



TESIS - EE185401

**DETEKSI LOKASI TULANG MANUSIA PADA CITRA  
*ULTRASOUND* BERBASIS *DEEP LEARNING*  
MENGUNAKAN CNN ARSITEKTUR YOLOv3**

R. ARIF FIRDAUS LAZUARDI  
07111650052003

DOSEN PEMBIMBING  
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.  
Dr. Eko Mulyanto Yuliarno, ST., MT.

PROGRAM MAGISTER  
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2019





TESIS - EE185401

**DETEKSI LOKASI TULANG MANUSIA PADA CITRA  
*ULTRASOUND* BERBASIS *DEEP LEARNING*  
MENGUNAKAN CNN ARSITEKTUR YOLOv3**

R. ARIF FIRDAUS LAZUARDI  
07111650052003

DOSEN PEMBIMBING  
Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.  
Dr. Eko Mulyanto Yuliarno, ST., MT.

PROGRAM MAGISTER  
BIDANG KEAHLIAN JARINGAN CERDAS MULTIMEDIA  
DEPARTEMEN TEKNIK ELEKTRO  
FAKULTAS TEKNOLOGI ELEKTRO  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA  
2019



## LEMBAR PENGESAHAN TESIS

Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar  
**Magister Teknik (MT)**

di

**Institut Teknologi Sepuluh Nopember**

Oleh:

**R. ARIF FIRDAUS LAZUARDI**

**NRP: 07111650052003**

Tanggal Ujian: 25 Juni 2019

Periode Wisuda: September 2019

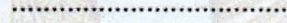
Disetujui oleh:

**Pembimbing:**

1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.  
NIP: 195809161986011001



2. Dr. Eko Mulyanto Yuliano, ST., MT.  
NIP: 196806011995121009



**Penguji:**

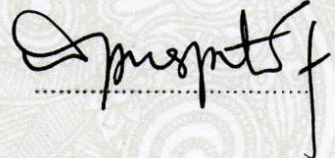
1. Dr. I Ketut Eddy Purnama, ST., MT.  
NIP: 196907301995121001



2. Dr. Supeno Mardi Susiki Nugroho, ST., MT.  
NIP: 197003131995121001



3. Dr. Diah Puspito Wulandari, ST., M.Sc.  
NIP: 198012192005012001



Kepala Departemen Teknik Elektro  
Fakultas Teknologi Elektro



Dr. Eng. Ardyono Priyadi, ST., M.Eng.  
NIP: 197309271998031004

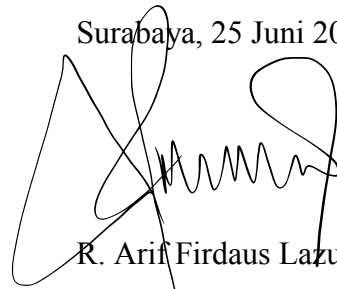
*Halaman ini sengaja dikosongkan*

## PERNYATAAN KEASLIAN TESIS

Dengan ini saya menyatakan bahwa isi keseluruhan Tesis saya dengan judul “**DETEKSI LOKASI TULANG MANUSIA PADA CITRA ULTRASOUND BERBASIS DEEP LEARNING MENGGUNAKAN CNN ARSITEKTUR YOLOv3**” adalah benar-benar hasil karya intelektual mandiri, diselesaikan tanpa menggunakan bahan-bahan yang tidak diijinkan dan bukan merupakan karya pihak lain yang saya akui sebagai karya sendiri.

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

Surabaya, 25 Juni 2019



R. Arif Firdaus Lazuardi

NRP. 07111650052003

*Halaman ini sengaja dikosongkan*



# **DETEKSI LOKASI TULANG MANUSIA PADA CITRA ULTRASOUND BERBASIS DEEP LEARNING MENGUNAKAN CNN ARSITEKTUR YOLOv3**

Nama mahasiswa : R. Arif Firdaus Lazuardi  
NRP : 07111650052003  
Pembimbing : 1. Prof. Dr. Ir. Mauridhi Hery Purnomo, M.Eng.  
2. Dr. Eko Mulyanto Yuliarno, ST., MT.

## **ABSTRAK**

Deteksi tulang manusia pada citra ultrasound memiliki tantangan yang cukup kompleks. Hal ini disebabkan, representasi pantulan gelombang suara yang dipancarkan oleh sensor B-scan USG tidak hanya menampilkan spesimen tulang saja, melainkan di dalamnya juga terdapat otot, jaringan lunak, dan bagian-bagian lain di bawah jaringan kulit. Oleh sebab itu dibutuhkan sebuah sistem yang dapat mengenali secara otomatis spesimen tulang pada citra USG. Penelitian ini mengimplementasikan pembelajaran sistem berbasis *deep learning* menggunakan metode *convolutional neural network* (CNN) dengan arsitektur YOLOv3. Hasil pelatihan sistem dengan *threshold* IoU 0.5 dapat mengenali objek tulang pada  $mAP_{50}$ ,  $mAP_{75}$ , dan  $mAP_{50:95}$  dengan masing-masing nilai sebesar 99.98, 97.68, dan 85.67. Dan untuk hasil pelatihan sistem dengan *threshold* IoU 0.75 dapat mengenali objek tulang pada  $mAP_{50}$ ,  $mAP_{75}$ , dan  $mAP_{50:95}$  dengan masing-masing nilai sebesar 99.96, 97.46, dan 86.35.

Kata kunci: USG tulang, CNN, YOLOv3

*Halaman ini sengaja dikosongkan*

# DETECTION OF HUMAN BONE LOCATION IN ULTRASOUND IMAGE BASED ON DEEP LEARNING USING YOLOv3 CNN ARCHITECTURE

By : R. Arif Firdaus Lazuardi  
Student Identity Number : 07111650052003  
Supervisor(s) : 1. Prof. Dr. Ir. Mauridhi Hery P, M.Eng.  
2. Dr. Eko Mulyanto Yuliarno, ST., MT.

## ABSTRACT

Detection of human bones in ultrasound images has quite complex challenges. This is due to the representation of reflections of sound waves emitted by the B-scan ultrasound sensors not only displaying bone specimens, but also muscle, soft tissue, and other parts under the skin tissue. Therefore we need a system that can automatically recognize bone specimens in ultrasound images. This study implements deep learning based systems using the convolutional neural network (CNN) method with the YOLOv3 architecture. The results of system training with IoU threshold 0.5 can recognize bone objects in  $mAP_{50}$ ,  $mAP_{75}$ , and  $mAP_{50:95}$  with values of 99.98, 97.68, and 85.67 respectively. And for the results of system training with IoU 0.75 threshold can recognize bone objects in  $mAP_{50}$ ,  $mAP_{75}$ , and  $mAP_{50:95}$  with values of 99.96, 97.46, and 86.35 respectively.

Key words: bone USG, CNN, YOLOv3

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Alhamdulillah, segala puja dan puji syukur kita haturkan kepada Allah SWT yang telah memberikan nikmat kepada penulis untuk bisa menyelesaikan tesis dengan judul “DETEKSI LOKASI TULANG MANUSIA PADA CITRA *ULTRASOUND* BERBASIS *DEEP LEARNING* MENGGUNAKAN CNN ARSITEKTUR YOLOv3.” Penulis menyadari bahwa banyak pihak yang telah membantu menyelesaikan penulisan tesis. Untuk itu, seiring dengan doa dan ucapan terima kasih yang sebesar-besarnya penulis sampaikan kepada:

1. Bapak Prof. Dr. Ir. Mauridhy Hery Purnomo, M.Eng dan Bapak Dr., Eko Mulyanto Yuniarno, S.T, M.T selaku dosen pembimbing yang senantiasa sabar memberikan bimbingan dan arahan dalam pengerjaan tesis kepada penulis.
2. Segenap dosen Program Magister Bidang Keahlian Jaringan Cerdas Multimedia Departemen Teknik Elektro ITS yang selalu ikhlas berbagi ilmu kepada para mahasiswa.
3. Manajemen PT. Infoglobal Teknologi Semesta yang telah memberikan kesempatan kepada penulis untuk melanjutkan studi.
4. Orang tua tersayang (Alm) Ayahanda R. Moch Sjarifudin Miftah, (Almh) Ibunda Wiwin Winarti, Ayahanda Abdul Madjid, dan Ibunda Anifah yang telah menanamkan nilai-nilai dan prinsip kehidupan.
5. Istri dan anak tercinta Zunif Ermayanti dan R. Azka Altruis Lazuardi yang selalu memberikan dukungan dan semangat di tiap kesempatan.
6. Kakanda R. Arief Perdana Kharisma dan istri Farida Amalia, Adinda Nurmutiara Lintang Sari, dan Adinda Putri Fitria serta Kemenakan R. Hasan Mevlana Azzaher yang senantiasa memberikan warna kehidupan.
7. Rekan mahasiswa Laboratorium Visikom serta rekan-rekan Program Magister Bidang Keahlian Jaringan Cerdas Multimedia, Teknik Elektro ITS.
8. Dan rekan kerja di PT. Infoglobal Teknologi Semesta.

Akhirnya atas segala kekurangan dari penyusunan tesis ini, sangat diharapkan kritik dan saran untuk dapat menutupi ketidaksempurnaan penulis dalam menyusun buku ini.

Surabaya, 25 Juni 2019

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN TESIS.....	iii
PERNYATAAN KEASLIAN TESIS.....	v
ABSTRAK.....	vii
ABSTRACT.....	ix
KATA PENGANTAR.....	xi
DAFTAR ISI.....	xiii
DAFTAR GAMBAR.....	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	3
1.3 Tujuan.....	4
1.4 Batasan Masalah.....	4
1.5 Kontribusi.....	5
BAB 2 KAJIAN PUSTAKA.....	7
2.1 Kajian Penelitian Terkait.....	7
2.2 <i>Convolutional Neural Network</i> (CNN).....	11
2.3 Arsitektur Yolo Versi 1 (YOLOv1).....	13
2.4 Arsitektur Yolo Versi 2 (YOLOv2).....	16
2.5 Arsitektur Yolo Versi 3 (YOLOv3).....	17
BAB 3 METODOLOGI PENELITIAN.....	19
3.1 Pengambilan Dataset.....	20
3.2 Anotasi Citra USG.....	21
3.3 Pelatihan Algoritma pada Sistem.....	23
3.4 Pengujian dan Evaluasi.....	25
BAB 4 HASIL DAN PEMBAHASAN.....	29
4.1 Anotasi Data Citra USG.....	29

4.2	Pelatihan Sistem .....	30
4.2.1	Pelatihan Sistem dengan <i>Threshold</i> IoU > 0.5 .....	32
4.2.2	Pelatihan Sistem dengan <i>Threshold</i> IoU > 0.75 .....	33
4.2.3	Analisis Proses Pelatihan dengan Berbagai Nilai <i>Threshold</i> IoU ...	34
4.3	Pengujian .....	37
4.3.1	Analisis <i>Metric Evaluation</i> .....	38
4.3.2	Analisis Proses Pengujian Secara Keseluruhan.....	42
4.3.3	Pengamatan Hasil Deteksi Secara Visual .....	43
4.4	Perbandingan Metode.....	48
4.4.1	Penambahan Dataset.....	48
4.4.2	Pelatihan Sistem Dataset Tambahan .....	50
4.4.3	Pengujian Dataset Tambahan .....	50
4.4.4	Pengamatan Visual .....	52
BAB 5 KESIMPULAN .....		57
DAFTAR PUSTAKA.....		59
BIODATA PENULIS.....		61



## DAFTAR GAMBAR

Gambar 1.1 Representasi Citra USG Tulang Lengan Manusia .....	2
Gambar 2.1 Perbandingan mAP YoloV1 dengan Arsitektur CNN yang Lain [1].	9
Gambar 2.2 <i>Receptive Fields</i> Le-Net [5] .....	12
Gambar 2.3 Arsitektur Umum CNN .....	13
Gambar 2.4 Model Sistem Deteksi Algoritma YoloV1 [1] .....	14
Gambar 2.5 Arsitektur Jaringan YoloV1 [1].....	15
Gambar 2.6 Prediksi Lokasi dan Dimensi <i>Bounding Box</i> [14] .....	17
Gambar 2.7 Arsitektur <i>Backbone</i> Darknet-53 [15] .....	18
Gambar 3.1 Blok Diagram Alur Deteksi Tulang Menggunakan CNN .....	19
Gambar 3.2 <i>Cropping</i> Citra USG .....	22
Gambar 3.3 Penentuan RoI Secara Manual .....	23
Gambar 3.4 Ilustrasi dari <i>Intersection over Union Bounding Box</i> .....	24
Gambar 4.1 Anotasi RoI Objek pada Citra USG .....	30
Gambar 4.2 Alur Data Pelatihan Sistem .....	31
Gambar 4.3 Grafik <i>Average Loss</i> dan Menggunakan <i>Threshold IoU&gt;0.5</i> .....	32
Gambar 4.4 Grafik <i>Average Loss</i> dan Menggunakan <i>Threshold IoU&gt;0.75</i> .....	34
Gambar 4.5. Perbandingan grafik <i>average loss</i> skenario P1 dan P2 .....	35
Gambar 4.6 Perbandingan grafik mAP skenario P1 dan P2 .....	36
Gambar 4.7 Blok Diagram Skenario Pengujian.....	38
Gambar 4.8 Nilai IoU Terendah Untuk Skenario Uji U1 .....	44
Gambar 4.9 Nilai IoU Terendah Untuk Skenario Uji U2 .....	45
Gambar 4.10 Nilai IoU Tertinggi Untuk Skenario Uji U1.....	46
Gambar 4.11 Nilai IoU Tertinggi Untuk Skenario Uji U2.....	47
Gambar 4.12 Bidang Pengambilan Data Citra USG.....	49
Gambar 4.13 Hasil Deteksi pada <i>Transverse Plane</i> Faster R-CNN .....	53
Gambar 4.14 Hasil Deteksi pada <i>Sagital Plane</i> Faster R-CNN.....	54
Gambar 4.15 Hasil Deteksi YoloV3 .....	55

*Halaman ini sengaja dikosongkan*

## DAFTAR TABEL

Tabel 4.1 Dataset Tulang Lengan Manusia .....	29
Tabel 4.2 Tabel Perbandingan <i>Average Loss</i> Tiap 600 <i>Batch</i> .....	36
Tabel 4.3 Perbandingan nilai mAP di pencuplikan beberapa batch.....	37
Tabel 4.4 Skenario Uji .....	38
Tabel 4.5 Nilai Parameter <i>precision<sub>bone</sub></i> untuk Semua Skenario Uji .....	39
Tabel 4.6 Nilai Parameter <i>Precision</i> untuk Semua Skenario Uji.....	40
Tabel 4.7 Nilai Parameter Average IoU untuk Semua Skenario Uji .....	41
Tabel 4.8 Nilai Parameter Average mAP untuk Semua Skenario Uji .....	41
Tabel 4.9 Pengujian dengan <i>mAP<sub>50</sub></i> .....	42
Tabel 4.10 Pengujian dengan <i>mAP<sub>75</sub></i> .....	43
Tabel 4.11 Pengujian dengan <i>mAP<sub>50:95</sub></i> .....	43
Tabel 4.12 Penambahan Dataset Untuk Perbandingan Metode .....	48
Tabel 4.13 Nilai mAP perbandingan YoloV3 dan Faster R-CNN.....	51
Tabel 4.14 Perbandingan mAP skenario uji <i>U3</i> dan <i>U4</i> semuar aristektur .....	52
Tabel 4.15 Jumlah Hasil Deteksi YoloV3 dan Faster R-CNN.....	53
Tabel 4.16 Jumlah deteksi Faster R-CNN dan YoloV3 berdasarkan nilai IoU ....	56

*Halaman ini sengaja dikosongkan*

# **BAB 1**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Teknologi kesehatan membantu para praktisi dunia kesehatan seperti dokter dan perawat dalam proses diagnosis, pengobatan, dan perawatan sebuah penyakit yang diderita oleh pasien. Semakin canggih teknologi kesehatan yang dimiliki oleh sebuah daerah/negara, maka pelayanan dalam bidang kesehatan pun semakin meningkat.

Dalam bidang kedokteran, teknologi kesehatan menjadi sebuah kebutuhan. Sebab dengan hadirnya teknologi tersebut, dokter dapat mengambil sebuah keputusan yang tepat dan efisien dalam menangani sebuah permasalahan. Di sisi pasien pun, kehadiran teknologi kesehatan membantu mereka untuk memahami penyakit yang diderita.

Beberapa diantara teknologi kesehatan menggunakan teknik pencitraan medis seperti *Computed Tomography (CT-scan)*, *Magnetic Resonance Imaging (MRI)*, dan *Ultrasonography (USG)*. Citra medis menjadi sebuah alat ukur yang tepat untuk merepresentasikan bentuk dan dimensi dari struktur tubuh dan organ bagian dalam manusia. Dimana dengan representasi yang presisi dari citra medis tersebut, dokter dapat mengukur dengan tepat dalam mendiagnosis sebuah permasalahan pun dalam memberikan solusi atas permasalahan tersebut.

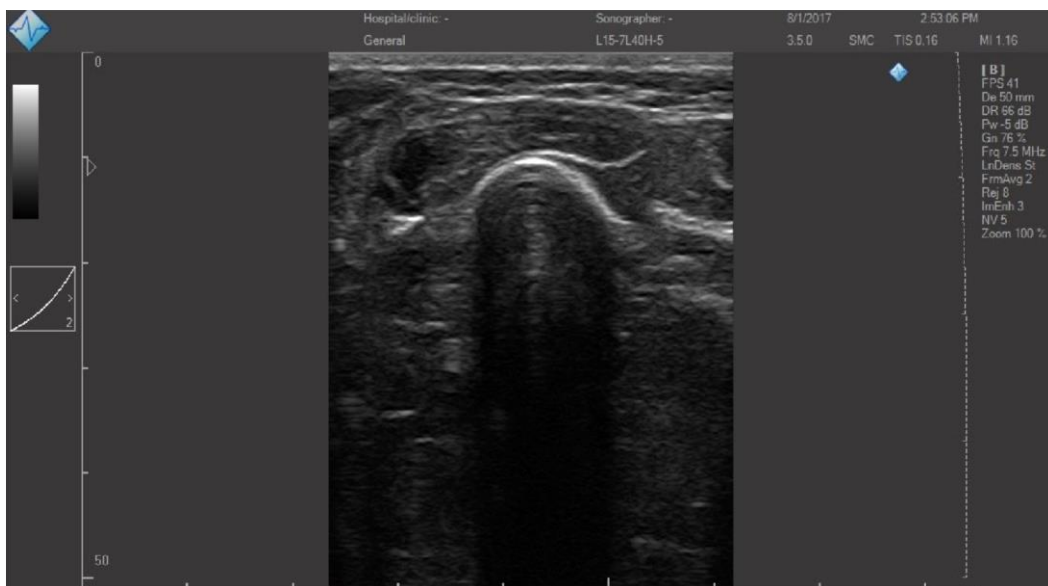
MRI dan *CT-scan* merupakan teknik pencitraan medis yang paling populer dalam dunia kesehatan. Hal ini dikarenakan hasil representasi citra dari kedua alat teknologi tersebut menghasilkan kualitas citra yang sangat baik. Akan tetapi ada beberapa kekurangan dari penggunaan kedua teknologi kesehatan tersebut.

*CT-scan* memproduksi banyak ion dan radiasi kepada pasien, dan MRI merupakan sebuah teknologi yang mahal dan membutuhkan waktu yang lama dalam penggunaannya. Dibandingkan dengan kedua teknologi tersebut, USG menjadi sebuah alat yang paling banyak digunakan untuk merepresentasikan citra

jaringan lunak, akan tetapi citra USG juga berpotensi untuk digunakan merepresentasikan struktur keras seperti tulang .

Hal itu dikarenakan teknologi USG aman digunakan, tidak invasif, berbasis gelombang suara, relatif lebih murah, menyajikan data secara *real-time*, dan mudah untuk dibawa berpindah lokasi (*portable*) . USG banyak digunakan di beberapa fasilitas kesehatan. Akan tetapi kekurangan dari teknologi USG ini adalah representasi citra yang dihasilkan kualitasnya lebih rendah dibandingkan dengan teknologi *CT-scan* atau *MRI*.

Penelitian terkait teknik rekonstruksi citra 3D yang dibangun dari representasi citra 2D dari USG semakin banyak berkembang. Ada beberapa cara untuk merekonstruksi citra 3D tersebut. Diantaranya adalah dengan mengambil beberapa serangkaian citra 2D representasi dari B-scans USG. Pada Gambar 1.1 merupakan representasi citra 2D dari tulang lengan, diambil menggunakan sensor B-scans USG.



Gambar 1.1 Representasi Citra USG Tulang Lengan Manusia

Teknik pengambilan serangkaian citra tersebut pun memiliki banyak metode. Yang paling sering digunakan adalah menggunakan teknik *freehand* (

tangan bebas) dengan sensor posisi yang dipasang pada sensor USG, menggunakan elektromagnetik, dan/ataupun sistem optik.

Dalam membangun sistem *freehand 3D ultrasound* yang dapat merepresentasikan bentuk 3D dari citra 2D sensor B-scans USG. Pada umumnya, tahapan yang dilakukan adalah sebagai berikut: 1) pengambilan data serangkaian citra medis, 2) segmentasi citra medis, memisahkan antara objek yang dikehendaki dengan latar belakang citra, 3) registrasi citra 2D ke 3D, 4) dan rekonstruksi citra 3D.

Keempat tahapan tersebut sangatlah penting untuk dilakukan dengan ukuran yang tepat. Hal ini disebabkan, bentuk rekonstruksi citra pada bidang 3D diharapkan merepresentasikan dengan tepat bentuk dan ukuran dari objek pada organ tubuh manusia.

Dan untuk dapat melakukan segmentasi secara optimal, sistem diharapkan dapat mendeteksi *region of interest* (RoI) tulang pada citra USG. Sehingga bagian-bagian lain yang direpresentasikan oleh sensor USG, semisal jaringan lunak, pembuluh darah, otot, kulit dsb dapat dianggap sebagai noise dan dihilangkan. Sehingga proses segmentasi representasi tulang pada citra USG lebih efisien dan optimal.

Penelitian ini lebih fokus pada teknik deteksi objek citra medis. Hal ini dikarenakan, hasil dari deteksi tersebut sangat berguna agar sistem dalam melakukan proses segmentasi citra medis. Hasil dari segmentasi citra tersebut memiliki pengaruh yang sangat besar dalam merekonstruksi citra 3D. Semakin bagus kualitas segmentasi citra medis tersebut, maka hasil rekonstruksi citra 3D pun juga semakin tepat, baik dari segi bentuk maupun ukuran.

## **1.2 Rumusan Masalah**

Representasi citra medis menggunakan sensor USG untuk tulang terdapat juga tangkapan objek lain selain tulang. Seperti; daging, pembuluh darah, dan jaringan lunak. Sehingga permasalahan utama pada penelitian ini adalah melakukan deteksi area tulang pada citra USG.

Algoritma yang hendak digunakan untuk melakukan deteksi lokasi tulang berbasis *Deep Learning* menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur jaringan *You Only Look Once* (YOLO). Algoritma tersebut digunakan dengan harapan sistem dapat beradaptasi dengan representasi citra yang berbeda-beda dari setiap subjek.

### **1.3 Tujuan**

Tujuan dari penelitian ini adalah membangun sebuah sistem yang dapat melakukan deteksi tulang dari representasi citra 2D sensor *B-scans* USG. Beberapa hal yang hendak dicapai dalam penelitian ini adalah sebagai berikut.

1. Melakukan pelatihan terhadap sistem menggunakan data pelatihan citra medis USG berbasis algoritma *deep learning* menggunakan metode CNN dengan arsitektur YOLO.
2. Dengan menggunakan parameter hasil dari pelatihan sistem, digunakan untuk mendeteksi lokasi objek tulang pada citra ultrasonografi.

### **1.4 Batasan Masalah**

Ada beberapa batasan masalah terkait penelitian yang hendak dilakukan sebagai berikut.

1. Representasi citra tulang manusia didapatkan dari sensor *B-scans* USG.
2. Bagian tulang manusia yang di-*scanning* adalah bagian tulang lengan bawah.
3. Algoritma yang digunakan untuk melakukan deteksi menggunakan metode CNN dengan arsitektur jaringan YOLO.

Representasi citra 2-D dari sensor *B-scan* diambil terlebih dahulu dan disimpan dalam media penyimpanan untuk selanjutnya diproses dengan algoritma yang dikembangkan.



## **1.5 Kontribusi**

Beberapa keuntungan yang didapatkan dari penelitian ini adalah hasil deteksi area tulang pada citra USG. Dapat membantu agar sistem dapat melakukan pengolahan informasi selanjutnya, yaitu segmentasi objek tulang pada citra 2D USG. Semakin baik hasil segmentasi, maka hasil rekonstruksi citra 3D pun juga semakin mendekati bentuk dan ukuran asli dari organ tulang panjang manusia.

*Halaman ini sengaja dikosongkan*

## **BAB 2**

### **KAJIAN PUSTAKA**

Pada bab ini akan dijelaskan beberapa metode yang digunakan untuk melakukan segmentasi citra ultrasonografi berbasis *deep learning*. Penjelasan metode yang digunakan akan dibagi menjadi 2 sub bab.

Sub bab pertama menjelaskan tentang penelitian sebelumnya yang terkait dengan penelitian ini. Sub bab kedua menjelaskan tentang algoritma *deep learning* yang digunakan untuk mengenali area objek yang hendak dipisah dari latar belakangnya. Dan sub bab ketiga menjelaskan tentang algoritma pengolahan citra digital untuk memisahkan objek dengan latar belakang.

Pada sub bab kedua: algoritma *deep learning* yang digunakan adalah *Convolutional Neural Network* (CNN). CNN merupakan salah satu metode *Deep Learning* yang digunakan untuk memberikan pembelajaran kepada mesin terkait fitur objek yang hendak dikenali. Arsitektur CNN yang digunakan pada penelitian ini adalah arsitektur *CNN You Only Look Once* (YOLO) yang dikembangkan oleh Joseph Redmon, Santosh Divala, Ross Girshick, dan Ali Farhadi [1]. Kelebihan dari arsitektur ini adalah mengklasifikasi objek dan mengenali objek tersebut beserta lokasinya dalam satu tahapan jaringan arsitektur CNN.

#### **2.1 Kajian Penelitian Terkait**

Penelitian yang berhubungan dengan pencitraan medis telah banyak dilakukan, sebagian diantaranya menggunakan algoritma pengolahan citra digital seperti yang dilakukan oleh R. Jia untuk melakukan segmentasi tulang secara otomatis pada citra USG menggunakan *Local Phase Features* [11]. Penelitian juga dilakukan oleh J. Kowal untuk melakukan segmentasi tulang otomatis untuk meminimalkan resiko tindakan medis berbasis komputer [9].

Sedangkan penggunaan algoritma berbasis *deep learning* juga sudah dilakukan oleh beberapa peneliti. Diantaranya adalah untuk melakukan deteksi

kanker serviks [7], segmentasi citra smear serviks [16], mendeteksi penyakit tumor hati [17], dan klasifikasi citra termal kanker payudara [18].

Dalam beberapa tahun terakhir, metode CNN banyak mengalami perkembangan, arsitektur yang sering digunakan adalah Faster R-CNN yang diajukan oleh S. Ren dkk [12]. Arsitektur ini menggunakan pendekatan dengan melakukan pengajuan beberapa region yang diasumsikan sebagai objek, yang sering diistilahkan sebagai *region proposal network* (RPN). Kemudian dari beberapa proposal region objek tersebut diklasifikasi dengan *network classifier* untuk menentukan *score class prediction* di tiap *region of interest* yang dideteksi.

*Network detector* Faster R-CNN membutuhkan dua tahapan untuk dapat mendeteksi dan mengklasifikasi objek pada suatu citra. Tahap pertama untuk menentukan beberapa RoI yang diasumsikan terdapat suatu objek. Dan tahap kedua dilakukan klasifikasi tingkat probabilitas suatu objek.

Hal tersebut membuat Joseph Redmon dkk mengajukan pendekatan baru dalam membuat arsitektur jaringan deteksi objek yang hanya mengalami satu tahapan *network detector*. Yang disebut dengan arsitektur YOLO (*You Only Look Once*) [1] pada tahun 2016. Arsitektur YOLO ini menggunakan algoritma dasar *Regression Problem* untuk membuat suatu jaringan syaraf yang dalam satu kali tahapan dapat mendeteksi objek sekaligus mengenalinya.

Metode YOLO sangat cocok diterapkan untuk sistem yang bekerja secara *real-time*, karena pada CPU yang memiliki komputasi *Video Card* NVIDIA Titan X dapat memproses hingga 45 frame di setiap detiknya (45 *fps*). Metode ini lebih cepat memproses untuk mengenali objek dibandingkan dengan metode sebelumnya yaitu DPM dan R-CNN.

Pada Gambar 2.1 di bawah ini menunjukkan hasil perbandingan antara metode YOLO dengan metode serupa lainnya, dimana hasil dari percobaan pada penelitian yang dilakukan oleh Joseph Redmon dkk lebih baik dibandingkan dengan metode sebelumnya.

Real-Time Detectors	Train	mAP	FPS
100Hz DPM [30]	2007	16.0	100
30Hz DPM [30]	2007	26.1	30
Fast YOLO	2007+2012	52.7	<b>155</b>
YOLO	2007+2012	<b>63.4</b>	45
Less Than Real-Time			
Fastest DPM [37]	2007	30.4	15
R-CNN Minus R [20]	2007	53.5	6
Fast R-CNN [14]	2007+2012	70.0	0.5
Faster R-CNN VGG-16[27]	2007+2012	73.2	7
Faster R-CNN ZF [27]	2007+2012	62.1	18
YOLO VGG-16	2007+2012	66.4	21

Gambar 2.1 Perbandingan mAP YoloV1 dengan Arsitektur CNN yang Lain [1]

Tidak berselang lama, di akhir tahun 2016, Joseph Redmon mengajukan pembaruan arsitektur di YOLOv1, dengan mengganti layer terakhir yang digunakan untuk melakukan deteksi. Dimana sebelumnya menggunakan *fully connected layer* dan hanya bisa memberikan nilai tertinggi suatu *class* di tiap objek. Menjadi layer konvolusi yang menghasilkan beberapa *box detector* di tiap gridnya dan bisa memberikan alternatif multi label di tiap objek yang terdeteksi pada suatu citra. Arsitektur yang diperbarui ini diberi istilah YOLOv2 [15].

Pada arsitektur YOLOv2 terdapat kelemahan tidak dapat mendeteksi dengan baik objek yang memiliki ukuran kecil. Oleh karena itu pada tahun 2018. Memberikan alternatif arsitektur yang baru YOLOv3 [16], yang dapat mengatasi permasalahan tersebut. Pembaruan yang dilakukan pada YOLOv3 ini adalah dengan menambahkan tahapan *feature pyramid network* (FPN) untuk memberikan kesempatan jaringan melakukan perubahan skala pada citra masukkan hingga 3 kali skala. Sehingga pembaruan arsitektur ini memunculkan *backbone network detector* yang baru darket53 untuk mengakomodasi tahapan FPN.

Pada bidang citra medis, CNN telah digunakan oleh You Li [3] dalam penelitiannya yang berjudul berjudul, “*Segmentation of Medical Ultrasound Images Using Convolutional Neural Networks with Noisy Activating Functions*,” untuk melakukan segmentasi citra medis berbasis USG. Penelitian You Li

menggunakan algoritma CNN arsitektur *U-Net-Like* sebagai backbone. Sedangkan untuk fungsi aktivasi yang digunakan adalah *Noisy Rectified Linear Unit* (NRelu), *Noisy Hard Sigmoid* (NHSigmoid), *Noisy Hard Tanh* (NHTanh).

*Dataset* yang digunakan menggunakan *dataset* publik yang terdapat pada *Kaggle Challenges* (Kaggle, 2016). *Dataset* tersebut berisi sekumpulan citra medis leher manusia. Penelitiannya bertujuan untuk melakukan segmentasi sekumpulan citra syaraf *Brachial Plexus* (BP). Pada gambar 2.2 adalah representasi citra USG leher manusia beserta segmentasi manual area syaraf BP.

Hasil penelitian ini adalah algoritma CNN dengan fungsi aktivasi *Noisy* dapat melakukan segmentasi area sekumpulan syaraf BP. Peneliti menyebutkan bahwa kelemahan dalam penelitiannya adalah akibat inkonsistensi segmentasi manual yang dilakukan oleh para relawan (*ground truth segmentation*) sebelumnya. Menyebabkan hasil segmentasi sesuai dengan pelatihan yang diberikan dari data *ground truth segmentation*. Diharapkan validitas area syaraf BP pada segmentasi manual dapat dipertanggungjawabkan keabsahannya.

Di tahun 2016, K. Tita dkk memulai penelitian untuk dapat merekonstruksi citra 3D spesimen tulang panjang manusia pada citra USG. Penelitian dimulai dengan melakukan deteksi otomatis objek tulang pada citra USG menggunakan algoritma berbasis pengolahan citra digital [2]. Penelitian tersebut berjudul, “*Automatic Bone Outer Contour Extraction from B-Modes Ultrasound Images Based on Local Phase Symmetry and Quadratic Polynomial Fitting*”.

Penelitian ini dilakukan untuk mengekstraksi fitur tulang pada citra ultrasonografi. Algoritma *Local Phase Symmetry* (LPS) digunakan untuk memisahkan objek tulang dengan latar belakangnya. Sedangkan algoritma *Quadratic Polynomial Fitting* ( QPC) digunakan untuk mengisi kembali beberapa piksel yang hilang saat proses LPS, dimana piksel yang hilang tersebut merupakan bagian dari objek tulang. Objek tulang panjang yang digunakan pada penelitian ini adalah tulang sapi dan kambing yang ditaruh di dalam kotak berisi air, kemudian sensor B-Scane USG berjalan secara bertahap untuk pengambilan data citra USG.

Hasil dari algoritma cukup bagus karena spesimen tulang yang digunakan adalah tulang hewan yang ditempatkan pada suatu alat khusus untuk

mempermudah peneliti dalam mengambil dataset citra tulang menggunakan sensor USG.

Kemudian di tahun 2018, K. Tita melakukan pendekatan baru untuk segmentasi tulang pada citra USG berbasis algoritma *deep learning*. Metode yang digunakan adalah CNN dengan arsitektur Faster R-CNN [12]. Penelitiannya mampu menentukan RoI tulang secara otomatis pada citra USG tulang manusia. Hal ini cukup memberikan bantuan agar proses segmentasi tulang lebih mudah, karena deteksi berbasis *deep learning* ini mampu menentukan RoI tulang dengan tepat. Sehingga bagian lain dari citra USG selain tulang dapat direduksi.

## 2.2 Convolutional Neural Network (CNN)

CNN merupakan salah satu algoritma *deep learning*, dikarenakan menggunakan banyak layer dalam satu jaringan syaraf tiruan untuk mengenali dan/ataupun mengklasifikasi sebuah objek. Arsitektur dasar dari algoritma CNN ini adalah *multilayer perceptron*, akan tetapi menggunakan konsep konvolusi pada citra, sehingga parameter yang dibutuhkan untuk satu jaringan semakin sedikit dibandingkan jika menggunakan arsitektur full *multilayer perceptron*.

Pada umumnya algoritma CNN terdiri dari 3 tensor input yang merupakan citra RGB. Kemudian input tersebut mengalami beberapa proses perhitungan. Tiap hasil proses perhitungan disebut sebagai layer. Dimana layer-layer tersebut dapat berupa; *convolution layer*, *pooling layer*, *normalization layer*, *fully connected layer*, dan *loss layer* [4].

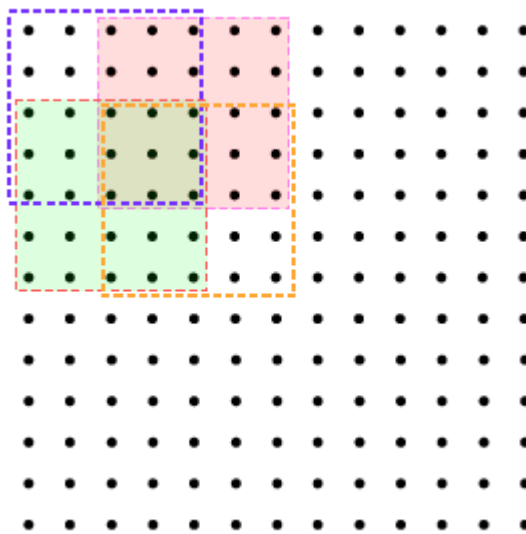
Berikut adalah model abstrak dari struktur CNN [4].

$$x^1 \rightarrow [w^1] \rightarrow x^2 \rightarrow \dots \rightarrow x^{L-1} \rightarrow [w^{L-1}] \rightarrow x^L \rightarrow [w^L] \rightarrow z \quad (2.1)$$

Dimana  $x^1$  merupakan citra RGB (tensor orde 3), kemudian diproses dengan layer pertama yang direpresentasikan dengan kurung siku. Proses secara keseluruhan pada kurung siku dinotasikan dengan  $w^1$ . Sehingga keluaran dari proses tersebut adalah  $x^2$ , yang juga merupakan input untuk proses berikutnya. Proses ini berulang hingga layer yang terakhir.

Proses pelatihan algoritma CNN dapat menggunakan metode *backpropagation*, sehingga ada 2 tahapan dalam setiap 1 kali *loop* pelatihan; *forward run* dan *back-pass*. Dimana untuk melakukan update parameter dalam tahap *back-pass* menggunakan *Stochastic Gradient Descent* (SGD).

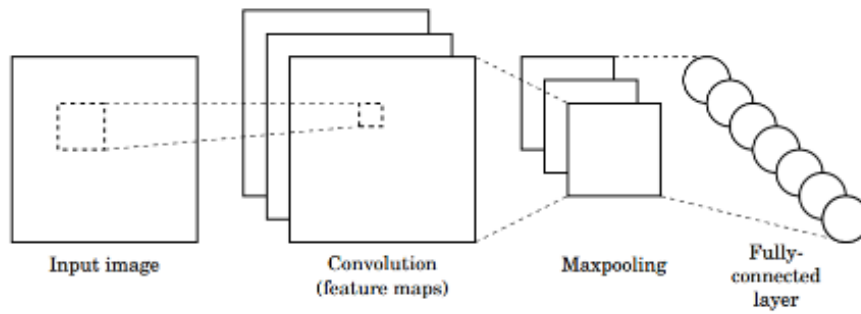
Layer yang paling utama dalam CNN adalah *convolutional layer*, pada arsitektur Le-Net 5, ukuran *weight* yang digunakan untuk konvolusi adalah 5x5 dan 13x13 [5]. Pada gambar 2.2 diilustrasikan layer konvolusi arsitektur Le-Net 5 (gambar diambil dari internet).



Gambar 2.2 *Receptive Fields* Le-Net [5]

Secara keseluruhan arsitektur algoritma CNN diilustrasikan pada Gambar 2.3, dimana input citra melalui beberapa layer sebagai berikut: konvolusi, *sub sampling*, dan *fully connected*.





Gambar 2.3 Arsitektur Umum CNN

### 2.3 Arsitektur Yolo Versi 1 (YOLOv1)

Kelebihan dari arsitektur YOLO dibandingkan dengan arsitektur CNN sejenis lainnya adalah dalam melakukan pelatihan data maupun pengujian data hanya melalui satu tahapan di blok jaringan syaraf tiruan. Hal ini membuat komputasi algoritma Yolo lebih cepat dibandingkan arsitektur yang lain. Arsitektur Yolo memodifikasi dari arsitektur CNN Google Net, dimana layer inception pada Google Net diubah menjadi layer konvolusi berukuran 1 x 1.

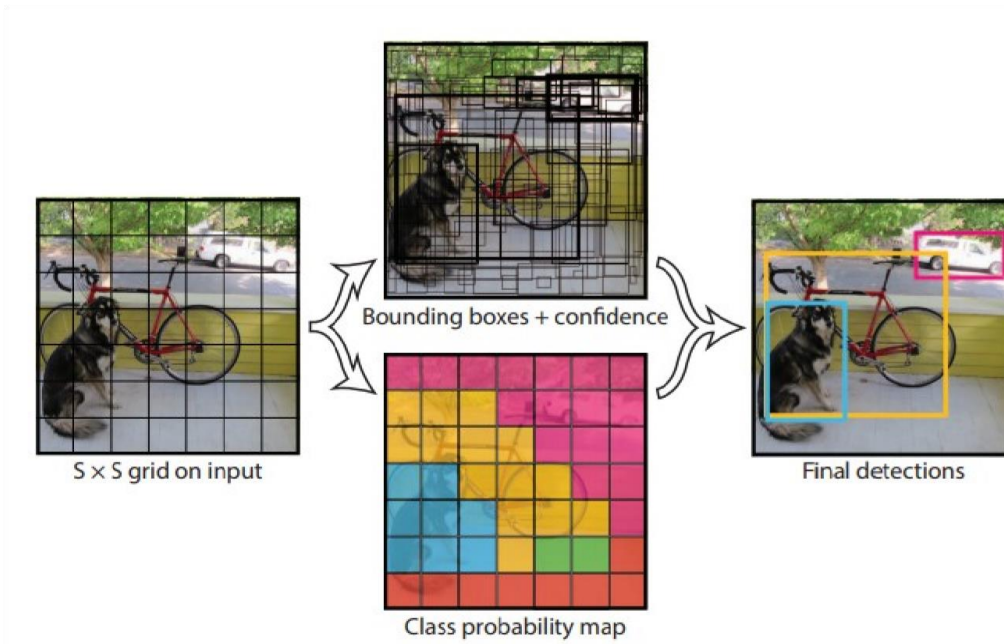
Model deteksi arsitektur Yolo adalah *regression problem*, dimana pada citra dibagi menjadi beberapa blok grid. Dan di setiap gridnya memprediksi nilai ukuran dan posisi objek, serta memiliki tingkat kepercayaan hasil deteksi objek pada grid tersebut.

Prediksi menggunakan arsitektur Yolo dapat pula direpresentasikan sebagai tensor dengan ukuran  $S*S*(B*5+C)$ , dimana  $S$  adalah jumlah grid bilangan positif 1 s.d maksimal jumlah baris pada citra,  $B$  adalah jumlah *bounding box* yang dikehendaki di setiap grid,  $C$  adalah jumlah jenis objek yang dijadikan target dalam satu pelatihan, dan bilangan 5 didapatkan dari nilai parameter pada *bounding box* yaitu: nilai posisi objek ( $x,y$ ) dan panjang serta lebar objek ( $w,h$ ) serta nilai *confidence* objek.

Untuk mendapatkan nilai *confidence* tiap objek, YoloV1 [1] menggunakan persamaan sebagai berikut,

$$Pr(Class_i | Object) * Pr(Object) * IOU_{pred}^{truth} = Pr(Class_i) * IOU_{pred}^{truth} \quad (2.2)$$

Dimana  $Pr(Object)$  probabilitas keberadaan objek,  $Pr(Class_i)$  adalah probabilitas *class* ke- $i$  ( $i$  merupakan indeks *class*) dan  $IOU_{pred}^{truth}$  adalah *intersection over union* antara *box* prediksi dengan *box* trutuh. Pada Gambar 2.4 merupakan ilustrasi klasifikasi objek menggunakan algoritma YoloV1.

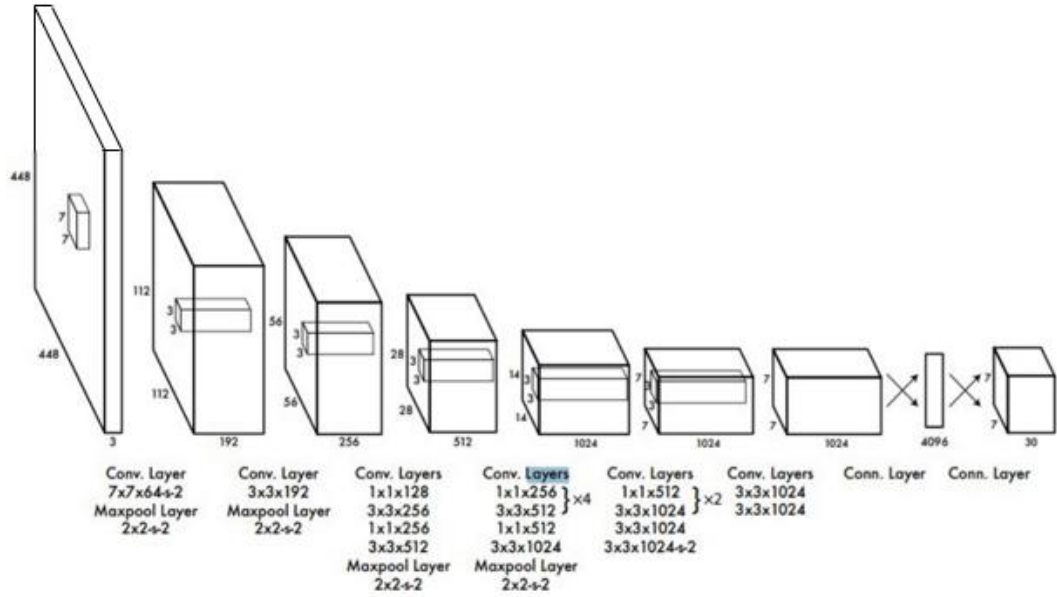


Gambar 2.4 Model Sistem Deteksi Algoritma YoloV1 [1]

Arsitektur YoloV1 menggunakan 24 layer konvolusi yang diikuti dengan 2 layer *fully connected* di akhir layer. Dan di beberapa layer konvolusi diikuti oleh layer pooling yang berjumlah 4 layer. Hasil akhir dari layer CNN adalah tensor prediksi yang berjumlah  $S*S*(B*5+C)$ . Setiap *box* memiliki 5 nilai, yakni nilai *bounding box* area yang terdeteksi dan probabilitas *class*. Pada gambar 2.5 merupakan arsitektur algoritma YoloV1.

Layer konvolusi YoloV1 diaktivasi menggunakan fungsi *leaky rectified linier*, sedangkan di layer terakhir YoloV1 diaktivasi menggunakan *linear function*,

$$\phi(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.1x, & \text{otherwise} \end{cases} \quad (2.3)$$



Gambar 2.5 Arsitekur Jaringan YoloV1 [1]

Di setiap iterasi pelatihan, perhitungan *loss function* yang digunakan adalah sebagai berikut,

$$\begin{aligned}
 & \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(x_1 - \hat{x}_1)^2 + (y_1 - \hat{y}_1)^2] \\
 & + \lambda_{coord} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(\sqrt{w_i} - \sqrt{\hat{w}_i})^2 + (\sqrt{h_i} - \sqrt{\hat{h}_i})^2] \\
 & + \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(C_1 - \hat{C}_1)^2] + \lambda_{noobj} \sum_{i=0}^{S^2} \sum_{j=0}^B \mathbb{I}_{ij}^{obj} [(C_1 - \hat{C}_1)^2] \\
 & + \sum_{i=0}^{S^2} \mathbb{I}_i^{obj} \sum_{C \in classes} (p_i(C) - \hat{p}_i(C))^2
 \end{aligned}$$

Dimana  $\lambda_{coord} = 5$ ,  $\lambda_{noobj} = .5$ ,  $(x,y)$  adalah titik tengah *box*,  $(w,h)$  adalah panjang lebar *box*,  $\mathbb{I}_i^{obj}$  menunjukkan objek ditemukan di cell  $i$ , dan  $\mathbb{I}_{ij}^{obj}$

menunjukkan *bounding box* ke  $j$  di cell  $i$  adalah nilai yang digunakan untuk memprediksi objek tersebut.

## 2.4 Arsitektur Yolo Versi 2 (YOLOv2)

Di arsitektur YOLOv2 terdapat beberapa peningkatan kemampuan arsitektur yang cukup signifikan dan perbedaan susunan layer konvolusi dan layer terakhir yang digunakan untuk memprediksi objek. Beberapa peningkatan kemampuan deteksi pada arsitektur YOLOv2 ini adalah adanya penambahan fitur *batch normalization*, penggunaan *high resolution classifier* hingga piksel berukuran 224x224 untuk tahap pelatihan dan 448 untuk tahap deteksi.

Juga penggunaan layer konvolusi dengan *anchor boxes* yang digunakan untuk memprediksi *bbox* dan objek seperti arsitektur *faster R-CNN*. Sehingga pada arsitektur YOLOv2 ini tidak menggunakan *fully connected* layer untuk memprediksi *bbox*. Perubahan ini dikarenakan isu ketidakstabilan memprediksi nilai tengah *bbox* ( $x,y$ ) pada arsitektur YoloV1. Pada *region proposal network* (RPN) di arsitektur *faster R-CNN* memprediksi nilai  $t_x$ ,  $t_y$ , dan koordinat titik tengah ( $x,y$ ), dihitung menggunakan persamaan berikut,

$$x = (t_x * W_a) - x_a ; y = (t_y * h_a) - y_a \quad (2.5)$$

Dengan  $t_x$  dan  $t_y$  adalah nilai offset *bbox*, sehingga  $t_x = 1$  memiliki arti geser ke kanan sejumlah lebar dari anchor box dan  $t_x = -1$  memiliki arti geser ke kiri sejumlah lebar dari anchor box.

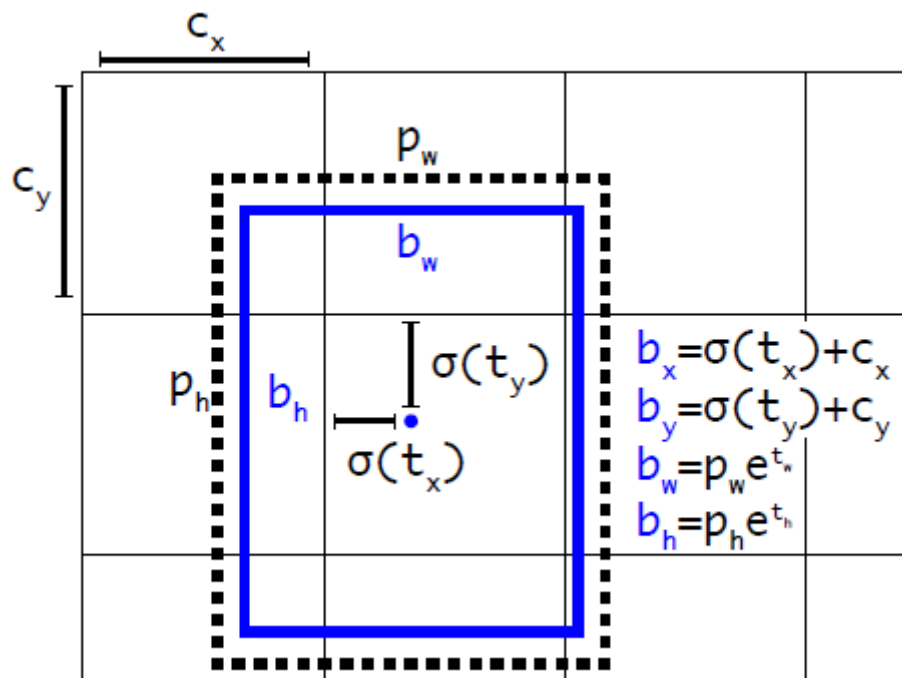
Persamaan 2.5 digunakan oleh YoloV2 dengan modifikasi memperdiksi 5 nilai koordinat untuk setiap *bounding box* ( $t_x, t_y, t_w, t_{yh}$ ) dan  $t_o$ . Jika tiap *cell* adalah offset dari kiri atas pojok citra dinotasikan dengan ( $c_x, c_y$ ) dan *bbox prior* memiliki lebar dan tinggi ( $P_w, P_h$ ), maka prediksi *bbox* didapatkan dengan cara berikut,

$$b_x = \sigma(t_x) + c_x \quad ; \quad b_y = \sigma(t_y) + c_y \quad (2.6)$$

$$b_w = P_w e^{t_w} \quad ; \quad b_h = P_h e^{t_h} \quad (2.7)$$

$$P_r(\text{object}) * IOU(b, \text{object}) = \sigma(t_o) \quad (2.8)$$

Persamaan 2.6, 2.7, dan 2.8 diilustrasikan pada Gambar 2.6 untuk memprediksi lokasi dan dimensi *bounding box*. Prediksi lebar dan tinggi ( $w, h$ ) *bounding box* sebagai offset dari nilai tengah ( $x, y$ ). Prediksi nilai tengah didapatkan dari perhitungan aktivasi fungsi sigmoid.



Gambar 2.6 Prediksi Lokasi dan Dimensi *Bounding Box* [14]

## 2.5 Arsitektur Yolo Versi 3 (YOLOv3)

Arsitektur YoloV3 tidak terlalu berbeda dengan YoloV2 dalam hal layer yang digunakan untuk memprediksi *bounding box* dan objek. Akan tetapi perbedaan yang signifikan terjadi adalah penggunaan arsitektur *backbone* Yolo, yang sebelumnya menggunakan *backbone* darknet 19 layer, pada YOLOv3 menggunakan *backbone* baru darknet 53 layer. Perubahan arsitektur ini mengakomodasi kelemahan YoloV2 yang kurang optimal untuk mendeteksi objek yang berukuran kecil.

Sehingga pada YoloV3 ini mengakomodasi metode *feature pyramid network* (FPN) yang umum digunakan pada arsitektur 2 tahap seperti *faster R-CNN* dan metode *sliding window*. Penambahan layer FPN ini menjadikan YOLOv3 melakukan *scaling* citra sebanyak 3 kali, agar *network* dapat memprediksi benda dari berbagai ukuran.

Adapun arsitektur jaringan darknet-53 ditampilkan dalam Gambar 2.7,

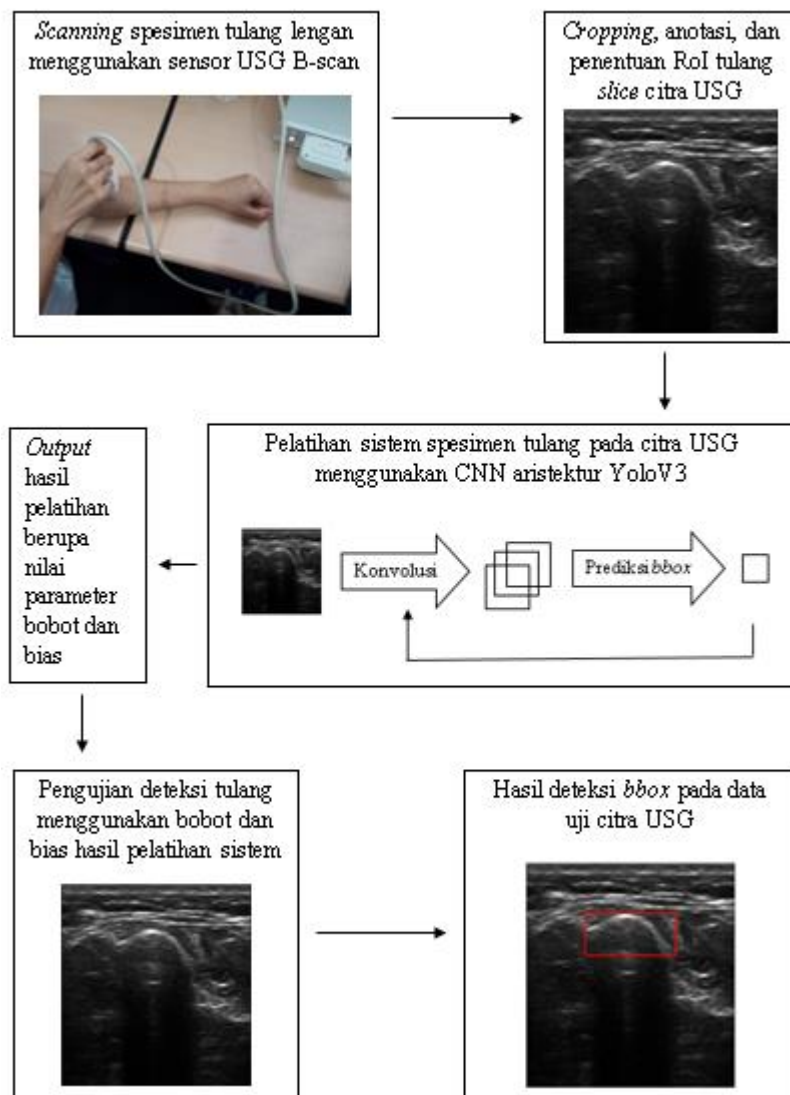
	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1x	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2x	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8x	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8x	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4x	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Gambar 2.7 Arsitektur *Backbone* Darknet-53 [15]

### BAB 3

## METODOLOGI PENELITIAN

Pada bab ini akan dijelaskan beberapa tahapan yang akan dilakukan pada penelitian ini. Penelitian dibagi menjadi empat tahap, tahap pertama adalah akuisisi data, tahap kedua pemberian label (anotasi) pada citra, tahap ketiga pelatihan data kepada sistem menggunakan CNN dengan arsitektur jaringan YOLO, dan tahap keempat adalah pengujian algoritma. Blok diagram metodologi penelitian disajikan pada Gambar 3.1.



Gambar 3.1 Blok Diagram Alur Deteksi Tulang Menggunakan CNN

Tahap pertama merupakan tahapan untuk mengambil citra USG spesimen tulang panjang manusia. Pengambilan data dilakukan dengan metode *free hand scanning* dari beberapa subjek. Di tahap yang kedua, data yang sudah diambil disimpan dalam bentuk citra, kemudian diberi label terkait posisi area tulang (penentuan RoI) untuk dijadikan parameter dalam melakukan pelatihan algoritma pada sistem di tahap ketiga. Hasil pelatihan merupakan sekumpulan parameter bobot dan bias (*weights*) yang digunakan untuk pengujian pada tahap berikutnya.

Secara detail dan terperinci penjelasan dari keempat tahapan di atas adalah sebagai berikut:

### 3.1 Pengambilan Dataset

Data yang diambil adalah data serangkaian citra B-scan USG pada subjek yang berbeda-beda. Alat yang digunakan untuk mengambil data citra ultrasonografi tulang panjang manusia adalah sensor B-Scan *TELEMED Linear Transducer L15-7L40H-5* dengan jangkauan frekuensi 7,0-15,0 MHz dan 39 mm *field of view*. Teknik pengambilan citra USG menggunakan *freehand scanning*. Dimana operator mengambil data citra USG dengan cara menggerakkan probe USG dengan tangan secara bebas. Akan tetapi untuk menunjang penelitian ini agar citra yang didapat dapat diolah sebagaimana mestinya. Operator diusahakan untuk selalu memposisikan probe sensor USG tegak lurus dengan objek yang hendak diambil representasi datanya.

Dalam penelitian ini, data yang dibutuhkan adalah spesimen tulang panjang manusia. Sehingga diambil data citra tulang lengan manusia, dengan batas dari bagian tengah lengan hingga ke ujung lengan sebelum telapak tangan. Pada area tersebut terdapat dua sisi area tulang panjang manusia yang disebut dengan *ulna* dan *radius*. Karena keterbatasan area *scanning* probe sensor USG, maka representasi citra tidak dapat menjangkau semuanya. Karena hasil citra USG hanya menampilkan representasi gelombang suara yang dipantulkan. Oleh sebab itu di beberapa pengambilan spesimen data, diambil di kedua sisi area tersebut agar keragaman citra USG tulang panjang manusia bisa didapatkan sebanyak mungkin.



Konfigurasi sensor USG di tiap subjek berbeda. Sehingga setiap pengambilan data, operator harus menyesuaikan dulu konfigurasi sensor USG nya, sehingga representasi citra tulang panjang secara kasat mata dapat dikenali. Hal ini menjadi penting, sebab algoritma *deep learning* yang digunakan pada penelitian ini adalah *supervised learning*. Dalam arti, algoritma ini membutuhkan pengetahuan manusia untuk menerjemahkan informasi secara benar dari representasi tulang panjang manusia pada citra USG.

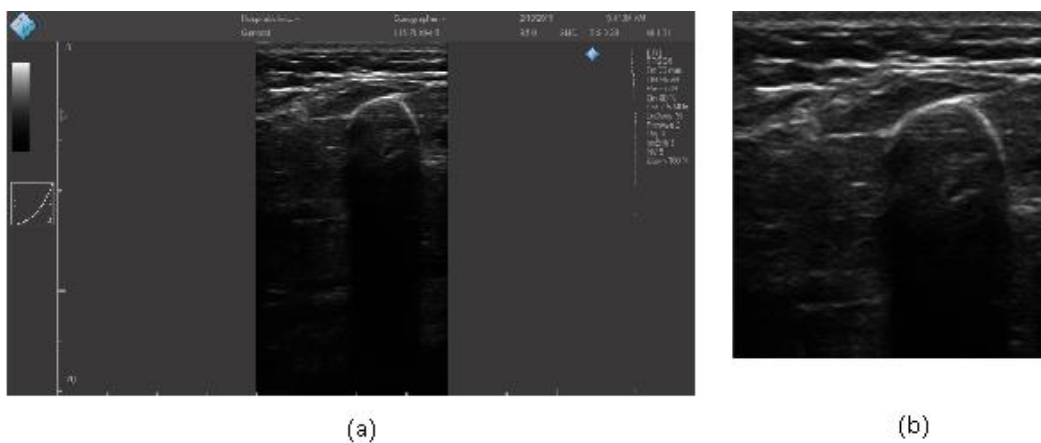
*Free hand scanning* dilakukan dengan dua cara. Cara yang pertama dengan melakukan penyisiran secara bertahap pada lengan tangan dan menyimpan hasil representasi citranya dengan ekstensi citra \*.jpg. Penyisiran secara bertahap ini dilakukan dengan memilah di berbagai posisi tangan, sehingga dalam penelitian ini didapatkan representasi citra tulang panjang manusia dalam berbagai sisi, sehingga keragaman informasi ini dapat membantu sistem untuk mengenali fitur-fitur yang terdapat pada citra representasi tulang pada sensor USG.

Sedangkan cara yang kedua, penyisiran dilakukan dengan menggerakkan probe sensor USG secara teratur, baik tempo maupun sudut kemiringan probe (selalu diusahakan tegak lurus dari area yang disisir) di sepanjang lengan tangan, baik tangan kanan maupun tangan kiri. Hasil penyisiran disimpan dalam format video dengan ekstensi data \*.avi, untuk selanjutnya video tersebut dicacah agar dapat diambil data di setiap framenya dan disimpan dengan ekstensi citra \*.jpg. Data citra yang berekstensi \*.jpg itu digunakan oleh sistem untuk dilatihkan kepada sistem berbasis algoritma *Deep Learning* dengan metode *Convolutional Neural Network* (CNN).

### **3.2 Anotasi Citra USG**

Hasil representasi tulang panjang manusia pada citra USG tidak hanya menampilkan representasi area tulang saja, melainkan juga menampilkan beberapa informasi konfigurasi sensor USG, seperti yang ditampilkan pada Gambar 1.1. Dikarenakan informasi tersebut tidak kita gunakan sebagai parameter pelatihan, maka citra yang ada kita lakukan *cropping* atau memotong di beberapa area

tertentu, sehingga yang kita dapatkan adalah citra representasi tulang, jaringan lunak, otot, daging, kulit, dsb. Citra USG yang sudah dipotong disajikan pada Gambar 3.2.



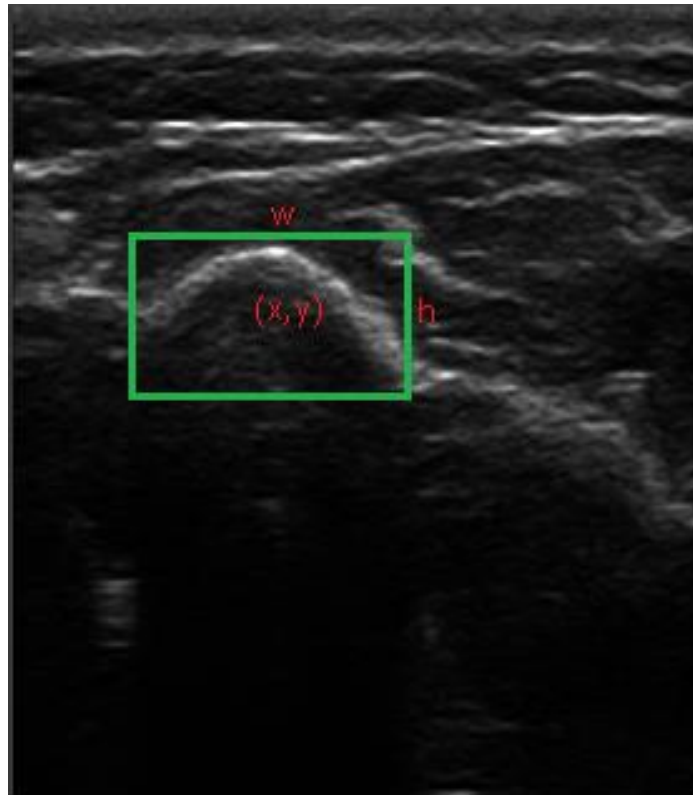
Gambar 3.2 *Cropping* Citra USG

Pada Gambar 3.2 (a) merupakan citra ultrasound spesimen tulang lengan manusia yang masih mengandung informasi konfigurasi sensor B-scan USG. Dan pada Gambar 3.2 (b) adalah citra USG yang sudah dipotong di area hasil scanning B-scan USG. Fokus pada penelitian ini adalah mengenali area tulang, oleh sebab itu secara manual peneliti dan tim penelitian harus mengenali area tulang terlebih dahulu pada citra USG secara kasat mata. Untuk kemudian memberi batasan *bounding box (bbox)* pada area tulang. Area tulang yang sudah diberi batasan kotak *bbox* itulah yang disebut sebagai RoI tulang.

Anggota tim penelitian telah berkonsultasi kepada Dr. Spesialis Radiologi untuk menentukan RoI tulang. Informasi dari seseorang yang ahli pada bidang citra radiologi sangat dibutuhkan dan penting. Hal ini sebab, informasi tersebut sebagai dasar parameter yang kita latihkan kepada sistem. Sehingga jika informasi tersebut sudah teruji benar. Maka hasil deteksi yang dilakukan oleh sistem setelah dilakukan pelatihan algoritma *deep learning* dapat dipertanggungjawabkan.

Nilai RoI pada umumnya direpresentasikan dengan nilai *bbox* yang terdiri dari batas bawah pixel  $(x1,y1)$ , batas atas pixel  $(x2,y2)$ , dan dimensi citra USG

(*width, height*). Gambar 3.3 merupakan ilustrasi dari citra USG tulang panjang manusia yang sudah ditentukan RoI tulangnya.



Gambar 3.3 Penentuan RoI Secara Manual

### 3.3 Pelatihan Algoritma pada Sistem

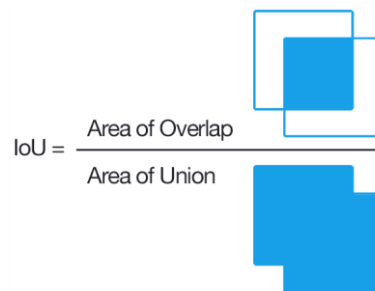
Implementasi algoritma berbasis *deep learning* menggunakan metode *Convolutional Neural Network* (CNN) dengan arsitektur *You Only Look Once* (YOLO) . YOLO digunakan untuk melakukan deteksi *region of interest* (RoI) tulang yang terdapat pada citra USG.

Dataset yang ada dibagi menjadi 2 data utama, yaitu data pelatihan dan data uji. Sedangkan di data pelatihan terbagi lagi menjadi 2 data, yaitu data latih dan data evaluasi. Data evaluasi dibutuhkan untuk melakukan tuning terhadap *hyper* parameter yang dinisialisasi sejak awal. Data ini juga bisa menjadi acuan untuk melakukan *early stop* saat tren pelatihan data terhadap sistem memasuki pola *overfitting*.

Pelatihan sistem menggunakan metode CNN terdiri dari dua tahapan. Tahapan pertama adalah *feed-forward network* dan tahapan kedua adalah *back-pass network*. Saat *feed-forward*, citra yang digunakan melalui beberapa proses konvolusi. Untuk arsitektur YoloV3 seperti pada Gambar 2-4, terdapat sebanyak total 53 layer, tahapan ini mengeluarkan nilai prediksi dalam 3 skala; skala besar, medium, dan kecil.

Di setiap prediksi nilai yang dikeluarkan adalah tensor box yang berisi 4 nilai *offset bbox*, *objectness*, dan *score class*. Di tiap tensor terdapat 3 box yang diprediksi, nantinya hanya dipilih 1 box yang mewakili grid tersebut untuk mendeteksi keberadaan tulang panjang manusia.

Di akhir tahapan *feed-forward* yang mengeluarkan prediksi *bbox* RoI tulang akan dibandingkan dengan *bbox ground truth* di citra tersebut. Nilai selisih antara *bbox ground truth* dan *bbox* prediksi diukur menggunakan *loss function*, sehingga nilai *loss* tersebut akan dipelajari oleh sistem, dan di tahapan *back-pass* algoritma akan mengoreksi nilai parameter bobot dan bias (*weights*) agar saat memasuki tahapan *feed-forward* berikutnya, nilai *error loss* dapat diperkecil. Gambar 3.4 di bawah ini mengilustrasikan cara menghitung IoU (Gambar diambil dari internet).



Gambar 3.4 Ilustrasi dari *Intersection over Union Bounding Box*

Pelatihan terbagi menjadi dua skenario, skenario pertama adalah pelatihan sistem dengan menggunakan *threshold*  $\text{IoU} > 0.5$ , dan skenario kedua adalah pelatihan sistem dengan menggunakan *threshold*  $\text{IoU} > 0.75$ . Pelatihan menggunakan dua skenario ini untuk mengukur respon pelatihan sistem terhadap

nilai *average loss* di setiap iterasi pelatihan dan *mean average precision* (mAP) yang dihitung dari data evaluasi saat pelatihan sistem berlangsung setiap 4 epoch.

IoU adalah perbandingan antara *bbox* hasil deteksi sistem dengan *bbox ground truth* seperti yang diilustrasikan pada Gambar 3.4. Dan persamaan untuk menghitung IoU sebagai berikut,

$$IoU = \frac{bbox_{pred} \cap bbox_{truth}}{bbox_{pred} \cup bbox_{truth}} \quad (2.8)$$

Berdasarkan perhitungan nilai mAP pada penelitian terakhir yang dilakukan oleh Joseph Redmon, dkk [1], didapatkan kesimpulan bahwa arsitektur YoloV3 memiliki nilai akurasi yang lebih baik dibandingkan arsitektur Yolo sebelumnya. Sehingga pada penelitian ini menggunakan arsitektur YoloV3 untuk melakukan pelatihan sistem dan pengujian deteksi RoI tulang pada citra USG.

### 3.4 Pengujian dan Evaluasi

Pengujian dilakukan dengan mengukur tingkat keberhasilan sistem dalam melakukan deteksi RoI tulang pada citra USG. Hasil deteksi diukur menggunakan parameter *metric evaluation*, untuk mengetahui seberapa besar akurasi sistem dalam melakukan deteksi.

*Metric evaluation* yang digunakan menghitung nilai *mean average precision* (mAP) untuk setiap *class* di tiap citra. Nilai mAP dihitung dengan mengambil rata-rata nilai *average precision* (AP). AP didapatkan dengan menghitung luas area yang ada di bawah kurva koordinat cartesian antara *precision* (sumbu *x*) dan *recall* (sumbu *y*).

Nilai *precision* mengukur seberapa akurat *network detector* dalam memprediksi objek yang terdapat di suatu citra. *Precision* merupakan prosentase prediksi objek yang berhasil, didapatkan dari perbandingan antara nilai prediksi *true positive* (TP) dengan jumlah dari deteksi objek yang positif, TP ditambah dengan *false positive* (FP).

Sedangkan nilai *recall* mengukur seberapa baik *network detector* dalam menemukan semua objek yang benar (baik benar ditemukan objek, maupun benar tidak ditemukan objek ). *Recall* didapatkan dengan membandingkan nilai TP dengan jumlah dari seluruh deteksi objek yang didapatkan pada citra tersebut, TP ditambah dengan dengan *false negative* (FN).

Suatu area *bbox* terhitung sebagai TP jika nilai IoU nya melebihi ambang batas yang ditentukan dan prediksi *class* nya sesuai dengan *class* yang terdapat pada *ground truth*. Jika tidak memenuhi ambang batas tetapi memiliki *class* yang sama dihitung sebagai FP. Jika benar tidak ditemukan objek pada citra terhitung sebagai FN.

Formulasi matematika dari perhitungan *precision*, *recall*, *average precision*, dan mAP adalah,

$$precision(p) = \frac{TP}{TP + FP} \quad (2.9)$$

$$recall(r) = \frac{TP}{TP + FN} \quad (2.10)$$

$$AP = \int_0^1 p(r)dr \quad (2.11)$$

Pendekatan AP juga bisa dengan melakukan interpolasi pada nilai recall di beberapa titik,

$$AP = \sum (r_{n+1} - r_n) P_{interpolation}(r_{n+1}) \quad (2.12)$$

$$P_{interpolation}(r_{n+1}) = \max p(\tilde{r}) ; \tilde{r} \geq r_{n+1} \quad (2.13)$$

Sehingga Parameter yang hendak dianalisis di tiap skenario pelatihan adalah sebagai berikut; *true positive* (TP), *false positive* (FP), *false negative* (FN), *precision*, *recall*, *mean average precision* (mAP), dan *average loss*. Nilai *average*

*loss* menunjukkan seberapa besar error yang didapatkan saat pelatihan sistem di tiap iterasi.

*Halaman ini sengaja dikosongkan*



## BAB 4

### HASIL DAN PEMBAHASAN

Dataset penelitian yang diambil citra USG spesimen tulang panjang manusia diambil dari 10 orang dan dibagi menjadi 7 subdata. Profil data terdiri dari 7 pria dan 3 wanita dengan rentang usia antara 15-50 tahun. Untuk pengelompokan subdata direpresentasikan dengan huruf A, B, C, D, E, F, dan G. Pembagian dataset yang digunakan untuk tahapan pelatihan dan pengujian disajikan pada Tabel 4.1.

Tabel 4.1 Dataset Tulang Lengan Manusia

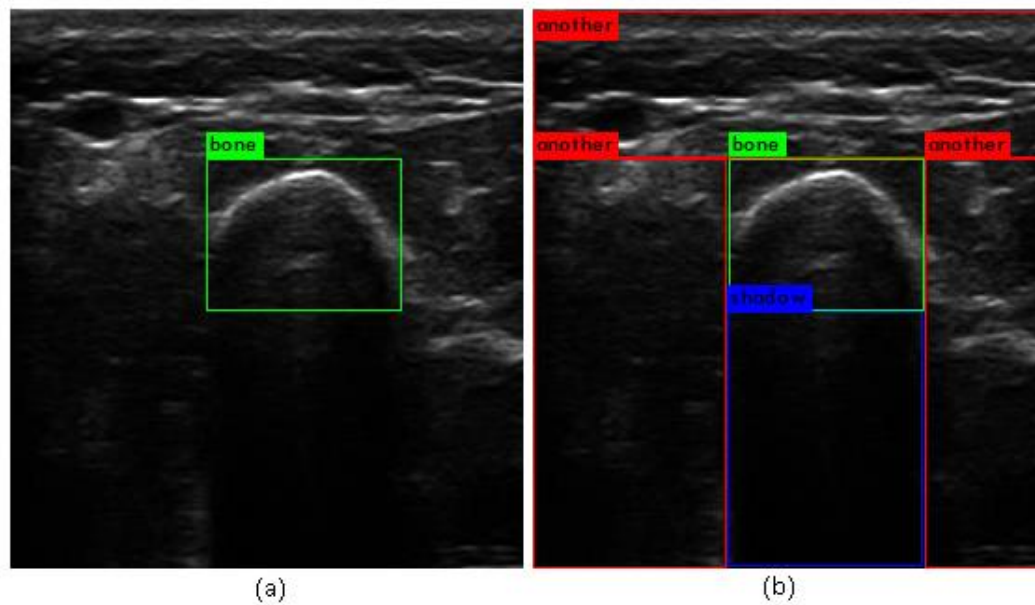
Subdata	Data Pelatihan		Data Pengujian
	Data Latih	Data Evaluasi	
A	540	60	120
B	450	50	100
C	360	40	80
D	450	50	100
E	54	6	-
F	180	20	100
G	-	-	120
Jumlah	2034	226	620

#### 4.1 Anotasi Data Citra USG

Setiap citra USG spesimen tulang lengan manusia dianotasikan sebagai 3 objek; *bone*, *another*, dan *shadow*. *Bone* merupakan permukaan tulang yang intensitas pikselnya relatif tinggi (putih) dan memiliki bentuk seperti kurva parabola. *Shadow* adalah bayangan dari permukaan tulang yang memiliki intensitas piksel relatif rendah (gelap) dibandingkan dengan piksel pada permukaan tulang, hal ini disebabkan gelombang *ultrasound* tidak dapat menembus tulang, sehingga gelombang yang kembali ke receiver probe sensor B-scan USG hanya mampu merepresentasikan permukaan tulang.

Sedangkan *another* merupakan area di luar *bone* dan *shadow*. Area tersebut berada di atas dan di kedua sisi *bone*. *Another* bisa merepresentasikan banyak hal dari jaringan tubuh lengan manusia, seperti; otot, pembuluh darah, dsb.

Sebab penelitian ini fokus untuk mengenali objek tulang, maka bagian-bagian tersebut tidak dilatihkan secara detail kepada sistem. Penentuan RoI manual objek bone dapat dilihat pada Gambar 4.1.



Gambar 4.1 Anotasi RoI Objek pada Citra USG

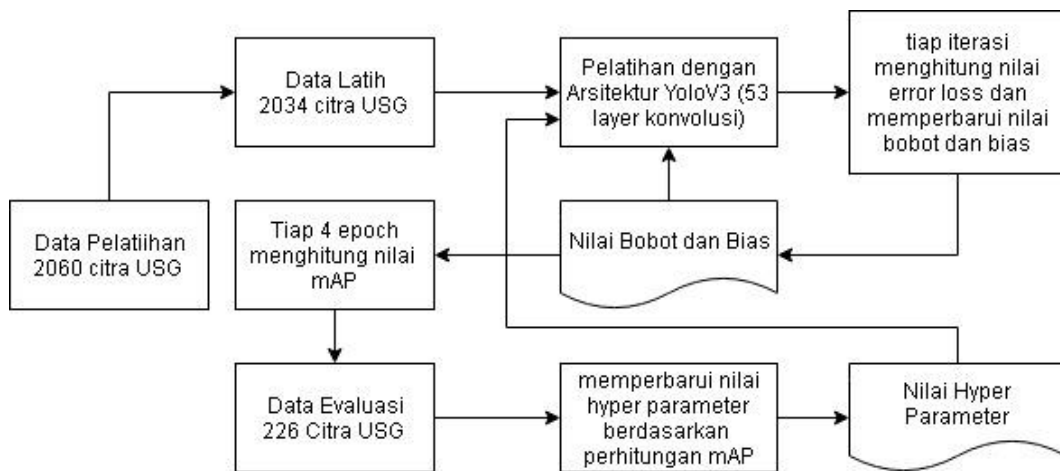
Gambar 4.1 (a) adalah ilustrasi RoI manual objek tulang, dan Gambar 4.1 (b) adalah ilustrasi RoI objek *another* dan objek *shadow* berdasarkan dari RoI tulang pada gambar (a). Di sebagian dataset citra USG, area *another* memiliki bentuk tekstur dan intensitas piksel yang hampir menyerupai objek *bone*. Secara kasat mata, bisa dibedakan bahwa objek tersebut bukanlah *bone*, dikarenakan tidak memiliki *shadow* di bawahnya. Oleh sebab itu pada tahap pelatihan sistem objek *another* dan *shadow* juga dilatihkan, agar sistem dapat mengenali dan membedakan objek *bone* dan yang menyerupainya di area *another*.

## 4.2 Pelatihan Sistem

Pelatihan dataset terhadap sistem dibagi menjadi 2 skenario. Skenario pertama, pelatihan sistem menggunakan *threshold* IoU  $> 0.5$  (skenario pelatihan

P1). Dan skenario kedua pelatihan sistem menggunakan *threshold* IoU > 0.75 (skenario pelatihan P2).

Sesuai dengan data pada Tabel 4.1, data citra tiap batch (iterasi) yang dilatih sebanyak 64 citra, sehingga untuk melatih keseluruhan data latih sebanyak 2034 citra dibutuhkan  $\approx 32$  iterasi (1 epoch). Maksimal iterasi yang digunakan pada pelatihan untuk semua skenario sebanyak 6000 iterasi ( $\approx 188$  epoch). Alur data pelatihan sistem dapat dilihat pada Gambar 4.2.



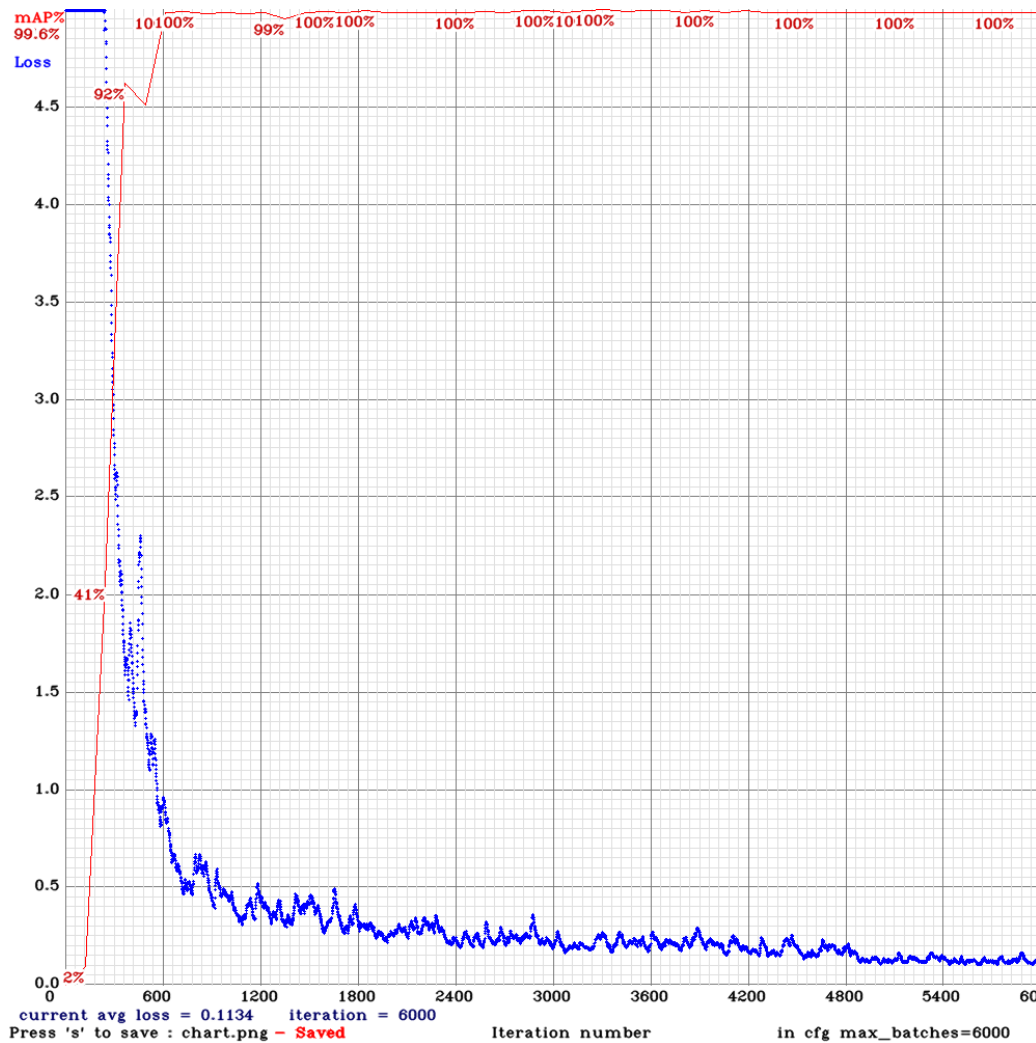
Gambar 4.2 Alur Data Pelatihan Sistem

Parameter yang dianalisis di tiap iterasi adalah nilai *average loss*. Dan untuk tiap 4 epoch, dilakukan penghitungan mAP untuk menganalisis respon sistem menggunakan nilai bobot dan bias (*weights*) yang sudah diperbarui di tiap-tiap batch sebelumnya. Parameter-parameter yang mempengaruhi nilai mAP adalah; *true positive* (TP), *false positive* (FP), *false negative* (FN), *precision*, *recall*, dan *average precision* (AP).

Sumber daya komputasi yang digunakan pada penelitian ini menggunakan fasilitas yang tersedia di Google Colab. Dimana Google Colab menyediakan sumber daya komputasi dengan kemampuan GPU Tesla K80 dan kapasitas RAM GPU sebesar 12GB.

#### 4.2.1 Pelatihan Sistem dengan *Threshold IoU > 0.5 (weights<sub>50</sub>)*

Grafik *average loss* dan mAP selama masa pelatihan menggunakan *threshold IoU 0.5* dapat dilihat pada Gambar 4.3. Pada awal pelatihan di iterasi ke-100, nilai *average loss* sangat tinggi dengan nilai >1000.0, sehingga nilai mAP pada citra evaluasi juga sangat kecil di nilai 0.02. Saat periode pelatihan memasuki iterasi ke-300 nilai mAP menjadi lebih baik di nilai 0.41, meskipun nilai *average loss* berada di angka 5.0.



Gambar 4.3 Grafik *Average Loss* dan Menggunakan *Threshold IoU>0.5*

Di iterasi ke-400 nilai mAP sudah sangat baik di angka 0.92 dengan nilai *average loss* di angka 1.5. Semakin bertambah iterasi, nilai *average loss* memiliki

tren grafik menurun, hingga di iterasi ke-600 nilainya sudah mencapai 1.0 dan perhitungan mAP mencapai nilai maksimal 1.0. Hal ini berarti pada iterasi tersebut, nilai  $weights_{50}$  sudah cukup baik karena mampu dengan tepat mendeteksi citra uji secara benar. Nilai mAP tetap bertahan di nilai maksimal 1.0 hingga iterasi pelatihan berakhir di batch ke-6000.

Dan nilai *average loss* terus memiliki tren grafik menurun, dimulai stabil saat memasuki iterasi ke-1200 ( $<0.5$ ), terus mengecil hingga akhir iterasi mencapai nilai 0.05. Hal ini menunjukkan bahwa sistem terus mempelajari nilai error yang didapatkan di tiap iterasi untuk memperbarui nilai bobot dan bias yang digunakan pada *network detector* YoloV3.

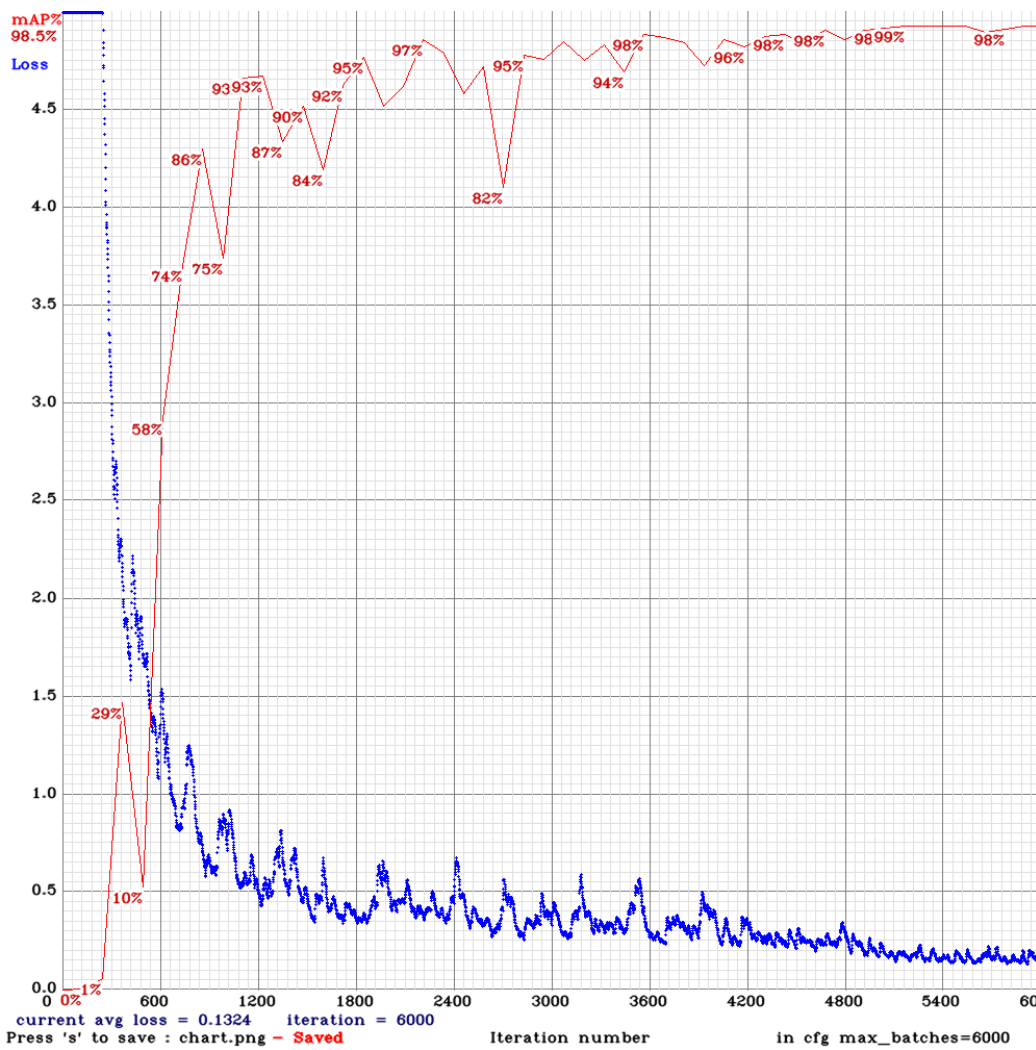
#### **4.2.2 Pelatihan Sistem dengan *Threshold IoU > 0.75 (weights<sub>75</sub>)***

Sedangkan untuk pelatihan sistem dengan *threshold IoU > 0.75*, parameter *average loss* memasuki nilai yang stabil sejak batch ke-3000, dengan perolehan nilai di bawah 0.5 dan terus mengecil hingga batch terakhir di posisi nilai 0.005. Hal ini menunjukkan bahwa sistem membutuhkan waktu iterasi yang lebih panjang untuk dapat beradaptasi dengan *threshold IoU > 0.75*. Hal ini terjadi karena hasil deteksi objek yang dianggap benar apabila memiliki IoU yang tinggi di atas  $> 0.75$ .

Dikarenakan *threshold IoU > 0.75* yang digunakan cukup tinggi untuk mengevaluasi parameter sistem. Maka nilai mAP pada awal iterasi memiliki nilai yang kecil. Pada iterasi ke-1200, nilai mAP mulai berada di posisi di atas 90%. Nilai tersebut terus membaik hingga di iterasi pelatihan yang terakhir di nilai 98%.

Perhitungan akurasi deteksi dengan mengukur nilai mAP dilakukan tiap 4 epoch pada citra evaluasi. Sebab, sistem selalu berupaya memperbaiki nilai error yang besar di tiap iterasi sehingga waktu yang dicukupkan relatif lebih lama. Akan tetapi tren mAP selalu semakin membaik di tiap penambahan periode iterasi pelatihan.

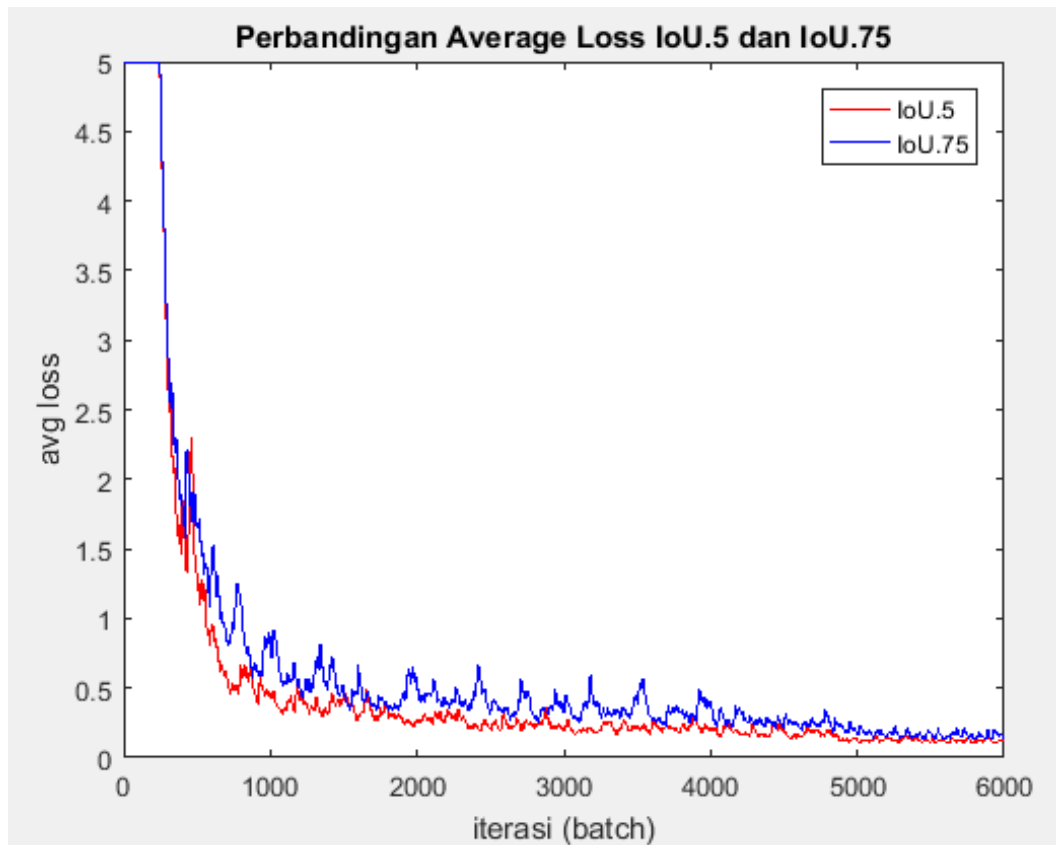
Perbandingan grafik *average loss* dan mAP selama masa pelatihan menggunakan *threshold IoU 0.75* dapat dilihat pada Gambar 4.4.



Gambar 4.4 Grafik Average Loss dan Menggunakan Threshold IoU>0.75

### 4.2.3 Analisis Proses Pelatihan Sistem dengan Berbagai Nilai Threshold IoU

Gambar 4.5 adalah grafik perbandingan nilai average loss pada tahapan pelatihan antara pelatihan sistem skenario PI dan P2.



Gambar 4.5. Perbandingan grafik *average loss* skenario P1 dan P2

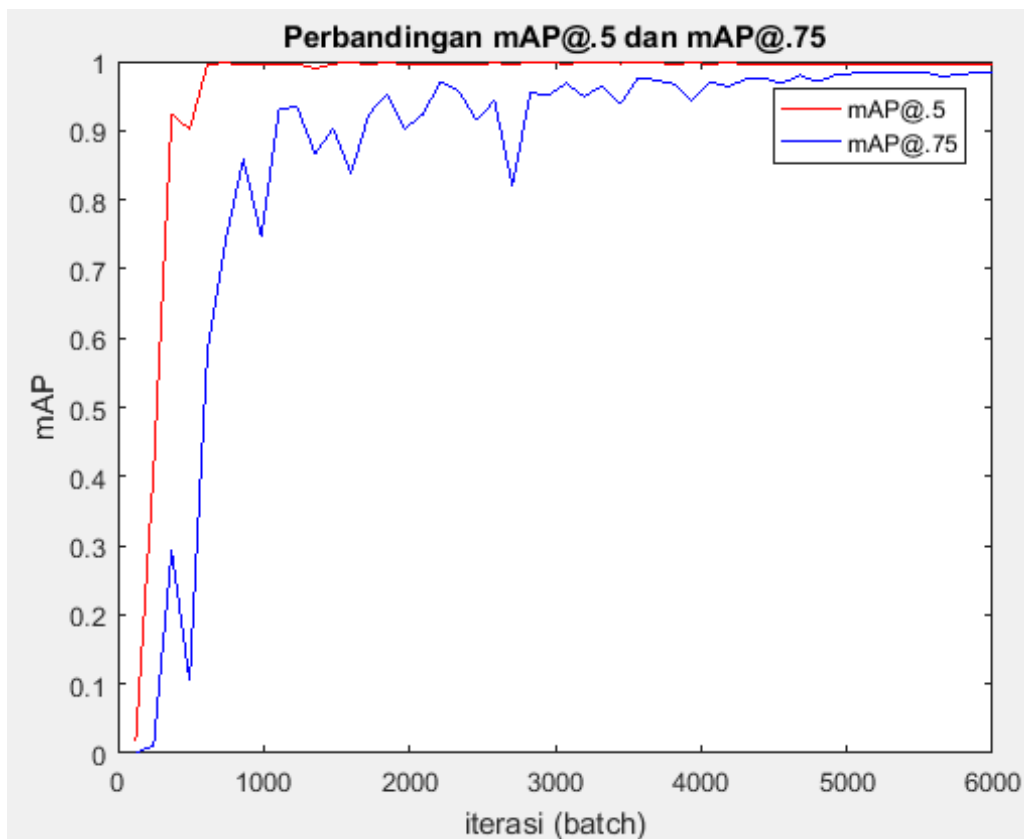
Dari Gambar 4.5 dapat dilihat bahwa pelatihan sistem dengan *threshold* IoU memiliki nilai loss yang lebih besar dari 0.75 dan membutuhkan iterasi yang lebih lama untuk bisa memperkecil nilai error di tiap iterasi. Akan tetap di akhir iterasi ke-6000, nilai *average loss* antara P1 dan P2 tidak jauh berbeda, berada di nilai 0.1134 dan 0.1324 dengan selisih yang sangat kecil 0.019.

Detail perbandingan nilai dapat dilihat pada Tabel 4.2. Hal ini menunjukkan bahwa tahapan pelatihan sistem di 2 skenario dengan jumlah iterasi yang sama menghasilkan nilai parameter bobot dan bias yang sama baiknya. Untuk seluruh iterasi nilai *average loss* menggunakan skenario pelatihan yang pertama memiliki nilai yang lebih kecil dibandingkan dengan skenario pelatihan yang kedua.

Tabel 4.2 Tabel Perbandingan *Average Loss* Tiap 600 *Batch*

Batch ke-	<i>Average loss P1</i>	<i>Average loss P2</i>
1	1.821	1.911
600	0.9248	1.3514
1200	0.4152	0.5055
1800	0.3267	0.3664
2400	0.2336	0.4403
3000	0.1906	0.3246
3600	0.2127	0.2627
4200	0.1665	0.3434
4800	0.1814	0.2949
5400	0.1371	0.1707
6000	0.1134	0.1324

Pada gambar 4.6 merupakan perbandingan nilai mAP antara pelatihan skenario P1 dengan pelatihan skenario P2. Nilai mAP dihitung pada dataset evaluasi, untuk mengukur seberapa tepat hasil prediksi menggunakan parameter *weights* yang sudah dihasilkan pada iterasi pelatihan sebelumnya.



Gambar 4.6 Perbandingan grafik mAP skenario P1 dan P2



Nilai mAP di skenario P1 sudah stabil di iterasi ke-600, sedangkan pada skenario P2, nilai mAP baru mulai stabil di iterasi ke-1800. Untuk skenario P2, nilai mAP terus beresilasi dan tren grafik menuju ke arah yang baik, yaitu nilai mAP di atas 95% pada saat iterasi ke-4500. Meskipun cukup lama, namun pada akhirnya sistem dapat melakukan deteksi RoI tulang dengan nilai IoU yang cukup tinggi, yaitu di atas 75%. Hal ini memiliki arti bahwa sistem dengan pelatihan skenario P2 mampu mendeteksi dengan sangat baik citra evaluasi, sehingga hasil prediksinya hampir menyerupai nilai ground truth nya (tingkat keserupaan di atas 90%).

Tabel 4.3 menunjukkan perbandingan nilai mAP di tiap pecuplikan pada beberapa iterasi pelatihan sistem.

Tabel 4.3 Perbandingan nilai mAP di pencuplikan beberapa batch

Batch ke-	mAP P1	mAP P2
123	0.0185	8.1251e-05
615	0.9958	0.5838
1230	0.9966	0.9340
1845	0.9979	0.9521
2460	0.9964	0.9153
3075	0.9964	0.9684
3690	0.9979	0.9730
4305	0.9964	0.9743
4920	0.9965	0.9800
5535	0.9964	0.9844

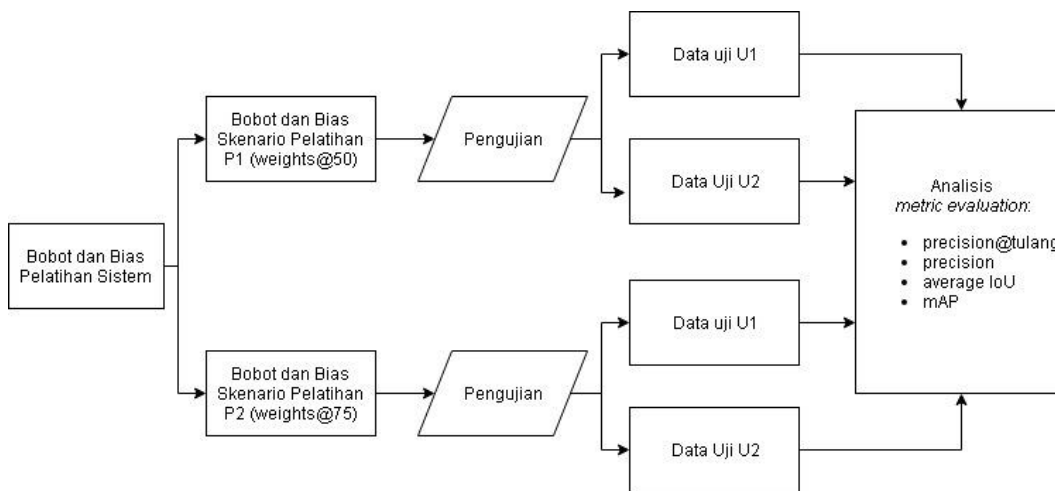
### 4.3 Pengujian

Untuk tiap masing-masing hasil pelatihan di semua skenario (P1 dan P2), diujikan ke data uji menggunakan 2 skenario. Pengujian skenario pertama kepada sub data kelompok A, B, C, D, dan F (skenario uji U1). Dan pengujian skenario kedua diujikan ke sub data kelompok G (skenario uji U2). Pembagian dua skenario ini dilakukan untuk menganalisis perilaku sistem saat pengujian dilakukan kepada sub data yang sebagian datanya sudah digunakan sebagai data pelatihan (skenario U1) dan sub data yang tidak satupun datanya digunakan sebagai data pelatihan (skenario U2). Daftar skenario pengujian dapat dilihat pada Tabel 4.4.

Tabel 4.4 Skenario Uji

Nama Skenario Uji	Weights	Skenario	Pengujian Sub Kelompok	Jumlah Citra Uji
$U1_{50}$	$weights_{50}$	U1	A, B, C, D, F	500
$U2_{50}$	$weights_{50}$	U2	G	120
$U1_{75}$	$weights_{75}$	U1	A, B, C, D, F	500
$U2_{75}$	$weights_{75}$	U2	G	120

Dalam pengujian ini, nilai parameter yang hendak dianalisis adalah  $precision_{bone}$ ,  $precision$ ,  $average IoU$ , dan mAP. Dan hasil deteksi citra diamati secara visual untuk mengamati profil dan perilaku sistem dalam mendeteksi RoI tulang pada citra USG. Gambar 4.7 merupakan blok diagram skenario pengujian yang dilakukan pada penelitian ini.



Gambar 4.7 Blok Diagram Skenario Pengujian

### 4.3.1 Analisis Metric Evaluation

Nilai *metric evaluation* diambil menggunakan *threshold IoU* yang berbeda-beda. Dimulai dari nilai *threshold* 0.5 hingga 0.95 dengan step 0.05. Nilai *precision* merupakan prosentase akurasi sistem dalam mendeteksi objek pada suatu citra. Pada penelitian ini, analisa lebih detil ditujukan kepada objek tulang, sehingga nilai *precision* untuk kelas *bone* disajikan juga.

Parameter *average IoU* untuk mengukur prosentasi ukuran *bbox* deteksi dibandingkan dengan *bbox ground truth*. Hal ini memiliki arti, bahwa ketepatan deteksi *bbox* diharapkan sesuai dengan nilai *ground truth*.

Sedangkan mAP merupakan salah satu parameter utama untuk mengukur keberhasilan *network detector* dalam melakukan deteksi objek.

#### 4.3.1.1 $Precision_{bone}$

Untuk data uji U1 penggunaan  $weights_{50}$  dan  $weights_{75}$  memiliki nilai precision yang sama dimulai dari *threshold IoU* 0.5 hingga 0.6. Untuk  $weights_{50}$  nilai precision yang tinggi hanya ada saat *threshold IoU* > 0.65, selebihnya  $weights_{75}$  di semua sisa *threshold IoU* memberikan nilai *precision* yang lebih tinggi di. Hal ini menunjukkan bahwa  $weights_{75}$  memiliki akurasi yang lebih baik dibandingkan  $weights_{50}$  dalam mendeteksi ROI tulang pada citra USG. Perbandingan nilai *precision* untuk objek tulang dapat dilihat pada Tabel 4.5.

Tabel 4.5 Nilai Parameter  $precision_{bone}$  untuk Semua Skenario Uji

IoU >	$precision_{bone}$			
	$U1_{50}$	$U1_{75}$	$U2_{50}$	$U2_{75}$
0.50	100.00	100.00	99.17	99.17
0.55	100.00	100.00	98.35	98.33
0.60	99.80	99.80	95.87	97.50
0.65	99.60	99.40	92.56	89.17
0.70	98.00	98.80	86.78	84.17
0.75	95.80	96.00	68.60	69.17
0.80	88.00	91.00	48.76	47.50
0.85	67.60	76.40	27.27	25.83
0.90	33.80	42.40	4.96	10.83
0.95	5.40	7.00	0.00	1.67

Sedangkan untuk data uji U2, penggunaan  $weights_{50}$  dan  $weights_{75}$  masing-masing memiliki nilai yang lebih unggul di beberapa urutan random *threshold IoU*. Hal ini memiliki makna, bahwa parameter bobot dan bias hasil pelatihan sistem skenario P1 dan P2, saat diujikan kepada citra USG yang tidak pernah dijadikan bahan pelatihan sistem, memiliki peluang yang sama untuk menghasilkan deteksi objek tulang yang baik.

#### 4.3.1.2 Precision

Untuk data uji U1 penggunaan  $weights_{50}$  dan  $weights_{75}$  masing-masing memiliki nilai yang lebih unggul di beberapa urutan random  $threshold$  IoU. Tetapi  $weights_{75}$  memiliki nilai  $precision$  yang relatif lebih sering di  $threshold$   $IoU > 0.75$ . Hal ini mungkin terjadi karena  $precision$  menghitung seluruh akurasi  $network$  detector dalam mendeteksi semua objek (*bone*, *shadow*, dan *another*). Perbandingan nilai  $precision$  dapat dilihat pada Tabel 4.6.

Tabel 4.6 Nilai Parameter  $Precision$  untuk Semua Skenario Uji

IoU >	<i>All Object Precision</i>			
	$U1_{50}$	$U1_{75}$	$U2_{50}$	$U2_{75}$
0.50	100.00	99.96	99.83	99.67
0.55	100.00	99.96	99.67	99.50
0.60	99.96	99.92	99.17	99.33
0.65	99.76	99.80	98.50	97.50
0.70	99.32	98.48	97.17	96.34
0.75	98.72	98.88	92.85	93.18
0.80	97.08	97.72	87.85	87.85
0.85	92.08	93.80	79.70	80.70
0.90	80.92	82.12	63.23	65.89
0.95	52.00	49.32	27.29	33.44

Sedangkan untuk data uji U2, penggunaan  $weights_{75}$  memiliki nilai relatif lebih baik dibandingkan  $weights_{50}$ . Hal ini berarti parameter bobot dan bias hasil pelatihan skenario P2 memiliki kecenderungan lebih baik dalam mendeteksi keseluruhan objek yang dilatih kepada sistem.

#### 4.3.1.3 Average IoU

Untuk data uji U1 penggunaan  $weights_{75}$  hampir di semua  $threshold$   $IoU$  memiliki nilai yang lebih tinggi dibandingkan dengan  $weights_{50}$ . Hal ini berarti  $network$  detector  $weights_{75}$  menghasilkan nilai prediksi  $bbox$  yang paling mendekati dengan nilai  $bbox$  *ground truth*. Perbandingan nilai *aveargae* IoU untuk objek tulang dapat dilihat pada Tabel 4.7.

Tabel 4.7 Nilai Parameter Average IoU untuk Semua Skenario Uji

IoU >	Average IoU			
	$U1_{50}$	$U1_{75}$	$U2_{50}$	$U2_{75}$
0.50	93.34	93.46	89.66	89.91
0.55	93.34	93.46	89.57	98.82
0.60	93.32	93.43	89.27	89.73
0.65	93.19	93.36	88.85	88.58
0.70	92.89	93.15	87.96	87.79
0.75	92.45	92.71	84.80	85.49
0.80	91.18	91.80	80.91	81.35
0.85	87.03	88.56	74.18	75.42
0.90	77.24	78.31	59.76	62.41
0.95	50.33	47.75	25.34	32.26

Begitu juga dengan data uji U2, *network detector weights<sub>75</sub>* memiliki nilai yang lebih baik dibandingkan *weights<sub>50</sub>* hampir di semua *threshold IoU*. Yang berarti *weights<sub>75</sub>* dapat beradaptasi dengan baik dalam mendeteksi objek pada sekelompok citra USG yang sebelumnya belum pernah menjadi bagian dari dataset pelatihan sistem.

#### 4.3.1.4 mAP

Pada data uji U1, penggunaan *weights<sub>50</sub>* dan *weights<sub>75</sub>* memiliki nilai yang lebih tinggi dengan proporsi jumlah yang sama di nilai random *threshold IoU*. Akan tetapi *network detector weights<sub>75</sub>* memiliki kecenderungan nilai mAP yang relatif lebih tinggi di *threshold IoU > 0.8*.

Tabel 4.8 Nilai Parameter Average mAP untuk Semua Skenario Uji

IoU >	mAP			
	$U1_{50}$	$U1_{75}$	$U2_{50}$	$U2_{75}$
0.50	99.98	99.96	99.91	99.73
0.55	99.98	99.96	99.86	99.37
0.60	99.97	99.89	98.91	98.90
0.65	99.59	99.73	96.98	94.50
0.70	98.78	99.09	93.46	91.65
0.75	97.68	97.46	81.98	83.89
0.80	92.96	94.16	72.62	72.86
0.85	81.09	85.28	58.16	61.17
0.90	60.92	63.61	33.93	39.79
0.95	25.75	24.37	6.54	10.79

Data yang disajikan pada Tabel 4.8 memiliki makna bahwa *network detector weights<sub>75</sub>* dengan parameter *threshold* yang relatif besar mampu dengan baik mendeteksi keberadaan objek di suatu citra.

Dan untuk data uji U2, penggunaan *weights<sub>75</sub>* memiliki pola yang relatif sama dengan data uji U1. *Network detector weights<sub>75</sub>* memiliki nilai yang lebih tinggi di parameter *threshold* yang relatif besar. Meskipun diujikan ke citra uji yang sebelumnya tidak masuk sebagai bahan pelatihan. Hal ini juga cukup memberikan kecenderungan perilaku bahwa *network detector weights<sub>75</sub>* mampu beradaptasi dengan citra USG spesimen tulang manusia.

#### 4.3.2 Analisis Proses Pengujian Secara Keseluruhan

Tabel 4.9 adalah penyajian data untuk *mAP<sub>50</sub>*, dari data yang ditampilkan dapat dilihat bahwa *weights<sub>50</sub>* untuk semua skenario uji memiliki nilai yang tinggi pada parameter *precision<sub>bone</sub>*, *precision* dan *mAP*. Dan untuk penggunaan *weights<sub>75</sub>* hanya unggul di parameter *average IoU*. Hal ini berarti untuk parameter *weights<sub>50</sub>* memiliki nilai yang optimal ketika diujikan menggunakan *threshold IoU* lebih besar dari 0.5.

Tabel 4.9 Pengujian dengan *mAP<sub>50</sub>*

Skenario Uji	IoU 0.5			
	<i>precision<sub>bone</sub></i>	<i>precision</i>	<i>Average IoU</i>	<i>mAP</i>
<i>U1<sub>50</sub></i>	100.00	100.00	93.34	99.98
<i>U1<sub>75</sub></i>	100.00	99.96	93.46	99.96
<i>U2<sub>50</sub></i>	99.17	99.83	89.66	99.91
<i>U2<sub>75</sub></i>	99.17	99.67	89.91	99.73

Sedangkan untuk Tabel 4.10 merupakan penyajian data untuk *mAP<sub>75</sub>*, dari data yang ditampilkan dapat dilihat bahwa *weights<sub>50</sub>* ketika diujikan ke kelompok citra uji yang pertama hanya memiliki nilai yang unggul di parameter *mAP* dengan selisih yang sangat kecil dibandingkan hasil nilai *mAP* menggunakan *weights<sub>75</sub>*. Dan untuk skenario uji yang kedua, *weights<sub>50</sub>* tidak dapat mengungguli perhitungan *metric evaluation* yang didapatkan dari *weights<sub>75</sub>*. Hal ini berarti

untuk *threshold* IoU lebih besar dari 0.75 lebih baik menggunakan bobot dan bias hasil pelatihan skenario yang kedua.

Tabel 4.10 Pengujian dengan  $mAP_{75}$

Skenario Uji	IoU .75			
	$precision_{bone}$	$precision$	$Average IoU$	$mAP$
$U1_{50}$	95.80	98.72	92.45	97.68
$U1_{75}$	96.00	98.88	92.71	97.46
$U2_{50}$	68.60	92.85	84.80	81.98
$U2_{75}$	69.17	92.85	85.49	83.89

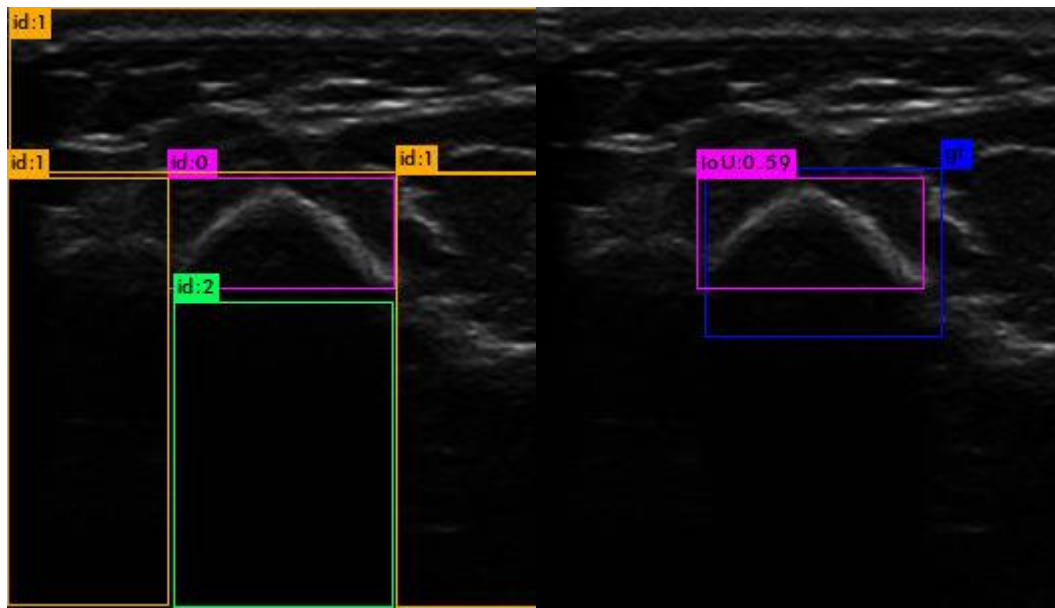
Dan untuk Tabel 4.11 merupakan penyajian data untuk  $mAP_{50:95}$ , untuk semua skenario uji tampak bahwa penggunaan  $weights_{75}$  lebih baik dibandingkan dengan  $weights_{50}$ . Dari hal ini dapat disimpulkan bahwa penggunaan *network detector*  $weights_{75}$  lebih baik dalam beradaptasi di semua skenario dataset uji U1 dan U2 untuk mendeteksi objek. Sehingga hasil deteksi dari *network detector*  $weights_{75}$  memiliki peluang yang lebih tinggi untuk mendekati nilai *bbox ground truth* dibandingkan penggunaan *network detector*  $weights_{50}$ .

Tabel 4.11 Pengujian dengan  $mAP_{50:95}$

Skenario Uji	$mAP_{50:95}$
$U1_{50}$	85.67
$U1_{75}$	86.35
$U2_{50}$	74.23
$U2_{75}$	75.27

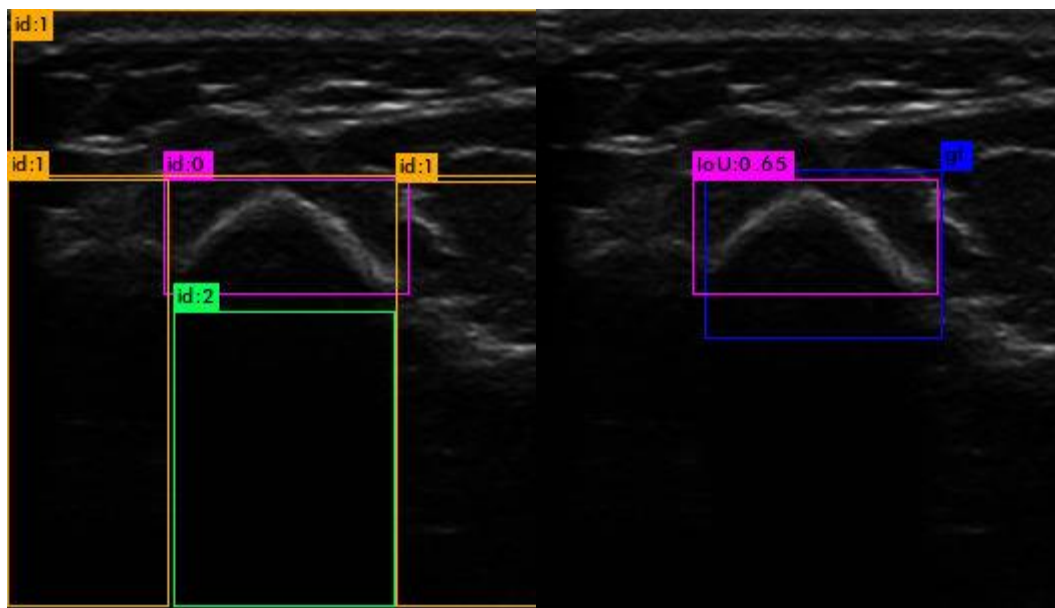
### 4.3.3 Pengamatan Hasil Deteksi Secara Visual

Gambar 4.8 adalah citra uji U1 dengan perolehan IoU yang paling rendah pada *network detector*  $weights_{50}$  memiliki nilai 0.59 dibandingkan dengan citra uji yang sama hasil *network detector*  $weights_{75}$  memiliki nilai 0.65. Gambar 4.8 (a,b) adalah hasil deteksi menggunakan  $weights_{50}$ , dan Gambar 4.8 (c,d) adalah hasil deteksi menggunakan  $weights_{75}$ . Gambar 4.8 (a,c) menampilkan hasil deteksi seluruh objek sedangkan Gambar 4.8 (b,d) hanya menampilkan hasil deteksi objek tulang dan *ground truth* tulang.



(a)

(b)



(c)

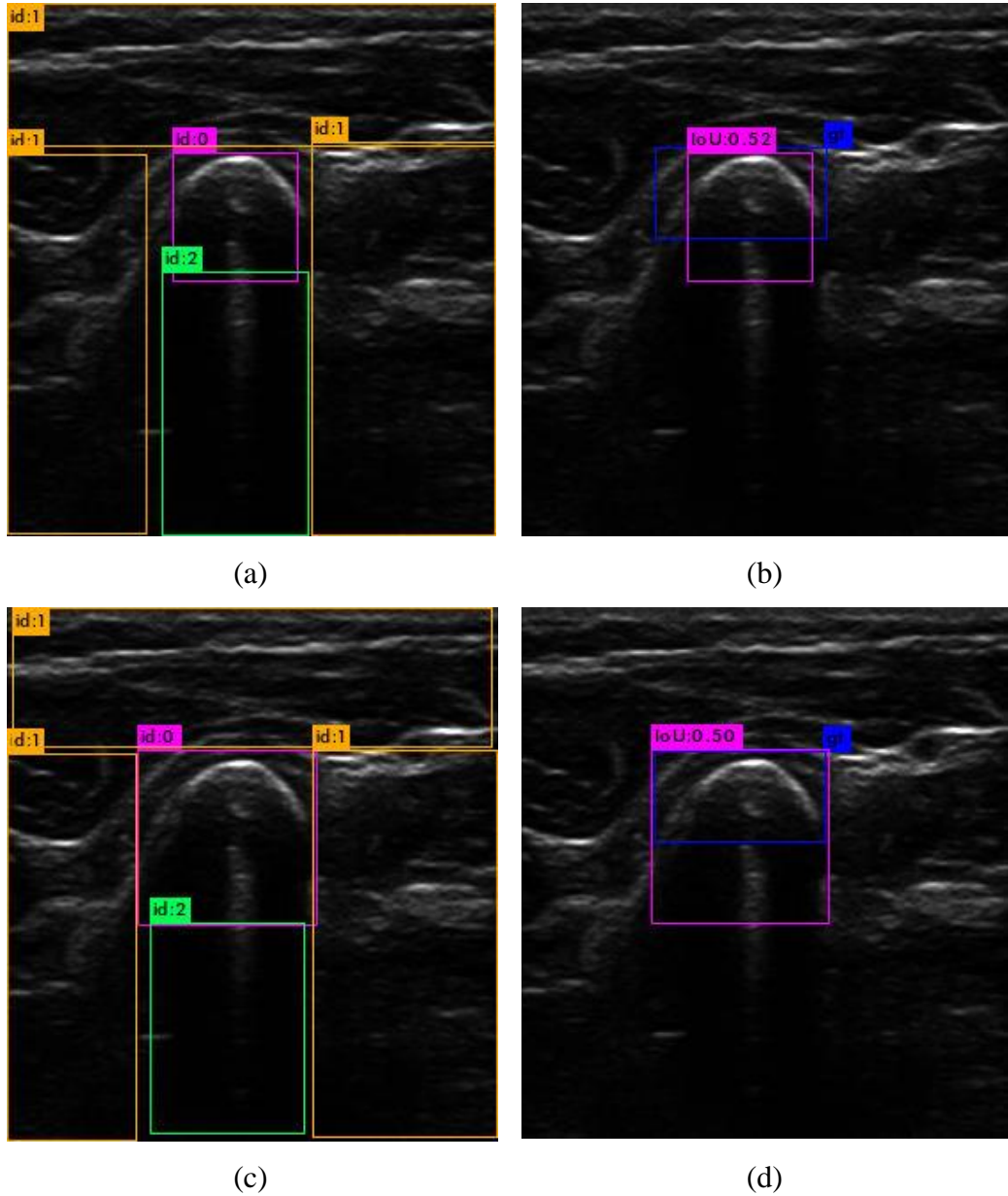
(d)

Gambar 4.8 Nilai IoU Terendah Untuk Skenario Uji U1

Gambar 4.9 adalah citra uji U2 dengan perolehan IoU yang paling rendah pada *network detector weights*<sub>50</sub> memiliki nilai 0.52 dibandingkan dengan citra uji yang sama hasil *network detector weights*<sub>75</sub>, memiliki nilai 0.50. Gambar 4.9 (a,b) adalah hasil deteksi menggunakan *weights*<sub>50</sub>, dan Gambar 4.9 (c,d) adalah hasil deteksi menggunakan *weights*<sub>75</sub>. Gambar 4.9 (a,c) menampilkan hasil



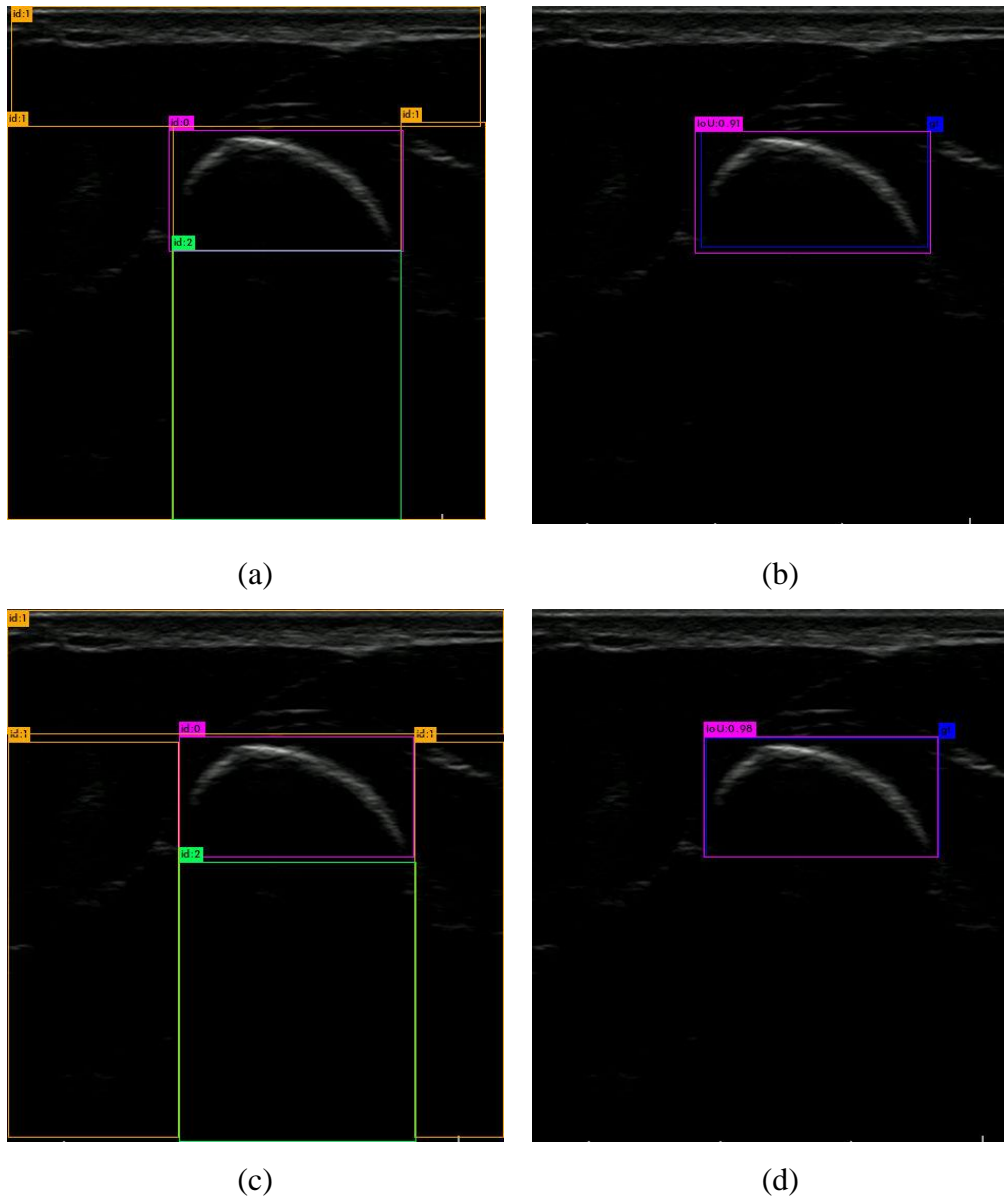
deteksi seluruh objek sedangkan Gambar 4.9 (b,d) hanya menampilkan hasil deteksi objek tulang dan *ground truth* tulang.



Gambar 4.9 Nilai IoU Terendah Untuk Skenario Uji U2

Gambar 4.10 adalah citra uji U1 dengan perolehan IoU yang paling tinggi pada *network detector weights*<sub>75</sub> memiliki nilai 0.98 dibandingkan dengan citra uji yang sama hasil *network detector weights*<sub>50</sub> memiliki nilai 0.91. Gambar 4.10 (a,b) adalah hasil deteksi menggunakan *weights*<sub>50</sub>, dan Gambar 4.10 (c,d) adalah

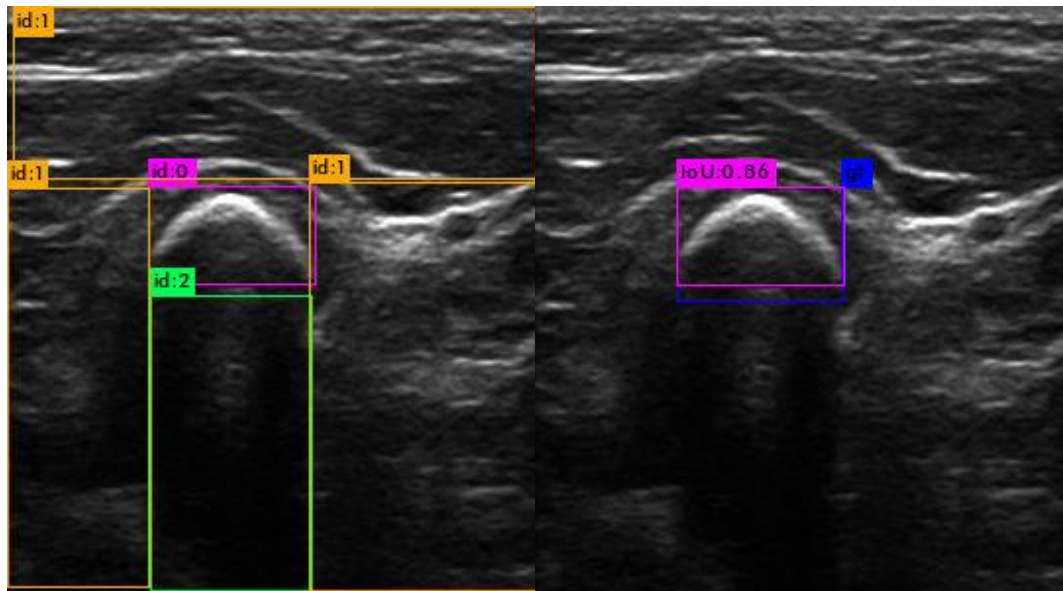
hasil deteksi menggunakan  $weights_{75}$ . Gambar 4.10 (a,c) menampilkan hasil deteksi seluruh objek sedangkan Gambar 4.10 (b,d) hanya menampilkan hasil deteksi objek tulang dan *ground truth* tulang.



Gambar 4.10 Nilai IoU Tertinggi Untuk Skenario Uji U1

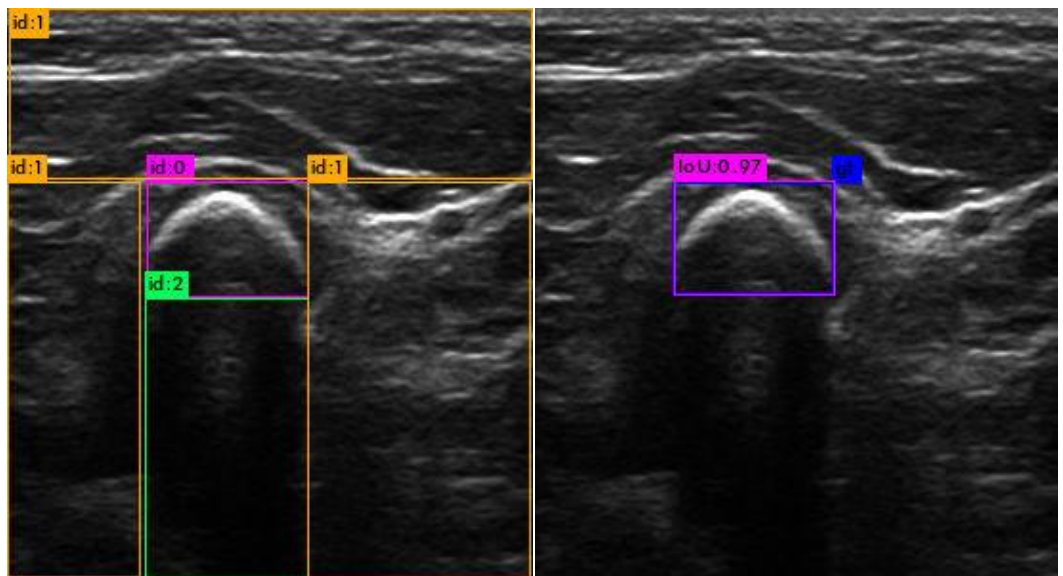
Gambar 4.11 adalah citra uji U2 dengan perolehan IoU yang paling tinggi pada *network detector*  $weights_{75}$  memiliki nilai 0.97 dibandingkan dengan citra uji yang sama hasil *network detector*  $weights_{50}$  memiliki nilai 0.86. Gambar 4.11 (a,b) adalah hasil deteksi menggunakan  $weights_{50}$ , dan Gambar 4.11 (c,d) adalah hasil deteksi menggunakan  $weights_{75}$ . Gambar 4.11 (a,c) menampilkan hasil

deteksi seluruh objek sedangkan Gambar 4.11 (b,d) hanya menampilkan hasil deteksi objek tulang dan *ground truth* tulang.



(a)

(b)



(c)

(d)

Gambar 4.11 Nilai IoU Tertinggi Untuk Skenario Uji U2

#### 4.4 Perbandingan Metode

Arsitektur YoloV3 yang digunakan pada penelitian ini dibandingkan dengan metode terdahulu yang lebih populer yaitu Faster R-CNN. Dimana terdapat perbedaan yang signifikan diantara kedua asitektur tersebut. Faster R-CNN membutuhkan 2 tahapan untuk dapat menentukan lokasi objek. Tahapan pertama adalah mencari sekumpulan RoI yang dianggap sebagai objek dan tahapan kedua adalah mengklasifikasi objek tersebut dan memprediksi lokasinya. Sedangkan pada arsitektur YoloV3 hanya melalui 1 tahapan dalam menentukan lokasi objek, dimana deteksi objek dan prediksi lokasi merupakan satu keluaran akhir arsitektur CNN.

##### 4.4.1 Penambahan Dataset

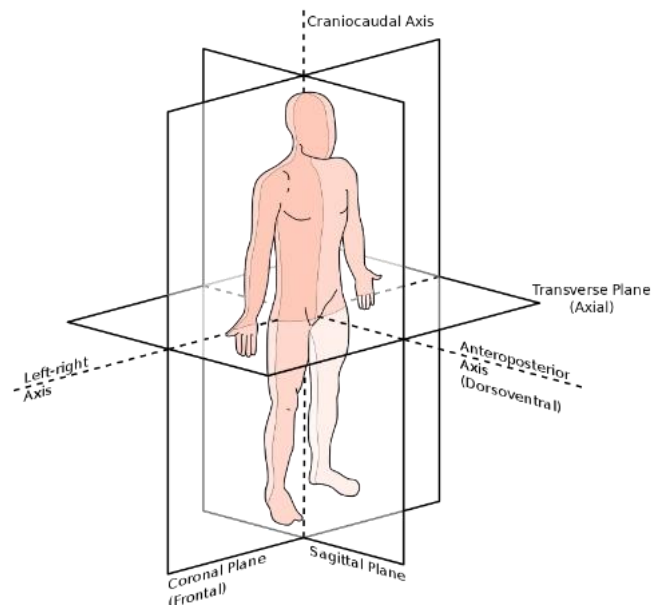
Pada perbandingan metode ini, data yang digunakan sebagai data latih dan data uji adalah seluruh dataset yang disajikan pada Tabel 4.1. Ditambah dengan sub dataset kelompok H dan I, dimana pada sub dataset H dan I memiliki karakteristik yang berbeda dibandingkan dengan sub dataset kelompok yang lain. Sehingga dataset yang digunakan pada perbandingan metode pada penelitian disajikan pada Tabel 4.12. Sehingga jumlah keseluruhan citra yang digunakan untuk data latih sebanyak 2268, data evaluasi 252, data uji pertama 740, data uji kedua 600.

Tabel 4.12 Penambahan Dataset Untuk Perbandingan Metode

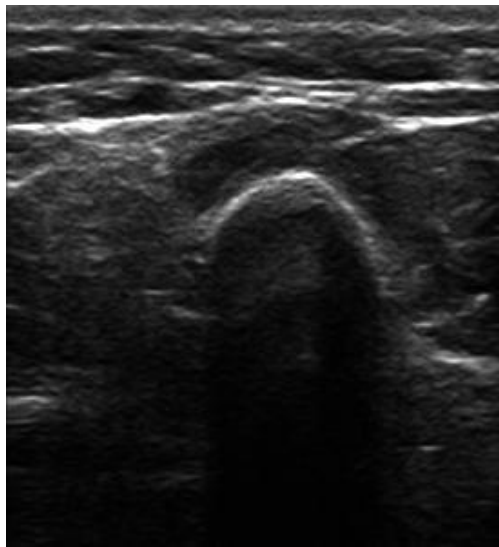
Subdata	Data Pelatihan		Data Pengujian
	Data Latih	Data Evaluasi	
A	540	60	120
B	450	50	100
C	360	40	80
D	450	50	100
E	54	6	-
F	180	20	100
G	-	-	120
H	234	26	120
I	-	-	600
Jumlah	2268	252	740

Pada Gambar 4.12 (a) adalah ilustrasi cara pengambilan data dari berbagai sisi pada bagian tubuh manusia, gambar diambil dari situs [https://commons.wikimedia.org/wiki/File:Anatomical\\_Sagittal\\_Plane-en.svg](https://commons.wikimedia.org/wiki/File:Anatomical_Sagittal_Plane-en.svg). Dan

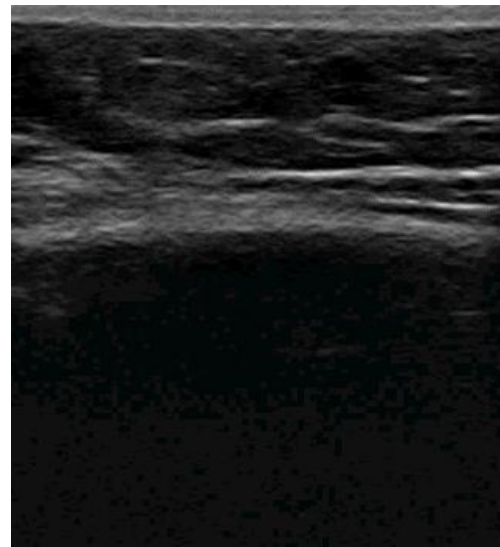
Gambar 4.12 (b) merupakan representasi tulang lengan dengan posisi probe *transverse plane*, Gambar 4.12.(c) merupakan representasi tulang lengan dengan posisi probe *sagittal plane*.



(a)



(b)



(c)

Gambar 4.12 Bidang Pengambilan Data Citra USG

Pada dataset kelompok A, B, C, D, E, F, dan G; representasi citra USG spesimen tulang manusia diambil menggunakan teknik *freehand scanning* dengan posisi probe tegak lurus dengan lengan tangan (*transverse plane*). Sedangkan untuk dataset tambahan di kelompok H dan I, spesimen tulang diambil dengan posisi probe sejajar dengan tulang manusia (*sagittal plane*). Pada Gambar 4.12 menggambarkan contoh dari representasi citra USG yang diambil menggunakan probe dengan posisi *transverse plane* dan *sagittal plane*.

#### 4.4.2 Pelatihan Sistem Dataset Tambahan

Pada arsitektur YoloV3, *backbone* jaringan yang dipakai adalah *Darknet53* dan konfigurasi yang digunakan adalah pelatihan dengan threshold  $IoU_{75}$ , hal ini dikarenakan pada pengujian sebelumnya, pelatihan dengan threshold  $IoU_{75}$  memberikan nilai mAP yang relatif lebih baik. Sedangkan pada arsitektur Faster R-CNN *backbone* jaringan yang digunakan adalah VGG16 dan threshold IoU yang dipakai adalah  $IoU_{70}$ . Pelatihan dihentikan ketika nilai average loss pada masing-masing arsitektur sudah memasuki nilai  $\pm 0.2$ . Hal ini didasarkan dari hasil pelatihan sebelumnya, dimana pada nilai tersebut, sistem sudah mampu mengenali data evaluasi dengan nilai mAP yang baik.

#### 4.4.3 Pengujian Dataset Tambahan

*Metric evaluation* yang diukur pada perbandingan ini adalah nilai  $mAP_{50}$ ,  $mAP_{75}$ , dan  $mAP_{50:95}$ . Kemudian dilakukan perbandingan ketepatan hasil deteksi melalui pengamatan visual untuk membandingkan hasil deteksi di beberapa citra uji antara YoloV3 dan Faster R-CNN.

Pengujian dilakukan dua kali, yang pertama pengujian dilakukan kepada seluruh dataset kelompok A, B, C, D, E, F, G, dan H. Dan yang pengujian yang kedua dilakukan kepada kelompok dataset I. Pengujian pertama disebut dengan data uji  $U_3$ , dimana sebagian sampel dataset kelompok tersebut digunakan juga sebagai data pelatihan. Sedangkan pengujian yang kedua disebut dengan data uji  $U_4$ , dimana tidak satupun bagian dari dataset  $U_4$  digunakan sebagai citra pelatihan.

Pembagian kedua skenario uji ini untuk menganalisis kemampuan sistem dalam mendeteksi objek yang sebagian datanya digunakan sebagian data latih, begitu juga kepada data yang sama sekali bagian datanya tidak pernah digunakan sebagai data latih.

Pada Tabel 4.13 ditampilkan nilai perbandingan mAP antara kedua arsitektur tersebut untuk seluruh skenario uji.

Tabel 4.13 Nilai mAP perbandingan YoloV3 dan Faster R-CNN

Skenario Uji Arsitektur	$U_3$		$U_4$	
	FR-CNN	YoloV3	FR-CNN	YoloV3
$mAP_{50}$	87.81	96.68	83.34	98.91
$mAP_{55}$	84.49	96.68	78.49	97.53
$mAP_{60}$	76.39	96.44	65.12	95.41
$mAP_{65}$	62.95	95.44	46.34	92.16
$mAP_{70}$	44.03	93.77	34.18	83.09
$mAP_{75}$	31.06	91.35	29.75	68.84
$mAP_{80}$	24.28	87.71	28.32	53.69
$mAP_{85}$	22.99	76.67	28.19	33.53
$mAP_{90}$	23.08	55.33	28.41	10.89
$mAP_{95}$	23.11	19.26	28.43	1.37

Dapat dianalisis dari Tabel 4.13, arsitektur YoloV3 memiliki nilai mAP yang lebih baik dibandingkan Faster R-CNN hampir di seluruh threshold mAP. Hanya pada di  $mAP_{90}$  untuk skenario uji  $U_3$  Faster R-CNN memiliki nilai yang lebih baik dibandingkan YoloV3. Dan pada  $mAP_{90}$  dan  $mAP_{95}$  untuk skenario uji  $U_4$  Faster R-CNN memiliki nilai yang lebih baik dibandingkan YoloV3. Hal ini disebabkan, pada Faster R-CNN tahapan untuk melakukan deteksi objek dengan mengumpulkan beberapa calon objek di beberapa RoI, sehingga Faster R-CNN bisa lebih adaptif dalam menghadapi dataset yang baru dan mengumpulkan lebih banyak kandidat objek dibandingkan dengan YoloV3.

Sehingga jika diukur pada parameter  $mAP_{50}$ ,  $mAP_{75}$  dan  $mAP_{50:95}$ , perbandingan di seluruh skenario uji ditampilkan pada Tabel 4.14.

Tabel 4.14 Perbandingan mAP antara skenario uji  $U_3$  dan  $U_4$  untuk semua arsitektur

Skenario Uji Arsitektur	FR-CNN		YoloV3	
	$U_3$	$U_4$	$U_3$	$U_4$
$mAP_{50}$	87.81	83.34	96.68	98.91
$mAP_{75}$	31.06	29.75	91.35	68.84
$mAP_{50:95}$	48.02	45.06	80.93	63.54

Dari Tabel 4.14 dapat dianalisis bahwa sistem memiliki nilai mAP yang lebih tinggi jika diujikan kepada data uji yang sebagian datanya pernah dijadikan sebagai bahan pelatihan (data uji  $U_3$ ), sedangkan untuk ke data uji yang sama sekali tidak pernah dijadikan bahan pelatihan, sistem menghasilkan nilai mAP yang lebih rendah, hal ini disebabkan sistem belum mempelajari secara detail beberapa *feature* yang mungkin saja berbeda yang terdapat pada data uji  $U_4$ .

#### 4.4.4 Pengamatan Visual

Hasil dari deteksi menggunakan arsitektur YoloV3 dan Faster R-CNN diamati secara visual. Dari 740 citra uji pada data uji  $U_3$ , Faster R-CNN memiliki prosentase keberhasilan dalam mendeteksi tulang sebesar 95.54%, sedangkan untuk YoloV3 prosentase keberhasilannya 100%. Untuk metode Faster R-CNN, terdapat beberapa citra uji yang sama sekali salah hasil deteksinya dan terdapat juga beberapa citra uji yang mendeteksi tulang di area yang tidak terdapat representasi tulang. Sedangkan YoloV3 sama sekali tidak ada hasil deteksi yang salah.

Sedangkan untuk data uji  $U_4$  yang berjumlah 600 citra, Faster R-CNN memiliki prosentase keberhasilan dalam mendeteksi objek tulang sebesar 92.33% dan YoloV3 memiliki prosentase keberhasilan sebesar 98.33%. Untuk data uji ini, terdapat beberapa citra yang tidak dapat dideteksi oleh YoloV3, hal ini berarti YoloV3 tidak mengenali beberapa *feature* baru yang terdapat pada data uji yang sama sekali datanya tidak ada pada data pelatihan.

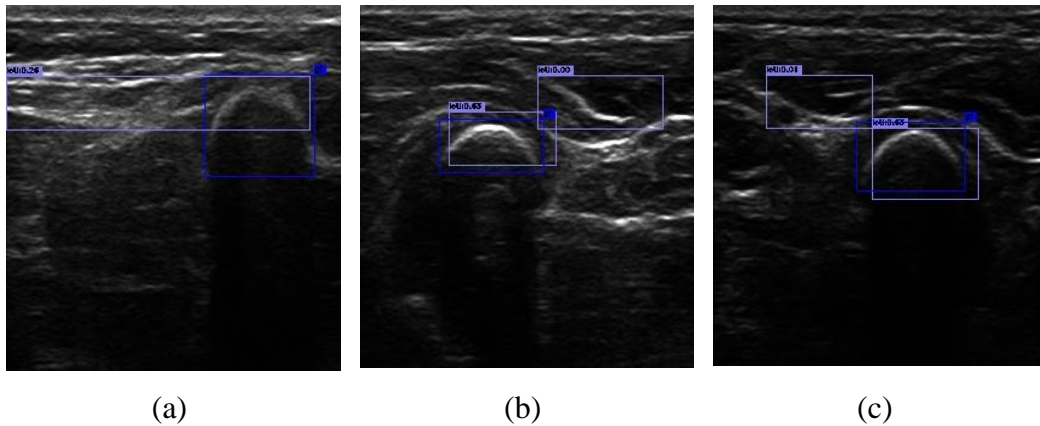
Pada Table 4.15 disajikan jumlah hasil deteksi antara YoloV3 dan Faster R-CNN di seluruh skenario uji.



Tabel 4.15 Jumlah Hasil Deteksi YoloV3 dan Faster R-CNN

Arsitektur	Citra Uji	Salah	Benar
Faster R-CNN	740 ( $U_3$ )	33	707
YoloV3	740 ( $U_3$ )	0	740
Faster R-CNN	600 ( $U_4$ )	46	554
YoloV3	600 ( $U_4$ )	10	590

Pada Gambar 4.13 disajikan beberapa contoh hasil deteksi tulang yang salah dari Faster R-CNN dengan metode pengambilan spesimen pada *transverse plane*. Gambar 4.13 (a,b,c) adalah deteksi yang salah menggunakan metoder Faster R-CNN pada *transverse plane*.

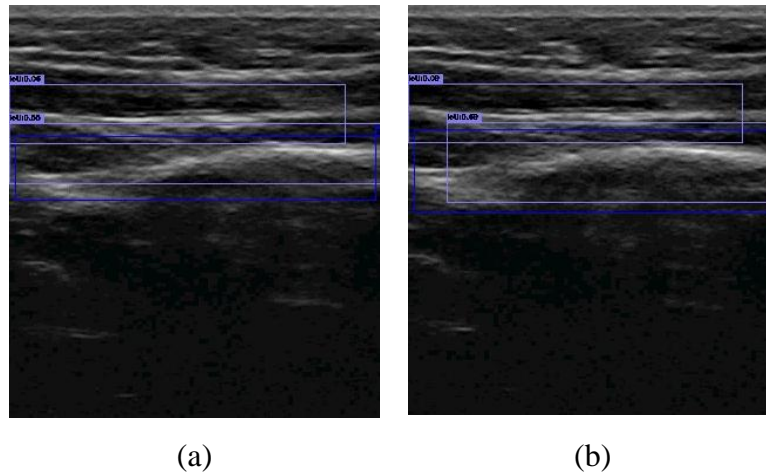


Gambar 4.13 Hasil Deteksi pada *Transverse Plane* Faster R-CNN

Kesalahan deteksi tersebut dikarenakan adanya representasi citra yang intensitas piksel dan bentuknya menyerupai objek tulang. Di Gambar 4.13 (a) terdapat representasi tulang yang menyerupai objek tulang yang diambil dari probe pada bidang *sagittal*. Sedangkan di Gambar 4.13 (b) dan (c), sistem mendeteksi adanya dua tulang pada satu citra. Hal tersebut dikarenakan juga terdapat representasi citra yang intensitas piksel dan bentuknya menyerupai objek tulang yang diambil pada bidang *transverse*.

Pada Gambar 4.14 di bawah ini disajikan contoh hasil deteksi tulang yang salah dari Faster R-CNN dengan metode pengambilan spesimen pada *sagittal plane*. Sistem mendeteksi area di atas tulang yang menyerupai objek tulang bidang *sagittal*. Sehingga dalam satu citra terdapat dua objek tulang, padahal sebenarnya

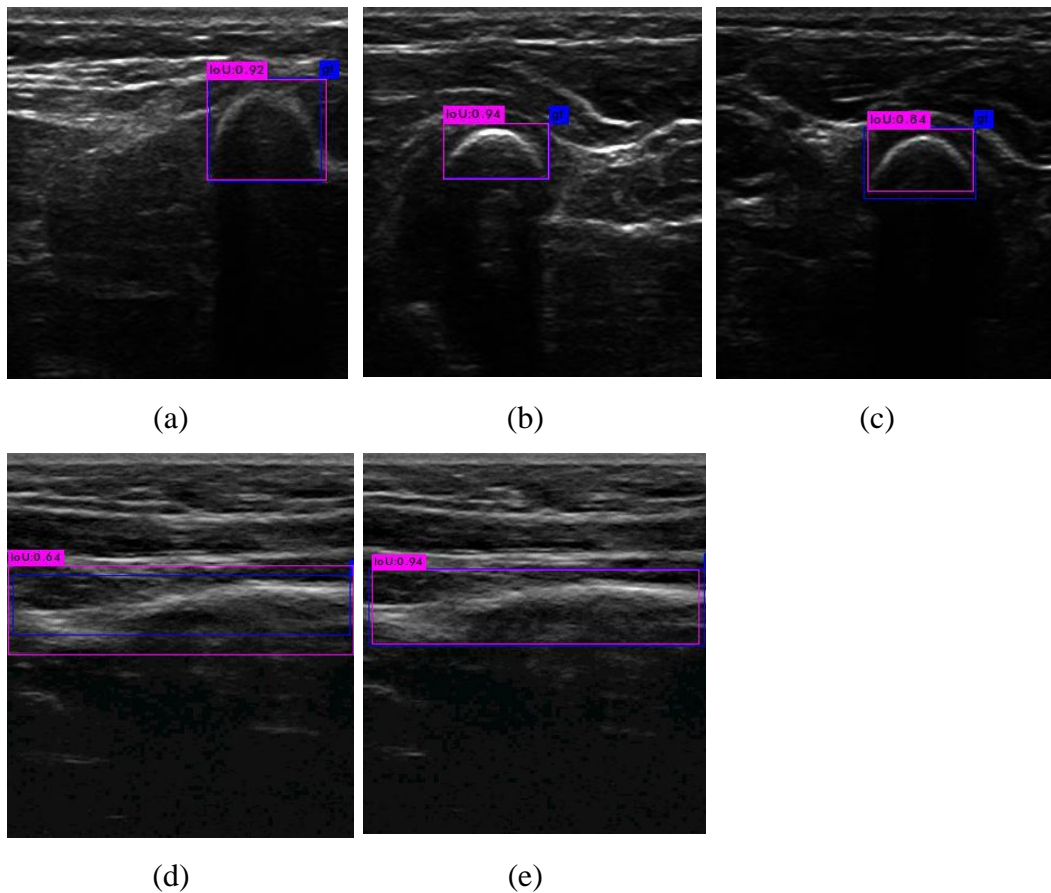
pada citra tersebut hanya mengandung satu objek tulang lengan manusia. Gambar 4.14 (a,b,c) adalah deteksi yang salah menggunakan metode Faster R-CNN pada *sagittal plane*.



Gambar 4.14 Hasil Deteksi pada *Sagittal Plane* Faster R-CNN

Deteksi yang salah pada citra uji menggunakan metode Faster R-CNN disebabkan pada metode ini dalam melakukan deteksi objek melalui 2 tahapan. Sehingga pada tahapan pertama, sistem mengenali beberapa area yang terindikasi sebagai objek tulang. Kemudian area-area yang diusulkan tersebut ketika diklasifikasi memiliki nilai probabilitas sebagai objek tulang yang sangat besar dikarenakan intensitas piksel dan bentuknya yang menyerupai objek tulang. Hal ini tidak terjadi pada metode YoloV3, dikarenakan metode YoloV3 mendeteksi objek dalam satu tahapan jaringan, yang di akhir layer jaringannya langsung mengeluarkan nilai *bounding box*, *objectness*, dan *score probability*.

Pada Gambar 4.15 adalah disajikan beberapa hasil deteksi dari YoloV3 dengan citra uji yang sama, dimana pada metode Faster R-CNN mendeteksi lokasi tulang yang salah. Gambar 4.15 (a,b,c,d,e) adalah hasil deteksi YoloV3 pada citra uji yang sama yang disajikan pada Gambar 4.13 dan 4.14, dimana gambar ini dideteksi sebagai objek yang salah menggunakan metode Faster R-CNN.



Gambar 4.15 Hasil Deteksi YoloV3

Hasil pengamatan secara visual, juga menghitung jumlah objek yang terdeteksi berdasarkan nilai IoU yang didapatkan. Arsitektur YoloV3 memiliki nilai prediksi dengan IoU yang sangat tinggi dibandingkan dengan arsitektur Faster R-CNN. Berdasarkan dari penilaian *metric evaluation* dan pengamatan visual hasil uji, metode YoloV3 memiliki hasil yang lebih baik dibandingkan dengan metode Faster R-CNN. Hal ini disebabkan YoloV3 memandang utuh sebuah objek dari keseluruhan citra sedangkan Faster R-CNN mengolah informasi citra secara parsial berdasarkan area yang diajukan sebagai kandidat objek tulang.

Pada Tabel 4.16 disajikan data jumlah deteksi yang berhasil dilakukan oleh Faster R-CNN dan YoloV3 diurutkan sesuai dengan nilai IoU yang didapatkan di setiap deteksi. Arsitektur YoloV3 memiliki nilai IoU yang sangat tinggi di atas 0.80, sedangkan Faster R-CNN hasil deteksi yang banyak berada di nilai IoU antara 0.65 s.d 0.80.

Tabel 4.16 Jumlah deteksi Faster R-CNN dan YoloV3 berdasarkan nilai IoU

IoU	Faster R-CNN	YoloV3
0 0.5	11	0
0.50 0.55	33	0
0.55 0.60	48	5
0.60 0.65	116	13
0.65 0.70	145	22
0.70 0.75	137	36
0.75 0.80	114	50
0.80 0.85	85	134
0.85 0.90	29	231
0.90 0.95	9	212
0.95 1.00	0	45

## BAB 5

### KESIMPULAN

Tahapan pelatihan sistem dibagi menjadi dua skenario. Dimana skenario pertama menggunakan threshold IoU  $> 0.5$  (P1) dan skenario kedua menggunakan threshold IoU  $> 0.75$  (P2). Kedua skenario ini dipakai untuk mengukur mengevaluasi hasil bobot dan bias di kedua skenario tersebut dalam melakukan deteksi ke citra uji.

Hasil dari selama pelatihan berlangsung, untuk nilai average loss di seluruh iterasi skenario P1 memiliki nilai yang lebih kecil dibandingkan dengan skenario P2. Hal ini dikarenakan untuk setiap deteksi objek selama pada fase *feed-forward* pada masa pelatihan jika nilai IoU  $> 0.5$  dianggap sebagai nilai yang benar dan nilainya tidak dihitung sebagai error untuk memperbaiki parameter bobot dan bias dan di fase *backpass network*.

Pada iterasi terakhir pelatihan sistem, nilai *average loss* skenario P1 dan P2 tidak jauh berbeda berada di nilai 0.1134 dan 0.1324 untuk masing-masing skenario uji. Selisih diantara keduanya hanya 0.019. Hal ini menunjukkan, di semua skenario pelatihan hingga iterasi terakhir sistem berhasil beradaptasi dengan nilai *error* di tiap iterasi sehingga mampu menuju nilai selisih *error* yang paling kecil.

Parameter bobot dan bias hasil pelatihan sistem P1 ( $weights_{50}$ ) dan P2 ( $weights_{75}$ ) digunakan untuk mendeteksi kelompok citra uji. Hasil akhir dari tiap deteksi untuk masing-masing nilai bobot dan bias di semua skenario uji dapat dengan baik mendeteksi objek tulang.

Hal ini dapat dilihat dari pengukuran nilai  $mAP_{50}$ ,  $mAP_{75}$ , dan  $mAP_{50:95}$  untuk  $weights_{50}$  pada citra uji kelompok UI (A,B,C,D,F) memiliki nilai masing-masingnya 99.98, 97.68, dan 85.67. Sedangkan pengujian untuk kelompok U2 (G) memiliki nilai masing-masingnya 99.91, 81.98, dan 74.23.

Dan untuk  $weights_{75}$  pada citra uji kelompok UI (A,B,C,D,F) memiliki nilai masing-masingnya 99.96, 97.46, dan 86.35. Sedangkan pengujian untuk kelompok U2 (G) memiliki nilai masing-masingnya 99.73, 83.39, dan 75.27.

Dari beberapa nilai parameter pengujian yang didapatkan dapat disimpulkan bahwa parameter bobot dan bias  $weights_{75}$  memiliki hasil yang lebih baik dibandingkan dengan  $weights_{50}$ . Kesimpulan tersebut diambil sebab nilai  $mAP_{50:95}$  menggunakan  $weights_{75}$  untuk semua skenario uji bernilai lebih besar dibandingkan  $weights_{50}$ .

Arsitektur YoloV3 kemudian dibandingkan dengan metode Faster R-CNN, metode YoloV3 memiliki hasil yang lebih baik, untuk pengujian di seluruh citra uji yang berjumlah 740 data. YoloV3 berhasil mendeteksi lokasi secara tepat spesimen objek tulang manusia dengan tingkat keberhasilan deteksi sebesar 100%, sedangkan jika menggunakan metode Faster R-CNN, tingkat keberhasilannya hanya 95.54%.

## DAFTAR PUSTAKA

- [1] R.J Joseph, D. Santosh, G. Ross, F. Ali, "You Only Look Once: Unified, Real Time Object Detection," *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.
- [2] K. Tita, Y. Eko Mulyanto, P. I Ketut Eddy, P. Mauridhi Hery, "Automatic Bone Outer Contour Extraction from B-Modes Ultrasound Images Based on Local Phase Symmetry and Quadratic Polynomial Fitting," *SPIE, Second International Workshop on Pattern Recognition*, 2017.
- [3] Li. You, "Segmentation of Medical Ultrasound Images Using Convolutional Neural Networks with Noisy Activating Functions," *Department of Bioengineering, Stanford University*, 2016.
- [4] Wu, Jianxin, "Introduction to Convolutional Neural Networks," *LAMDA Group, National Key Lab for Novel Software Technology, Nanjing University*, 2017.
- [5] C.-C, Jay Kuo, "Understanding Convolutional Neural Networks with A Mathematical Model," *CoRR, Cornell University Library*, 2017.
- [6] K. Tita, Y. Eko Mulyanto, P. I Ketut Eddy, P. Mauridhi Hery, "Deteksi Region of Interest Tulang pada Citra B-mode secara Otomatis menggunakan Region Proposal Networks," *JNTETI*, Vol. 8, No. 1, 2019.
- [7] D.A. Dharmawan, "Deteksi Kanker Serviks Otomatis Berbasis Jaringan Syaraf Tiruan LVQ dan DCT," *J. Nas. Tek. Elektro dan Teknol. Inf.* Vol. 3, No. 4, hal 269-272, 2014.
- [8] P.J.S. Gonçalves dan P. Torres, "Extracting Bone Contours in Ultrasound Images: Energetic Versus Probabilistic Methods," *Rom. Rev. Precis. Mech. Opt. Mechatronics*. Vol. 20, No. 37, hal. 105–110, 2010.
- [9] J. Kowal, C. Amstutz, F. Langlotz, H. Talib, dan M.G. Ballester, "Automated Bone Contour Detection in Ultrasound B-Mode Images For Minimally Invasive Registration in Computer-Assisted Surgery – An In Vitro Evaluation," *Int. J. Med. Robot. Comput. Assist. Surg. MRCAS*, Vol. 3, No. 4, hal. 341–348, 2007.
- [10] V. Chan dan A. Perlas, "Basics of Ultrasound Imaging," in *Atlas Ultrasound-Guided Proced. Interv. Pain Manag.*, S.N. Narouze, Ed., Toronto, ON, Canada, Springer Science+Business Media, 2011, hal. 13– 20.

- [11] R. Jia, S.J. Mellon, S. Hansjee, A.P. Monk, D.W. Murray, dan J.A. Noble, "Automatic Bone Segmentation in Ultrasound Images Using Local Phase Features and Dynamic Programming," *IEEE 13th Int. Symp. Biomed. Imaging*, 2016, hal. 1005–1008.
- [12] S. Ren, K. He, R. Girshick, dan J. Sun, "Faster R-CNN: Towards Real- Time Object Detection with Region Proposal Networks," *ArXiv Prepr. ArXiv1506.01497.*, Vol. 74, hal. 1–14, 2015..
- [13] T. Carneiro, R. Victor, M. Da, T. Nepomuceno, G. Bian, dan V.H.C.D.E. Albuquerque, "Performance Analysis of Google Colaboratory as a Tool for Accelerating Deep Learning Applications," *IEEE Access Trends, Perspect. Prospect. Mach. Learn. Appl. to Biomed. Syst. Internet Med. Things.*, Vol. 6, hal. 61677–61685, 2018.
- [14] R.J Joseph, F. Ali, "YOLO9000: Better, Faster, Stronger," arXiv: 1612.08242 [cs.CV], 2016.
- [15] R.J Joseph, D. Santosh, G. Ross, F. Ali, "YOLOv3: An Incremental Improvement," arXiv: 1804.02767 [cs.CV], 2018.
- [16] N.P. Husain dan C. Fatichah, "Segmentasi Citra Sel Tunggal Smear Serviks Menggunakan Radiating Component Normalized Generalized GVFS," *J. Nas. Tek. Elektro dan Teknol. Inf.*, Vol. 6, No. 1, hal. 107– 114, 2017.
- [17] N. Syakrani, Y. Widhiyasana, dan A.A. Efendi, "Deteksi Tumor Hati dengan Graph Cut dan Taksiran Volume Tumornya," *J. Nas. Tek. Elektro dan Teknol. Inf.*, Vol. 7, No. 1, hal. 35-43,2018.
- [18] O. Herliana, T.S. Widodo, dan I. Soesanti, "Klasifikasi Nonsupervised Citra Thermal Kanker Payudara Berbasis Fuzzy C-MEANS," *J. Nas. Tek. Elektro dan Teknol. Inf.*, Vol. 1, No. 3, hal. 55-59, 2012.



## BIODATA PENULIS



R. Arif Firdaus Lazuardi, lahir di Surabaya pada tanggal 01 Desember 1988. Menyelesaikan pendidikan sekolah dasar di SDN Dr. Sutomo V Surabaya pada tahun 2000. Kemudian melanjutkan ke Pondok Pesantren Darul Ulum Jombang dengan mengenyam pendidikan di bangku SLTPN 3 Peterongan Jombang hingga tahun 2003. Pendidikan menengah ke atas ditempuh di Pondok Modern Darussalam Gontor Ponorogo (Gontor 1)

dengan menggunakan kurikulum KMI (*Kulliyatul Muallimin al Islamiyah*) dan selesai pada tahun 2007. Melaksanakan pengabdian sebagai guru di Pondok Modern Darul Qiyam (Gontor 6) selama 1 tahun sebagai syarat untuk menyelesaikan pendidikan KMI. Dua tahun setelah lulus dari Gontor, pada tahun 2009 meneruskan jenjang S1 di Jurusan Matematika Fakultas Matematika dan Ilmu Pengetahuan Alam (FMIPA) Institut Teknologi Sepuluh Nopember (ITS) Surabaya, mengambil bidang minat Ilmu Komputer dan selesai pendidikan S1 selama 10 semester pada tahun 2014. Sembari bekerja di PT. Infoglobal Teknologi Semesta sebagai *software engineer*, meneruskan pendidikan S2 di Departemen Teknik Elektro mulai tahun 2016 semester genap, mengambil konsentrasi bidang Jaringan Cerdas Multimedia dan selesai selama 5 semester pada tahun 2019. Penulis dapat dihubungi pada email lazuardi.dp@gmail.com.

*Halaman ini sengaja dikosongkan*