kecil dan kalsium yang tidak sepenuhnya menutupi pembuluh darah akan membuat segmentasi lebih kecil dibandingkan dengan permasalahan segmentasi pada umumnya. *Preprocessing* diperlukan untuk membantu melawan ketidakseimbangan antara jumlah piksel dengan kalsium dan jumlah piksel tanpa kalsium.

##### **5.1.1. Pembahasan Hasil Pengujian Seleksi Data dan *Random Undersampling***

Proses seleksi data mengurangi  $\pm 6\%$  dari seluruh jumlah pasien. Pengurangan tersebut relatif kecil jika dibandingkan dengan jumlah pasien yang akhirnya digunakan untuk proses *training* model. Proses *random undersampling* yang diberikan pada data *training* menyamakan jumlah citra dengan CAC dan tanpa CAC. Proses tersebut mengurangi ketidakseimbangan data dalam segi jumlah citra dengan CAC dan tanpa CAC. Pada segi jumlah piksel dengan CAC dan tanpa CAC terdapat penurunan sebesar 82% pada jumlah piksel tanpa CAC yang mengurangi perbandingan awal jumlah piksel dengan CAC dan tanpa CAC dari 0,005%:99,995% menjadi 0,027%:99,973% setelah *random undersampling*. Walaupun peningkatan dari perbandingan antar jumlah piksel dengan CAC dan tanpa CAC tidak terlihat cukup besar, jika dikalkulasikan perbandingan tersebut mengalami peningkatan hingga 440%. Melalui proses *random undersampling* peningkatan jumlah piksel dengan CAC membuat model akan lebih mudah mempelajari perbedaan piksel dengan CAC dan tanpa CAC. Data validasi dan *testing* tidak melalui proses *random undersampling* untuk memastikan persebaran data yang digunakan untuk pengujian model tetap mengikuti dengan kondisi nyata yang ada, dimana perbandingan jumlah citra serta piksel dengan CAC akan jauh lebih kecil dibandingkan dengan jenis citra tanpa CAC. Berdasarkan grafik pada Gambar 4.3, persebaran antara jumlah citra dan piksel dengan CAC dan tanpa CAC memiliki perbandingan yang sama, dengan perbedaannya hanya terletak pada magnitudo dari jumlah data *testing* dan validasi yang timbul akibat pembagian data menjadi 70%:20%:10%.

##### **5.1.2. Pembahasan Hasil Pengujian *Preprocessing* Segmentasi**

Proses rasterisasi yang dilakukan pada koordinat sumbu x dan y dari lesi menghasilkan hasil diskritisasi yang cukup akurat seperti yang bisa dilihat pada Gambar 4.4. Perbandingan polygon yang dibentuk oleh vektor koordinat seluruhnya sudah tercakup pada hasil rasterisasi. Pada beberapa hasil rasterisasi terdapat beberapa piksel tambahan yang terlihat. Hal ini disebabkan oleh *library*

*geo-rasterize* yang mengimplementasikan algoritma rasterisasi dari GDAL dengan menggunakan *flag* “*ALL\_TOUCHED=True*” dimana dengan *flag* tersebut algoritma rasterisasi akan tetap memasukkan piksel ke dalam hasil diskritisasi jika piksel tersebut tersentuh oleh garis. Konfigurasi ini tidak dapat dinonaktifkan pada *library geo-rasterize* karena hanya ada satu implementasi dari algoritma rasterisasi tersebut. Peningkatan luasan lesi akibat hal ini tidak terlalu signifikan karena hanya berpengaruh pada keliling dari lesi dan tidak terjadi pada seluruh lesi.

### 5.1.3. Pembahasan Hasil Pengujian *Preprocessing* Citra

Proses *preprocessing* citra dilakukan untuk menyiapkan citra dengan rentang nilai heterogen dan kecil. Nilai yang heterogen dan kecil akan membantu model mempelajari keunikan antar data dalam proses *training*. Proses konversi data mentah menjadi *Hounsfield Unit* dilakukan karena nilai HU sangat penting khususnya pada penelitian ini karena nilai HU merupakan standar dalam pengelompokan *weight* dari rentang kalsium yang digunakan pada proses kalkulasi nilai *Agatston*.

Proses *clipping* dilakukan dengan membatasi nilai citra pada rentang -800 – 1200. Data dengan nilai di luar rentang tersebut akan dibulatkan ke nilai minimum atau maksimum terdekat. Proses *clipping* digunakan untuk mengeliminasi data *outlier* dan memperbaiki kontras dari citra seperti yang terlihat pada Gambar 4.5 dimana sebelum *clipping* citra terlihat buram dan setelah *clipping* terlihat lebih tajam.

Proses *normalisasi* dilakukan untuk memperkecil rentang nilai citra yang sebelumnya berada pada rentang -800 – 1200 ke rentang 0 – 1. Mengubah rentang nilai ke rentang yang lebih kecil akan mengecilkan besar dari data dan dapat memudahkan model dalam melakukan *feed forward* dan *backward* tanpa menggunakan memori yang besar.

Proses *zero centering* dilakukan melalui pergeseran data ternormalisasi dengan cara membuat nilai rerata pada data ternormalisasi berada pada nilai nol. Proses ini akan membantu agar model tidak bias terhadap data dan belajar melalui perbedaan relatif antar piksel dibandingkan mempelajari melalui nilai absolut dari data.

Secara singkat, proses konversi ke HU dilakukan untuk mengikuti standar kalkulasi nilai *Agatston*, proses *clipping* dilakukan untuk meningkatkan ketajaman citra, dan dua proses terakhir, yaitu normalisasi dan *zero centering* mengubah rentang nilai citra ke rentang yang lebih mudah digunakan oleh model.

## 5.2. Pembahasan Hasil Pengujian Model

Berdasarkan hasil pengujian yang tertera pada Tabel 4.1 terdapat tren yang dapat dilihat pada metrik yang didapatkan. Untuk hasil *dice coefficient*, *precision*, dan *recall* terlihat mengalami penurunan seiring dengan berkurangnya jumlah parameter pada model. Penurunan tersebut sesuai dengan hasil yang didapatkan pada penelitian UNet++ oleh Zhou et al [12]. Penelitian tersebut menemukan



penurunan performa model seiring dengan *pruning model* ketika melalui proses *training* pada mode *deep supervision*. Model DS4 memiliki performa *dice* dan *precision* tertinggi dengan nilai 0,9378 untuk *dice coefficient* dan 0,9354 untuk *precision*. Model DS3 memiliki nilai *recall* terbaik dengan perbedaan 0,0021 poin dibandingkan dengan nilai *recall* model DS4.

Secara keseluruhan, performa *precision* untuk keempat model lebih tinggi dibandingkan dengan performa *recall*. Hal ini mengindikasikan model cenderung memiliki lebih banyak kasus *false negative* dibandingkan dengan *false positive*. Proses kalkulasi kasus *false negative* dilakukan melalui perbandingan untuk tiap piksel pada segmentasi. Terdapat dua kasus *false negative* yang dihasilkan oleh model berdasarkan contoh hasil prediksi pada Gambar 4.9, Gambar 4.10, dan Gambar 4.11. Kasus *false negative* pertama adalah *false negative* total dimana pada kasus ini model tidak menemukan kalsium sama sekali pada citra seperti yang dapat dilihat pada Gambar 4.9. Kasus *false negative* kedua adalah *false negative* sebagian dimana hasil prediksi model terdegradasi menyisakan lesi yang besar menjadi lesi yang lebih kecil dari seharusnya seperti yang dapat dilihat pada Gambar 4.10 dan Gambar 4.11. Kedua kasus *false negative* ini memiliki kecenderungannya masing-masing, dimana pada kasus *false negative* total cenderung terjadi pada lesi kecil, sedangkan pada kasus *false negative* sebagian cenderung terjadi pada lesi yang cukup besar.

Kasus *false negative* yang dialami model memiliki hubungan dengan ketidakseimbangan data yang digunakan. Proses *random undersampling* yang diberikan pada data melakukan peningkatan cukup signifikan pada perbandingan jumlah piksel dengan CAC dan tanpa CAC. Walau demikian mayoritas data yang digunakan untuk proses *training* tetap memiliki mayoritas piksel negatif (tanpa CAC). Ketidakseimbangan tersebut akan membuat model lebih bias terhadap prediksi piksel tanpa CAC dibandingkan piksel dengan CAC. Jumlah keragaman kasus piksel dengan CAC yang lebih sedikit juga berpengaruh pada model sehingga mempersulit model menggeneralisasi karakteristik area dengan CAC. Tanpa pemberian *random undersampling* pada data, ketidakseimbangan yang semakin signifikan akan mempersulit model akibat bias yang jauh lebih besar terhadap piksel tanpa CAC.

### 5.3. Pembahasan Hasil Pengujian Sistem

Model memiliki peranan besar terhadap sistem karena hasil segmentasi model digunakan untuk proses kalkulasi nilai *Agatston*. Hasil pengujian sistem pada Tabel 4.2 menunjukkan terdapat tren yang sama antara performa model dengan performa sistem. Terdapat penurunan terhadap metrik *Cohen's Kappa*, *MAE* dan waktu rata-rata prediksi per pasien seiring dengan turunnya jumlah parameter pada tiap tingkatan *pruning model*. Penurunan waktu rata-rata sistem dalam prediksi nilai per pasien terjadi akibat rendahnya jumlah parameter seiring dengan menurunnya jumlah parameter dalam model.

Performa *Cohen's Kappa* menunjukkan perbandingan tingkat kesepakatan klasifikasi suatu sistem terhadap *ground truth*. Sistem yang menggunakan model DS4 menunjukkan nilai *Cohen's Kappa* tertinggi dengan nilai 0,813 yang jika dikategorikan berarti model ini memiliki kesepakatan yang hampir sempurna. Sistem yang menggunakan model DS3 dan DS2 dengan masing-masing nilai *Cohen's Kappa* 0,794 dan 0,612 berada pada rentang kategori yang sama dimana model dikategorikan memiliki kesepakatan yang signifikan. Model DS1 memiliki *Cohen's Kappa* 0,086 yang mengategorikan model memiliki kesepakatan yang kecil. Model DS4 dan DS3 memiliki kemampuan lebih baik dalam stratifikasi kelas risiko penyakit jantung dibandingkan model DS2 dan DS1.

Secara keseluruhan, nilai performa *MAE* sistem dari seluruh jenis model cukup tinggi, dengan model DS4 memiliki nilai *MAE* terendah 90,74. Nilai *MAE* yang cenderung tinggi akan tetapi nilai *Cohen's Kappa* yang tinggi juga menunjukkan bahwa dengan *error* dari hasil nilai *Agatston* yang cukup tinggi sistem masih dapat mengklasifikasikan pasien ke rentang klasifikasi dengan cukup akurat.

*Confusion matrix* dapat menunjukkan kecenderungan hasil prediksi untuk tiap kelas klasifikasi. Sistem yang menggunakan model DS1 memiliki *confusion matrix* yang menunjukkan model memiliki kesulitan untuk memberikan klasifikasi yang tepat. Nilai *precision* yang rendah dari model DS1 menyebabkan banyaknya *false positive* yang diprediksi, sehingga seperti yang bisa dilihat pada Gambar 4.12, cenderung memberikan hasil klasifikasi prediksi kelas yang lebih tinggi dibandingkan dengan *ground truth*. Sistem yang menggunakan model DS2, DS3, dan DS4 memiliki *confusion matrix* seperti yang bisa dilihat pada Gambar 4.13 sampai Gambar 4.15 yang menunjukkan sistem dapat memberikan klasifikasi yang cukup akurat dengan sedikit kasus klasifikasi dengan tingkatan yang lebih rendah dibanding dengan *ground truth*. Kasus klasifikasi yang lebih rendah merupakan efek dari rendahnya nilai *recall* dari model sehingga ada kecenderungan dari sistem untuk menghasilkan nilai *Agatston* yang lebih rendah dibandingkan *ground truth*.

#### **5.4. Perbandingan Dengan Penelitian Sebelumnya**

Pada penelitian ini, penulis mengembangkan sistem otomasi skoring arteri kalsium menggunakan model CNN berbasis UNet++ menggunakan *pretrained encoder* MobileNetV2. Tabel 5.1 merangkum perbandingan model serta sistem dari penelitian sebelumnya yaitu penelitian oleh Eng et al, Hong et al, Wang et al, Sandstedt et al, dan Gogin et al. Jumlah parameter model pada penelitian Eng et al, Hong et al, dan Gogin et al merupakan estimasi menggunakan informasi yang ada penelitian tersebut. Pada penelitian Eng et al, model yang digunakan merupakan model UNet dengan *encoder* SE-ResNext50 [6]. SE-ResNext50 memiliki 28 juta parameter [28], sehingga dengan mengasumsikan jumlah parameter pada *decoder* merupakan satu per empat dari parameter pada *encoder*, model penelitian Eng et al memiliki sekitar 35 juta parameter. Penelitian Hong et al menyebutkan model penelitian tersebut menggunakan parameter filter 32, 64, 128, 256, dan 512 [7].

Parameter tersebut sama dengan model dasar UNet++ tanpa *pretrained encoder* dengan total parameter sekitar 9 juta [12]. Penelitian Wang et al menggunakan 3D Resnet biasa dan memiliki 2 miliar parameter [29]. Penelitian Gogin et al menyebutkan penggunaan 5 model 3D UNet dalam mode *ensemble* [30]. Model 3D UNet yang digunakan menggunakan filter 16, 32, 64 dan 128 atau sekitar setengah dari filter yang digunakan pada model asli 3D UNet. Model 3D UNet biasa memiliki 19,1 juta parameter, sehingga jumlah total parameter yang digunakan pada model Gogin et al adalah sekitar setengah untuk masing-masing model, dengan total parameter 5 kali 9,55 juta atau sekitar 47,8 juta parameter. Penelitian Sandstedt tidak menjelaskan tipe model yang digunakan pada penelitian tersebut dan hanya menjelaskan penelitian tersebut menggunakan *software* otomatis dengan model CNN [31]. Protokol pengujian waktu rata-rata prediksi pasien oleh Eng et al dan Sandstedt et al tidak disebutkan sehingga terdapat kemungkinan perbedaan kecepatan akibat perbedaan GPU yang digunakan. Berdasarkan tabel tersebut terdapat peningkatan dan penurunan dibandingkan dengan penelitian sebelumnya. Peningkatan yang didapatkan dari penelitian ini berupa penurunan jumlah parameter yang drastis dari penelitian sebelumnya dengan varian model terbesar (DS4) penelitian ini memiliki 4,1 juta parameter dengan performa *precision* dan waktu rata-rata prediksi yang unggul. Melalui penggunaan mode *deep supervision* pada penelitian ini, model turunan dari model terbesar (DS4), pada model DS3 memiliki parameter yang lebih rendah dibandingkan DS4 dengan sedikit penurunan pada performa. Penurunan yang didapatkan dari penelitian ini berupa performa *dice*, *recall* dan *Cohen's Kappa* yang lebih rendah dibandingkan dengan penelitian sebelumnya. Namun, varian model DS4 dan DS3 masih berada pada rentang nilai *Kappa* yang mengkategorikannya memiliki kesepakatan hampir sempurna dan kesepakatan signifikan serta masih bersaing dengan model UNet++ biasa.

Tabel 5.1 Perbandingan Performa Model Dan Sistem Penelitian Ini Terhadap Penelitian Dengan Pendekatan *One Shot* Sebelumnya

Penelitian	Parameter	Dice	<i>Precision</i>	<i>Recall</i>	<i>Kappa</i>	Waktu(s)
Wang et al [29]	2 Miliar	-	-	-	0,77	-
Gogin et al [30]	~47,8 Juta 5x9,6 Juta	0,974	-	-	0,894	-
Sandstedt et al [31]	-	-	-	-	0,919	9
Eng et al [6]	~35 Juta	-	-	-	0,89	3,5±2,1
Hong et al [7]	~9 Juta	0,86	0,88	0,87	0,931	-
UNet++	9 Juta	0,9396	0,9326	0,7229	0,747	3,36 ± 0,5
Penelitian ini: DS4	4,1 Juta	0,9378	0,9354	0,6203	0,813	2,59 ± 0,4
Penelitian ini: DS3	850 Ribu	0,9092	0,9121	0,6224	0,794	2,35 ± 0,33

## **BAB VI**

### **KESIMPULAN**

#### **6.1. Kesimpulan**

Pada penelitian ini dikembangkan sebuah sistem otomasi skoring kalsium arteri koroner menggunakan model UNet++ dengan *pretrained encoder* MobileNetV2. Data yang digunakan merupakan data COCA dari Stanford AIMI yang berisikan 789 pasien. Untuk mengatasi ketidakseimbangan data, data diproses melalui *random undersampling* dan model menggunakan fungsi *loss Log Cosh Dice Loss* yang dibuat untuk mengatasi ketidakseimbangan data selama proses *training*. Varian model terbesar (DS4) memiliki 4,1 juta parameter, *dice coefficient* bernilai 0,9378; *Kappa* bernilai 0,813; *precision* bernilai 0,9354; *recall* bernilai 0,6203; dan waktu rata-rata prediksi pasien bernilai  $2,59 \pm 0,4$ . Turunan dari model terbesar, khususnya varian DS3 memiliki parameter dan waktu rata-rata prediksi lebih rendah dibandingkan model DS4 dengan sedikit penurunan pada performa. Model pada penelitian ini memiliki jumlah parameter dan waktu rata-rata prediksi pasien yang lebih rendah serta performa *precision* yang lebih tinggi dibandingkan dengan penelitian sebelumnya dengan penurunan pada performa *dice coefficient*, *recall* dan *Kappa* yang masih berada pada rentang kategori klasifikasi yang hampir sempurna untuk model DS4, kategori klasifikasi yang signifikan untuk model DS3 serta masih bersaing dengan model UNet++ biasa sehingga arsitektur model CNN yang diusulkan pada penelitian ini dapat melakukan stratifikasi risiko penyakit jantung pasien.

#### **6.2. Saran**

Untuk pengembangan kedepan dari Tugas Akhir ini terdapat beberapa hal yang dapat dilakukan. Proses pengolahan data yang sebelumnya sangat sulit akibat minimnya informasi seperti indeks segmentasi dan indeks citra yang terbalik sudah dilalui melalui penelitian ini, sehingga penelitian selanjutnya dapat melakukan iterasi yang lebih cepat dalam pengembangan jenis model yang baru. Perubahan parameter pada penelitian ini seperti penggantian fungsi *loss*, penggantian parameter filter hingga penggantian *encoder* berbeda pada arsitektur UNet++ dapat memberikan hasil yang lebih optimal dari penelitian ini. Pada penelitian selanjutnya diharapkan ketidakseimbangan data dipertimbangkan kembali untuk memastikan model dapat mencapai titik konvergensi yang lebih baik lagi.

## DAFTAR PUSTAKA

- [1] ..., “Cardiovascular diseases (CVDs),” *WHO Newsroom*, 2021. [https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-\(cvds\)](https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)) (accessed Nov. 21, 2021).
- [2] A. Nurmasani and Y. Pristyanto, “Algoritme Stacking Untuk Klasifikasi Penyakit Jantung Pada Dataset Imbalanced Class,” *Pseudocode*, vol. 8, no. 1, pp. 21–26, 2021, doi: 10.33369/pseudocode.8.1.21-26.
- [3] P. Greenland, J. S. Alpert, G. A. Beller, E. J. Benjamin, M. J. Budoff, Z. A. Fayad, E. Foster, M. A. Hlatky, J. M. B. Hodgson, F. G. Kushner, M. S. Lauer, L. J. Shaw, S. C. Smith, A. J. Taylor, W. S. Weintraub, and N. K. Wenger, “2010 ACCF/AHA guideline for assessment of cardiovascular risk in asymptomatic adults: Executive summary: A report of the American College of cardiology foundation/American Heart association task force on practice guidelines,” *Circulation*, vol. 122, no. 25, pp. 2748–2764, Dec. 2010, doi: 10.1161/CIR.0b013e3182051bab.
- [4] D. C. Goff, D. M. Lloyd-Jones, G. Bennett, S. Coady, R. B. D’Agostino, R. Gibbons, P. Greenland, D. T. Lackland, D. Levy, C. J. O’Donnell, J. G. Robinson, J. S. Schwartz, S. T. Shero, S. C. Smith, P. Sorlie, N. J. Stone, and P. W. F. Wilson, “2013 ACC/AHA guideline on the assessment of cardiovascular risk: A report of the American college of cardiology/American heart association task force on practice guidelines,” *Circulation*, vol. 129, no. 25 SUPPL. 1, pp. 49–73, Jun. 2014, doi: 10.1161/01.CIR.0000437741.48606.98/-/DC1.
- [5] P. Greenland, M. J. Blaha, M. J. Budoff, R. Erbel, and K. E. Watson, “Coronary Calcium Score and Cardiovascular Risk,” *J. Am. Coll. Cardiol.*, vol. 72, no. 4, pp. 434–447, 2018, doi: 10.1016/j.jacc.2018.05.027.
- [6] D. Eng *et al.*, “Automated coronary calcium scoring using deep learning with multicenter external validation,” *npj Digit. Med.*, vol. 4, no. 1, 2021, doi: 10.1038/s41746-021-00460-1.
- [7] J. S. Hong, Y. H. Tzeng, W. H. Yin, K. T. Wu, H. Y. Hsu, C. F. Lu, H. R. Liu, and Y. Te Wu, “Automated coronary artery calcium scoring using nested U-Net and focal loss,” *Comput. Struct. Biotechnol. J.*, vol. 20, pp. 1681–1690, 2022, doi: 10.1016/j.csbj.2022.03.025.
- [8] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “MobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications,” 2017, Accessed: Oct. 10, 2022. [Online]. Available: <http://arxiv.org/abs/1704.04861>
- [9] M. Sandler, A. Howard, M. Zhu, A. Zhmoginov, and L. C. Chen, “MobileNetV2: Inverted Residuals and Linear Bottlenecks,” *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 4510–4520, 2018, doi: 10.1109/CVPR.2018.00474.
- [10] A. Arjmand, “Coronary Artery Calcium Score: A Review,” *Iran. Red Crescent Med. J.*, vol. 15, no. 12, p. 16616, 2013, doi: 10.5812/IRCMJ.16616.
- [11] O. Ronneberger, P. Fischer, and T. Brox, “U-Net: Convolutional Networks for Biomedical Image Segmentation”, Accessed: Dec. 14, 2022. [Online]. Available: <http://lmb.informatik.uni-freiburg.de/>
- [12] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, “UNet++:

- Redesigning Skip Connections to Exploit Multiscale Features in Image Segmentation,” *IEEE Trans. Med. Imaging*, vol. 39, no. 6, pp. 1856–1867, 2020, doi: 10.1109/TMI.2019.2959609.
- [13] J. M. Shin, T. H. Kim, J. Y. Kim, and C. H. Park, “Coronary artery calcium scoring on non-gated, non-contrast chest computed tomography (CT) using wide-detector, high-pitch and fast gantry rotation: comparison with dedicated calcium scoring CT,” *J. Thorac. Dis.*, vol. 12, no. 10, p. 5783, Oct. 2020, doi: 10.21037/JTD-20-1371.
- [14] J. Y. Kim, Y. J. Suh, K. Han, and B. W. Choi, “Reliability of Coronary Artery Calcium Severity Assessment on Non-Electrocardiogram-Gated CT: A Meta-Analysis,” *Korean J. Radiol.*, vol. 22, no. 7, p. 1034, Jul. 2021, doi: 10.3348/KJR.2020.1047.
- [15] H. S. Hecht, P. Cronin, M. J. Blaha, M. J. Budoff, E. A. Kazerooni, J. Narula, D. Yankelevitz, and S. Abbara, “2016 SCCT/STR guidelines for coronary artery calcium scoring of noncontrast noncardiac chest CT scans: A report of the Society of Cardiovascular Computed Tomography and Society of Thoracic Radiology,” *J. Cardiovasc. Comput. Tomogr.*, vol. 11, no. 1, pp. 74–84, Jan. 2017, doi: 10.1016/J.JCCT.2016.11.003.
- [16] ..., “TFRecord and tf.train.Example|TensorFlow Core,” *Tensorflow Core Documentation*. [https://www.tensorflow.org/tutorials/load\\_data/tfrecord](https://www.tensorflow.org/tutorials/load_data/tfrecord) (accessed Dec. 13, 2022).
- [17] J. E. Bresenham, “Algorithm for computer control of a digital plotter,” *IBM Syst. J.*, vol. 4, no. 1, pp. 25–30, 1965, doi: 10.1147/sj.41.0025.
- [18] ..., “DICOM PS3.3 2022d Information Object Definitions.” Medical Imaging & Technology Alliance, p. 1854, 2022. [Online]. Available: [https://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect\\_C.11.html#sect\\_C.11.1](https://dicom.nema.org/medical/dicom/current/output/chtml/part03/sect_C.11.html#sect_C.11.1)
- [19] A. Gholami, K. Kwon, B. Wu, Z. Tai, X. Yue, P. Jin, S. Zhao, and K. Keutzer, “SqueezeNext: Hardware-aware neural network design,” *IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. Work.*, vol. 2018-June, pp. 1719–1728, 2018, doi: 10.1109/CVPRW.2018.00215.
- [20] A. Howard, M. Sandler, B. Chen, W. Wang, L. C. Chen, M. Tan, G. Chu, V. Vasudevan, Y. Zhu, R. Pang, Q. Le, and H. Adam, “Searching for mobileNetV3,” *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2019-October, pp. 1314–1324, 2019, doi: 10.1109/ICCV.2019.00140.
- [21] S. Jadon, “A survey of loss functions for semantic segmentation,” *2020 IEEE Conf. Comput. Intell. Bioinforma. Comput. Biol. CIBCB 2020*, Jun. 2020, doi: 10.1109/CIBCB48159.2020.9277638.
- [22] X. Li, X. Sun, Y. Meng, J. Liang, F. Wu, and J. Li, “Dice loss for data-imbalanced NLP tasks,” *Proc. Annu. Meet. Assoc. Comput. Linguist.*, no. 2, pp. 465–476, 2020, doi: 10.18653/v1/2020.acl-main.45.
- [23] S. Narang, G. Damos, E. Elsen, P. Micikevicius, J. Alben, D. Garcia, B. Ginsburg, M. Houston, O. Kuchaiev, G. Venkatesh, and H. Wu, “Mixed precision training,” *6th Int. Conf. Learn. Represent. ICLR 2018 - Conf. Track Proc.*, pp. 1–12, 2018.
- [24] IEEE Computer Society, *754-2019 - IEEE Standard for Floating-Point Arithmetic*. 2019.
- [25] A. Rosenfeld and J. L. Pfaltz, “Sequential Operations in Digital Picture

- Processing,” *J. ACM*, vol. 13, no. 4, pp. 471–494, 1966, doi: 10.1145/321356.321357.
- [26] F. Bolelli, S. Allegretti, L. Baraldi, and C. Grana, “Spaghetti Labeling: Directed Acyclic Graphs for Block-Based Connected Components Labeling,” *IEEE Trans. Image Process.*, vol. 29, no. 1, pp. 1999–2012, 2020, doi: 10.1109/TIP.2019.2946979.
- [27] D. Shreya, D. I. Zamora, G. S. Patel, I. Grossmann, K. Rodriguez, M. Soni, P. K. Joshi, S. C. Patel, and I. Sange, “Coronary Artery Calcium Score - A Reliable Indicator of Coronary Artery Disease?,” *Cureus*, vol. 13, no. 12, pp. 1–8, 2021, doi: 10.7759/cureus.20149.
- [28] J. Hu, L. Shen, S. Albanie, G. Sun, and E. Wu, “Squeeze-and-Excitation Networks,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 2011–2023, 2020, doi: 10.1109/TPAMI.2019.2913372.
- [29] W. Wang, H. Wang, Q. Chen, Z. Zhou, R. Wang, N. Zhang, Y. Chen, Z. Sun, and L. Xu, “Coronary artery calcium score quantification using a deep-learning algorithm,” *Clin. Radiol.*, vol. 75, no. 3, pp. 237.e11–237.e16, 2020, doi: 10.1016/j.crad.2019.10.012.
- [30] N. Gogin, M. Viti, L. Nicodème, M. Ohana, H. Talbot, U. Gencer, M. Mekukosokeng, T. Caramella, Y. Diascorn, J. Y. Airaud, M. S. Guillot, Z. Bensalah, C. Dam Hieu, B. Abdallah, I. Bousaid, N. Lassau, and E. Mousseaux, “Automatic coronary artery calcium scoring from unenhanced-ECG-gated CT using deep learning,” *Diagn. Interv. Imaging*, vol. 102, no. 11, pp. 683–690, 2021, doi: 10.1016/j.diii.2021.05.004.
- [31] M. Sandstedt, L. Henriksson, M. Janzon, G. Nyberg, J. Engvall, J. De Geer, J. Alfredsson, and A. Persson, “Evaluation of an AI-based, automatic coronary artery calcium scoring software,” *Eur. Radiol.*, vol. 30, no. 3, pp. 1671–1678, 2020, doi: 10.1007/s00330-019-06489-x.

## BIODATA PENULIS



Penulis lahir di Batam, 11 Mei 2001. Penulis adalah anak terakhir dari dua bersaudara. Penulis menempuh SD dan SMP di Sekolah Djuwita Batam dan melanjutkan pendidikan menengah atas di SMA Negeri 1 Batam. Pada tahun 2019 penulis melanjutkan studi S1 Departemen Teknik Biomedik Institut Teknologi Sepuluh Nopember dengan peminatan Pencitraan dan Pengolahan Citra Medika. Penulis menghabiskan waktu kuliah sembari bekerja sebagai *Infrastructure Engineer* di *Zero One Group* dan *Freelance Python Developer*.





**BIOMEDICAL ENGINEERING ITS  
2023**