



**KLASIFIKASI FUNDUS *DIABETIC RETINOPATHY* MENGGUNAKAN  
*DEEP LEARNING***



**Oleh:**  
**GUSTI RAFI AFKARIANSYAH**  
**18410200008**

---

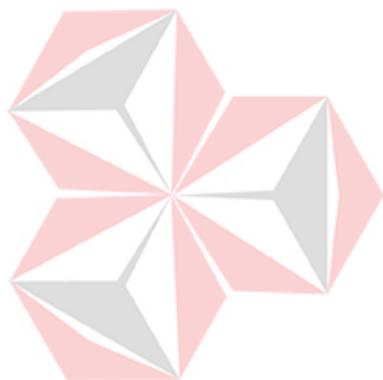
---

**FAKULTAS TEKNOLOGI DAN INFORMATIKA**  
**UNIVERSITAS DINAMIKA**  
**2022**

**KLASIFIKASI FUNDUS *DIABETIC RETINOPATHY* MENGGUNAKAN  
DEEP LEARNING**

**TUGAS AKHIR**

**Diajukan sebagai salah satu syarat untuk menyelesaikan  
Program Sarjana Teknik**



UNIVERSITAS  
**Dinamika**  
Disusun Oleh:

**Nama : Gusti Rafi Afkariansyah**  
**NIM : 18410200008**  
**Program Studi : S1 Teknik Komputer**

**FAKULTAS TEKNOLOGI DAN INFORMATIKA  
UNIVERSITAS DINAMIKA  
2022**

## TUGAS AKHIR

### KLASIFIKASI FUNDUS *DIABETIC RETINOPATHY* MENGGUNAKAN DEEP LEARNING

Dipersiapkan dan disusun oleh:

**Gusti Rafi Afkariansyah**

**NIM : 18410200008**

Telah diperiksa, dibahas dan disetujui oleh Dewan Pembahas

Pada: 12 Januari 2022

#### Susunan Dewan Pembahas

##### Pembimbing:

I. **Dr. Jusak**

NIDN: 0708017101

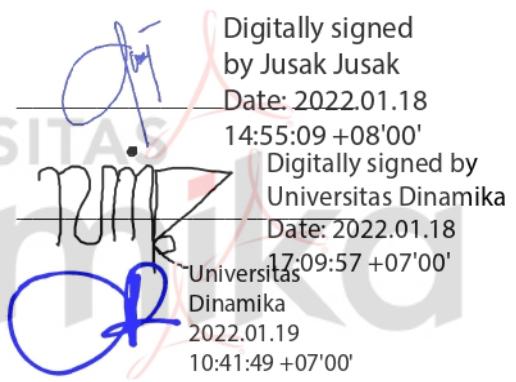
II. **Heri Pratikno, M.T., MTCNA., MTCRE.**

NIDN: 0716117302

##### Pembahas:

**Pauladie Susanto, S.Kom., M.T.**

NIDN: 0729047501

  
Digitally signed by Jusak Jusak  
Date: 2022.01.18 14:55:09 +08'00'  
Digitally signed by Universitas Dinamika  
Date: 2022.01.18 17:09:57 +07'00'  
Universitas Dinamika  
2022.01.19 10:41:49 +07'00'

Tugas Akhir ini telah diterima sebagai salah satu persyaratan

untuk memperoleh gelar Sarjana

  
Digitally signed by  
Universitas Dinamika  
Date: 2022.01.26 08:52:28 +07'00'

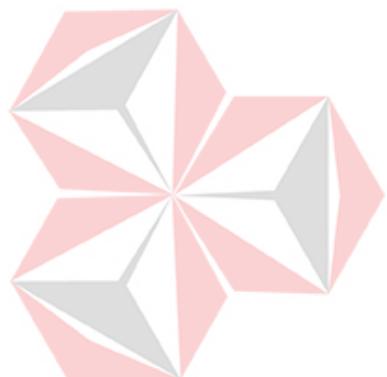
**Tri Sagirani, S.Kom., M.MT.**

NIDN: 0731017601

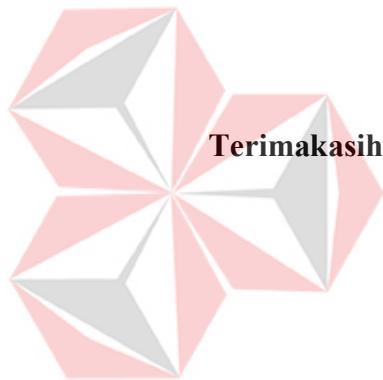
Dekan Fakultas Teknologi dan Informatika

UNIVERSITAS DINAMIKA

**Controlling something is enjoyable, but controlling mortals is Zufriedenheit.**



UNIVERSITAS  
**Dinamika**



Terimakasih atas segala macam bentuk dukungan yang telah ayah dan ibu  
berikan, Tugas Akhir ini demi kalian berdua.

UNIVERSITAS  
**Dynamika**

**SURAT PERNYATAAN**  
**PERSETUJUAN PUBLIKASI DAN KEASLIAN KARYA ILMIAH**

Sebagai mahasiswa Universitas Dinamika, saya :

Nama : Gusti Rafi Afkariansyah

NIM : 18410200008

Program Studi : S1 Teknik Komputer

Fakultas : Fakultas Teknologi dan Informatika

Jenis Karya : Laporan Tugas Akhir

Judul Karya : **KLASIFIKASI FUNDUS DIABETIC RETINOPATHY MENGGUNAKAN DEEP LEARNING.**

Menyatakan dengan sesungguhnya bahwa:

1. Demi pengembangan Ilmu Pengetahuan, Teknologi dan Seni, saya menyetujui memberikan kepada Universitas Dinamika Hak Bebas Royalti Non-Eksklusif (*Non-Exclusive Royalty Free Right*) atas seluruh isi/ sebagian karya ilmiah saya tersebut di atas untuk disimpan, dialihmediakan dan dikelola dalam bentuk pangkalan data (*database*) untuk selanjutnya didistribusikan atau dipublikasikan demi kepentingan akademis dengan tetap mencantumkan nama saya sebagai penulis atau sebagai pemilik pencipta dan Hak Cipta
2. Karya tersebut di atas adalah karya asli saya, bukan plagiat baik sebagian maupun keseluruhan. Kutipan, karya atau pendapat orang lain yang ada dalam karya ilmiah ini adalah semata hanya rujukan yang dicantumkan dalam Daftar Pustaka saya
3. Apabila dikemudian hari ditemukan dan terbukti terdapat tindakan plagiat pada karya ilmiah ini, maka saya bersedia untuk menerima pencabutan terhadap gelar kesarjanaan yang telah diberikan kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Surabaya, 5 Januari 2022

Yang menyatakan,



**Gusti Rafi Afkariansyah**  
NIM : 18410200008

## ABSTRAK

*Diabetic Retinopathy* adalah salah satu penyakit yang terjadi pada seseorang yang menderita penyakit diabetes, penyakit ini menimbulkan kerusakan pada mata penderita, yang apabila tidak ditangani sejak dini akan mengakibatkan kebutaan. Penelitian sebelumnya yang dilakukan oleh Dinda Ulima R.Y., meneliti *Diabetic Retinopathy* dengan mengklasifikasi tingkat keparahan dari NPDR (*Non-Proliferative Diabetic Retinopathy*) yang dibagi menjadi 2 tingkatan yaitu *moderate* (sedang) dan *severe* (parah). Penelitian ini mengklasifikasi tingkat keparahan pada NPDR berdasarkan *hard exudate* menggunakan ELM (*Extreme Learning Machine*), input yang digunakan citra fundus retina adalah yang memiliki *exudate* dan teridentifikasi *moderate* NPDR atau *severe* NPDR. Pada Tugas Akhir ini, peneliti melakukan klasifikasi fundus *diabetic retinopathy* dengan menggunakan lima model arsitektur *pre-trained model deep learning*, yaitu: *VGG-16*, *VGG-19*, *ResNet-101*, *ResNet-50* dan *Inception V3*. Dengan melakukan komparasi pada kelima *pre-trained model* ini maka akan diketahui *pre-trained model* mana yang memiliki akurasi dan tingkat komputasi terbaik.

Adapun hasil pengujian pada Tugas Akhir ini, *pre-trained model network Inception V3* mempunyai akurasi tertinggi, yaitu: sebesar 77,99% dengan waktu komputasi 15.594 detik, sedangkan yang memiliki akurasi terendah adalah *ResNet-101*, yaitu sebesar 73,67% dengan waktu komputasinya 14.987 detik. Pada penelitian ini, penulis menggunakan lima kelas yaitu: *Non-Diabetic Retinopathy*, *Mild*, *Moderate*, *Severe*, dan *Poliferative Diabetic Retinopathy*.

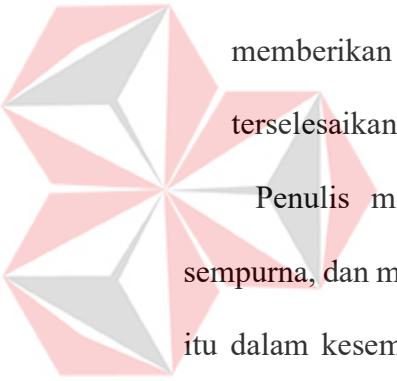
**Kata kunci :** *Diabetic Retinopathy*, *Deep Learning*, *Pre-trained model network*.

## KATA PENGANTAR

Dengan mengucapkan puji dan syukur terhadap kehadiran Tuhan Yang Maha Esa yang telah memberikan limpahan berkah dan rahmat-Nya, serta segala karunia yang telah diberikan kepada penulis, sehingga penulis dapat merampungkan Laporan Tugas Akhir yang berjudul *”KLASIFIKASI FUNDUS DIABETIC RETINOPATHY MENGGUNAKAN DEEP LEARNING”*. Laporan Tugas Akhir ini disusun dalam rangka memenuhi salah satu prasyarat dalam menyelesaikan Program Sarjana Teknik Komputer di Universitas Dinamika.

Pada kesempatan yang telah diberikan ini, penulis mengucapkan rasa terima kasih terhadap individu-individu yang memberikan dukungan dan juga saran serta bimbingan dalam upaya untuk menyelesaikan laporan Tugas Akhir ini. Oleh karena itu penulis ingin mengucapkan terima kasih kepada:

1. Orang Tua penulis, yang telah memberikan kontribusi besar berupa dukungan penuh atas apa yang akan penulis lakukan dan juga atas apa yang telah penulis lakukan sehingga dapat menyelesaikan Laporan Tugas Akhir ini.
2. Ibu Tri Sagirani, S.Kom., M.MT., selaku Dekan Fakultas Teknologi dan Informatika (FTI) Universitas Dinamika.
3. Bapak Pauladie Susanto, S.Kom., M.T., selaku Ketua Program Studi S1 Teknik Komputer Universitas Dinamika.
4. Bapak Dr. Jusak selaku Dosen Pembimbing I yang selalu memberikan waktu dan bimbingan serta ilmu dalam menyelesaikan Tugas Akhir beserta laporan ini.

- 
5. Bapak Heri Pratikno, M.T., MTCNA., MTCRE., selaku Dosen Pembimbing II yang juga selalu memberi waktu dan bimbingan dalam menyelesaikan Tugas Akhir beserta laporan ini.
  6. Seluruh Dosen pengajar Program Studi S1 Teknik Komputer Universitas Dinamika yang telah memberikan ilmu, dan juga bimbingan yang berharga dari semester 1 hingga sampai saat ini.
  7. Seluruh rekan-rekan S1 Teknik Komputer angkatan 2018 yang telah memberikan dukungan dan semangatnya untuk membantu penulis untuk menyelesaikan Laporan Tugas Akhir ini.
  8. Dan seluruh pihak yang tidak dapat disebutkan satu per satu yang telah memberikan dukungan serta bantuan dalam segala bentuk yang akhirnya terselesaikannya Laporan Tugas Akhir ini.

Penulis memahami bahwa Laporan Tugas Akhir ini belum mencapai sempurna, dan masih banyak kekurangan dalam menyusun laporan ini. Oleh karena itu dalam kesempatan ini, penulis meminta maaf apabila dalam Laporan Tugas Akhir ini masih terdapat kesalahan baik dalam penulisan maupun Bahasa yang digunakan. Penulis juga memerlukan kritik dan saran dari para pembaca yang sifatnya membangun untuk kesempurnaan laporan yang telah penulis susun.

Surabaya, 15 Desember 2021

Penulis

## DAFTAR ISI

### Contents

BAB I .....	15
PENDAHULUAN.....	15
1.1    Latar Belakang.....	15
1.2    Rumusan Masalah .....	16
1.3    Batasan Masalah.....	16
1.4    Tujuan.....	16
1.5    Manfaat.....	16
BAB II.....	18
LANDASAN TEORI .....	18
2.1 <i>Deep Learning</i> .....	18
2.2    VGG-16 .....	18
2.3    VGG-19 .....	19
2.4    ResNet-101 .....	20
2.5    ResNet-50 .....	20
2.6    Inception V3 .....	21
2.7 <i>Diabetic Retinopathy</i> .....	22
BAB III.....	25
METODOLOGI PENELITIAN.....	25
3.1    Metode Klasifikasi Citra Fundus <i>Diabetic Retinopathy</i> Menggunakan <i>Pre-trained Model Network</i> .....	25
3.1.1    Dataset .....	25
3.1.2    Arsitektur <i>Pre-trained Model Network</i> .....	26
3.1.3 <i>Accuracy</i> .....	28
3.1.4 <i>Loss</i> .....	28

3.1.5	<i>Precision</i> .....	29
3.1.6	<i>Recall</i> .....	29
3.1.7	<i>F1-Score</i> .....	30
BAB IV .....		31
HASIL DAN PEMBAHASAN.....		31
4.1	Pengujian Komparasi 5 <i>Pre-trained Model Network</i> .....	31
4.1.1	Tujuan Pengujian Komparasi 5 <i>Pre-trained Model Network</i> .....	31
4.1.2	Prosedur Pengujian Komparasi 5 <i>Pre-trained Model Network</i> .....	31
4.1.3	Hasil Pengujian Komparasi 5 <i>Pre-trained Model Network</i> .....	31
4.2	Hasil Perolehan Nilai 5 Parameter Uji .....	31
4.2.1	Tujuan Memperoleh Nilai Hasil 5 Parameter Uji .....	31
4.2.2	Prosedur Memperoleh Nilai Hasil 5 Parameter Uji .....	31
4.2.3	Hasil Nilai 5 Parameter Uji .....	32
A.	VGG-16 .....	32
4.3	Hasil evaluasi terhadap nilai dari akurasi dan loss .....	38
4.4	Ranking Pre-Trained Model Network.....	40
BAB V.....		41
PENUTUPAN .....		41
5.1	Kesimpulan.....	41
5.2	Saran .....	42
DAFTAR PUSTAKA .....		43
LAMPIRAN .....		<b>Error! Bookmark not defined.</b>
Lampiran 1     Source Code Program .....		<b>Error! Bookmark not defined.</b>
BIODATA PENULIS .....		<b>Error! Bookmark not defined.</b>

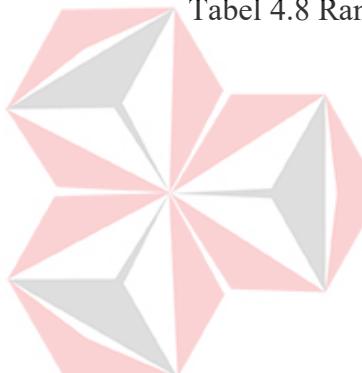
## DAFTAR GAMBAR

Gambar 2.1 Arsitektur ResNet-101.....	20
Gambar 2.2 Arsitektur ResNet-50.....	21
Gambar 2.3 Arsitektur Inception V3.....	22
Gambar 3.1 Metode Klasifikasi Citra Fundus Diabetic Retinopathy.....	25
Gambar 3.2 Diagram alir proses training.....	27
Gambar 4.1 Grafik model akurasi.....	38
Gambar 4.2 Grafik model Loss.....	39



## DAFTAR TABEL

Tabel 2.1 Arsitektur VGG-16.....	18
Tabel 2.2 Arsitektur VGG-19.....	19
Tabel 4.1 Hasil Pengujian Komparasi 5 Pre-trained model network.....	31
Tabel 4.2 Hasil Nilai Parameter VGG-16.....	32
Tabel 4.3 Hasil Nilai Parameter VGG-19.....	33
Tabel 4.4 Hasil Nilai Parameter ResNet-101.....	35
Tabel 4.5 Hasil Nilai Parameter ResNet-50.....	36
Tabel 4.6 Hasil Nilai Parameter Inception V3.....	37
Tabel 4.7 Ranking hasil komparasi nilai akurasi.....	40
Tabel 4.8 Ranking hasil komparasi waktu komputasi.....	40



UNIVERSITAS  
**Dinamika**

## DAFTAR LAMPIRAN

<b>Lampiran 1</b> Source code.....	44
<b>Lampiran 2</b> Hasil Turnitin.....	47



## BAB I

### PENDAHULUAN

#### 1.1 Latar Belakang

*Diabetic Retinopathy* adalah salah satu penyakit mata yang terjadi akibat dari komplikasi yang mempengaruhi mata penderita penyakit diabetes. *Diabetic Retinopathy* terjadi disebabkan oleh kerusakan pada pembuluh darah pada jaringan yang berada pada bagian belakang mata (retina). Salah satu faktor yang dapat mengakibatkan kerusakan ini adalah gula darah yang tidak terkontrol dengan baik.

*Diabetic Retinopathy* mengakibatkan seseorang kehilangan kemampuan penglihatannya dan di keadaan yang parah dapat mengakibatkan kebutaan.

Penelitian sebelumnya, meneliti *Diabetic Retinopathy* dengan mengklasifikasi tingkat keparahan dari NPDR (*Non-Proliferative Diabetic Retinopathy*) yang dibagi menjadi 2 tingkatan yaitu *moderate* (sedang) dan *severe* (parah). Penelitian ini mengklasifikasi tingkat keparahan pada NPDR berdasarkan *hard exudate* menggunakan ELM (*Extreme Learning Machine*), input yang digunakan citra fundus retina adalah yang memiliki *exudate* dan teridentifikasi *moderate* NPDR atau *severe* NPDR (Yani, 2017).

Pada Tugas Akhir ini, digunakan lima metode untuk mengklasifikasi *Diabetic Retinopathy* adalah diantaranya adalah model arsitektur *VGG-16*, *VGG-19*, *ResNet-101*, *ResNet-50* serta model arsitektur *Inception*. Dengan membandingkan lima metode tersebut, dapat membantu dalam mengidentifikasi model arsitektur yang memiliki akurasi tinggi dengan tidak mengorbankan waktu yang dibutuhkan dalam menjalankan sebuah model arsitektur.

## 1.2 Rumusan Masalah

Berlandaskan latar belakang di atas, maka dapat dirumuskan sebuah rumusan masalah pada Tugas Akhir ini sebagai berikut :

Bagaimana melakukan klasifikasi dan mengevaluasi citra fundus *Diabetic Retinopathy* menggunakan lima macam *pre-trained Deep Learning* model dengan parameter uji meliputi *Loss, Accuracy, Precision, F1-Score dan Recall*?

## 1.3 Batasan Masalah

Dalam perancangan Tugas Akhir ini, Pembahasan masalah memiliki sebuah batasan terhadap beberapa hal berikut :

1. Klasifikasi 5 tingkat stadium pada *Diabetic Retinopathy* (*Non-Diabetic Retinopathy, Mild, Moderate, Severe, Proliferative Diabetic Retinopathy*).
2. Dataset yang digunakan adalah *Diabetic Retinopathy* (Resized)

## 1.4 Tujuan

Berlandaskan latar belakang dan rumusan masalah di atas, mendapatkan tujuan pada tugas akhir ini sebagai berikut :

Mampu melakukan klasifikasi dan mengevaluasi citra fundus *Diabetic Retinopathy* menggunakan lima macam *Pre-trained Deep Learning* model dengan parameter uji meliputi *Error-rate, Accuracy, Precision, F1-Score dan Recall*.

## 1.5 Manfaat

Adapun dari Tugas Akhir ini dapat diperoleh sebuah manfaat berikut :

- 1 Dapat menambah pengetahuan dan penerapan sistem klasifikasi mengenai fundus *Diabetic Retinopathy* menggunakan *Deep Learning*.

- 2 Referensi bagi mahasiswa jika melakukan penelitian mengenai fundus *Diabetic Retinopathy*.
- 3 Memudahkan tenaga Kesehatan untuk mendiagnosa dan melakukan tindakan awal dari resiko kebutaan.



## BAB II

### LANDASAN TEORI

#### 2.1 Deep Learning

Deep Learning merupakan salah satu dari bermacam-macam *Machine Learning* yang memiliki sebuah tugas untuk melatih sebuah komputer untuk menjalankan sebuah perintah secara spesifik. Perintah-perintah tersebut dapat berupa deteksi sebuah objek, identifikasi gambar maupun memanifestasikan sebuah prediksi (Arsal et al., 2020).

#### 2.2 VGG-16

*VGG-16* merupakan model dari *CNN (Convolution Neural Network)* dan juga variasi dari *VGG* model yang memiliki 16 layer konvolusi. Dengan 3 *layer fully connected*, 5 *layer max-pooling* dan 1 *layer softmax*. Pada saat training dilakukan, model ini membutuhkan input berupa gambar dengan ukuran pixel sebesar 224 x 224 pixel (Simonyan & Zisserman, 2015).

Tabel 2.1 Arsitektur VGG-16

Convolution Network Configuration
16 Weight Layer
Input (224 x 224 RGB image)
Convolution 3-64
Convolution 3-64
Maxpool
Convolution 3-128
Convolution 3-128
Maxpool
Convolution 3-256
Convolution 3-256
Convolution 3-256
Maxpool
Convolution 3-512
Convolution 3-512
Convolution 3-512

Convolution 3-512
Convolution 3-512
Convolution 3-512
Maxpool
FC-4096
FC-4096
FC-1000
Softmax

### 2.3 VGG-19

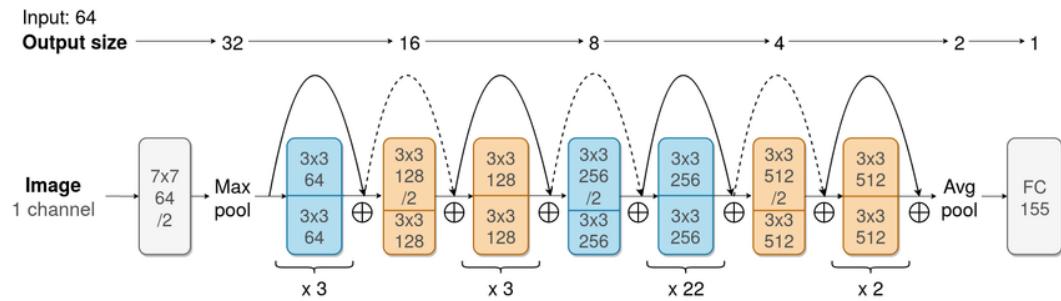
*VGG-19* adalah salah satu model arsitektur yang berasal dari VGGNet. Model ini memiliki 19 layer konvolusi. Dengan *3 layer fully connected*, *5 layer max-pooling* dan *1 layer softmax*. Pada saat training dilakukan, model ini membutuhkan input berupa gambar dengan ukuran pixel sebesar 224 x 224 pixel.

Tabel 2.2 Arsitektur VGG-19

Convolution Network Configuration
19 Weight Layer
Input (224 x 224 RGB image)
Convolution 3-64
Convolution 3-64
Maxpool
Convolution 3-128
Convolution 3-128
Maxpool
Convolution 3-256
Convolution 3-256
Convolution 3-256
Convolution 3-256
Maxpool
Convolution 3-512
Maxpool
FC-4096
FC-4096
FC-1000
Softmax

## 2.4 ResNet-101

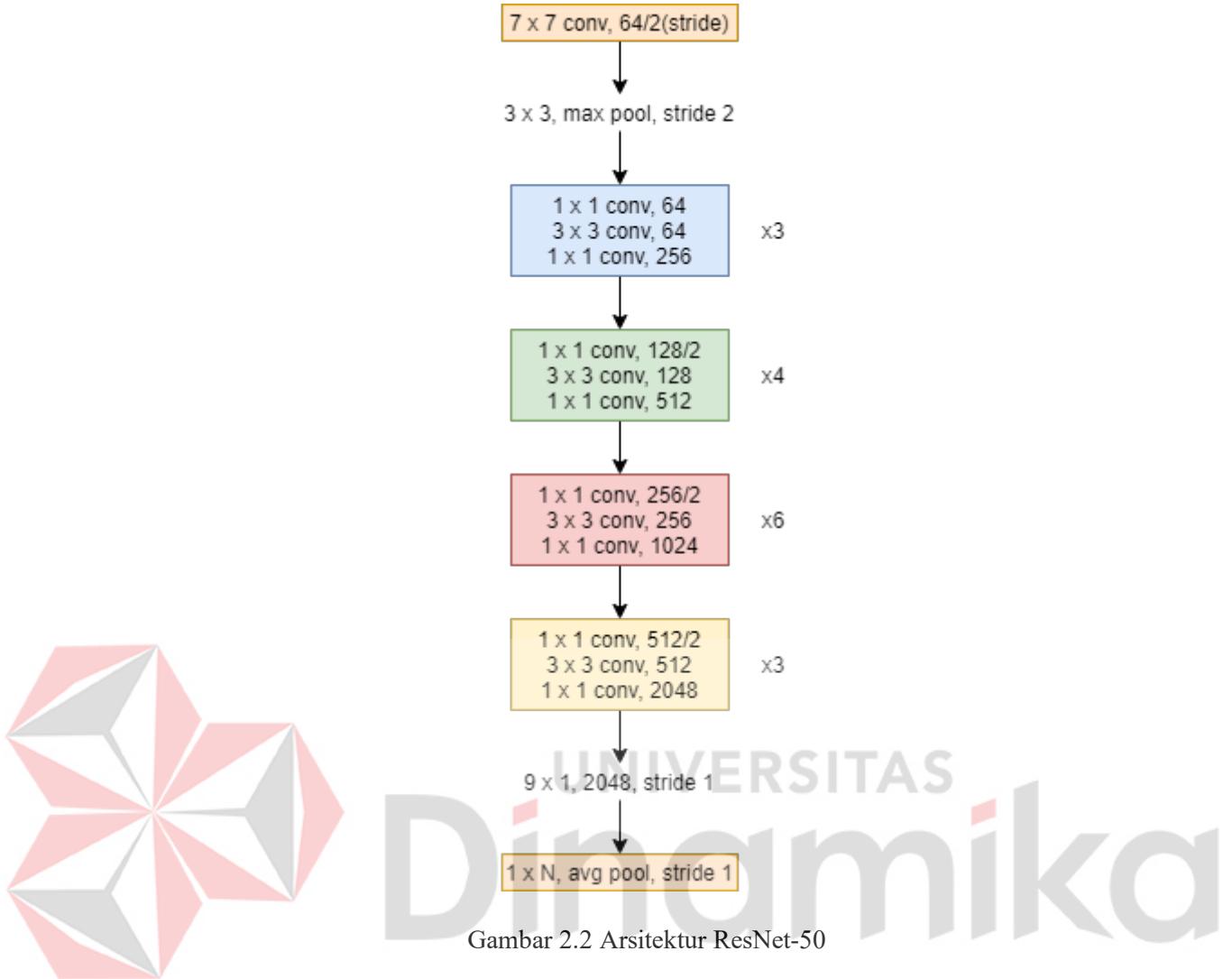
*ResNet-101* merupakan salah satu variasi dari model *Residual Network*. Model arsitektur ini memiliki 101 layer *Convolutional Neural Network*. Model arsitektur ini telah dilatih sebelumnya pada dataset *ImageNet* yang memiliki lebih dari 100.000 data image pada 200 kategori yang berbeda.



Gambar2.1 Arsitektur ResNet-101

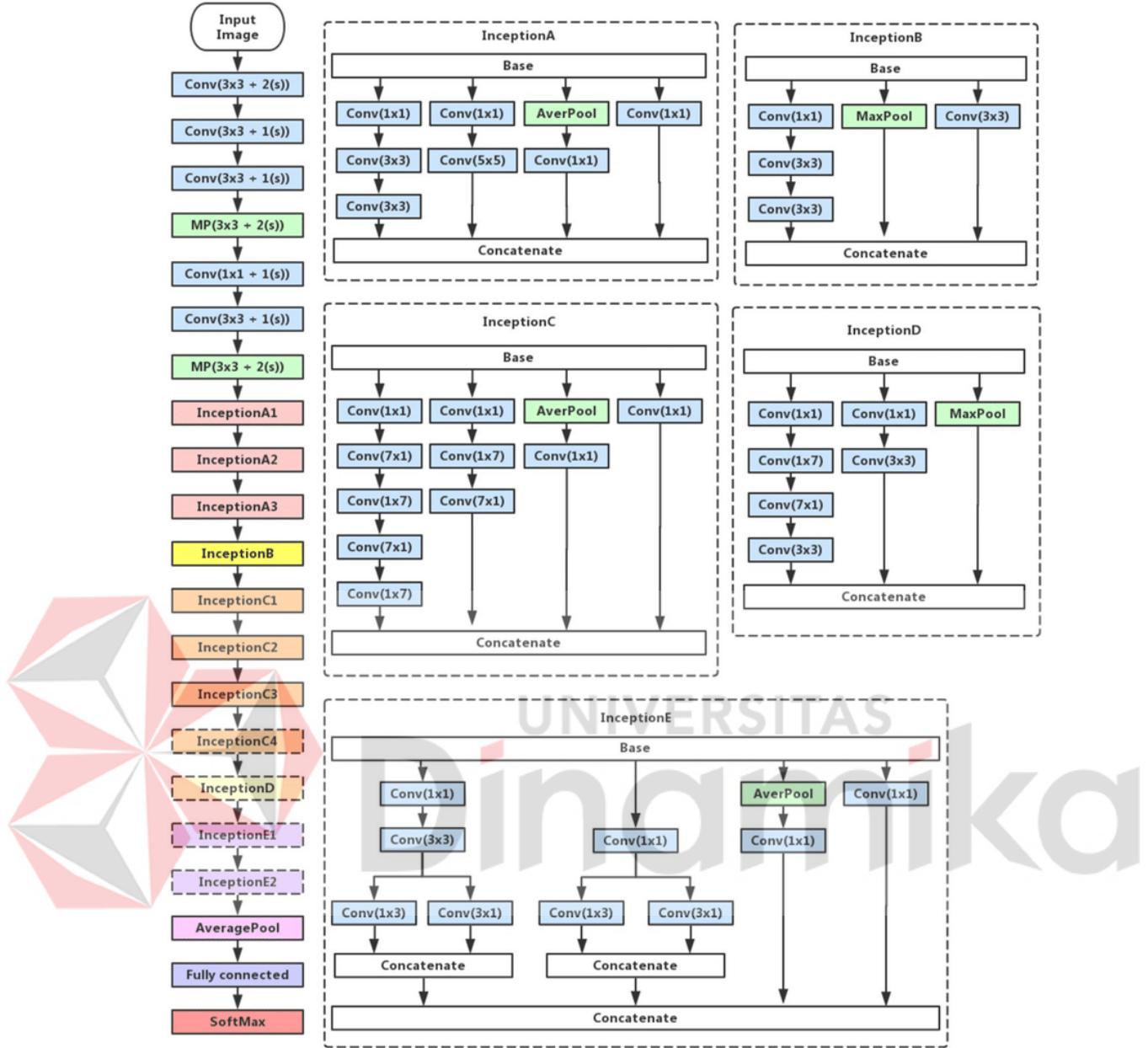
## 2.5 ResNet-50

*ResNet-50* merupakan salah satu variasi dari model *Residual Network* yang memiliki 48 layer konvolusi serta 1 layer *max-pooling* dan juga 1 layer *Average Pool Layer*. Model arsitektur ini dapat menggunakan data input berupa *image* dengan ukuran pixel 224 x 224. Struktur model arsitektur *ResNet-50* dapat dilihat di Gambar 2.2.



## 2.6 Inception V3

*Inception V3* merupakan salah satu varian dari *Inception Family* yang memiliki sebanyak 48 layer. Model arsitektur ini dapat digunakan untuk mengidentifikasi sebuah input berupa gambar dengan ukuran pixel 299 x 299. Struktur model arsitektur *Inception V3* dapat dilihat di Gambar 2.3.



Gambar 2.3 Arsitektur Inception V3

## 2.7 *Diabetic Retinopathy*

*Diabetic Retinopathy* adalah penyakit mata yang dapat dialami oleh pasien penderita diabetes. Penyebabnya adalah komplikasi dari penyakit diabetes yang memicu terjadinya penyumbatan pada pembuluh darah di bagian retina mata. Pada pasien yang menderita penyakit diabetes, memiliki tingkat gula darah yang tinggi

sehingga, berangsur-angsur, pembuluh darah pada bagian retina mata akan tersumbat, sehingga mengakibatkan persediaan darah pada retina berkurang. Hal ini menyebabkan pertumbuhan pembuluh darah baru demi mencukupi keperluan persedian darah, namun, pembuluh darah yang baru berkembang secara tidak sempurna sehingga memiliki kecenderungan untuk pecah atau bocor (Tjin, Willy. (2019)).

Semua penderita diabetes beresiko terserang *Diabetic Retinopathy* dan resiko akan semakin tinggi apabila penderita diabetes mempunyai kondisi lain seperti halnya kolesterol tinggi, sedang dalam masa kehamilan, dan seorang perokok aktif.

Gejala *Diabetic Retinopathy* akan muncul secara perlahan, beberapa gejalanya adalah menurunnya tingkat pandangan mata secara perlahan, munculnya bintik-bintik hitam pada penglihatan, serta munculnya flek atau yang melayang pada penglihatan (floaters), penglihatan berbayang, rasa sakit pada mata atau mata merah, dan lain sebagainya (Monika., & Yuke Lusiani, S. (2019)).

Penyakit *Diabetic Retinopathy* di golongkan menjadi 5 jenis, yaitu: *Non-Diabetic Retinopathy*, *Mild*, *Moderate*, *Severe*, serta *Proliferative Diabetic Retinopathy*. Pada jenis pertama, yaitu *Non Diabetic Retinopathy* tidak terlihat atau terdeteksinya penyakit *Diabetic Retinopathy* pada penderita penyakit diabetes. Jenis kedua, yaitu *Mild*, adalah tingkatan awal terlihatnya beberapa pembengkakan pada pembuluh darah yang berada pada mata penderita penyakit diabetes. Pada jenis ketiga, yaitu *Moderate* adalah tingkatan selanjutnya dari penyakit *Diabetic Retinopathy*, pada tingkat ini, telah terjadi pembengkakan serta pendarahan pada pembuluh darah pada retina mata. Jenis keempat adalah *Severe*. Merupakan

tingkatan dimana terjadinya pembengkakan serta pendarahan pada retina mata yang terjadi dibeberapa titik.

Pada tingkat terakhir yaitu *Proliferative Diabetic Retinopathy* memiliki ciri-ciri yang serupa seperti pada tingkat sebelumnya (*Severe*), yaitu terjadinya pendarahan pada retina mata, serta tumbuhnya pembuluh darah baru yang tidak normal pada retina mata yang disebabkan oleh kurangnya suplai darah.

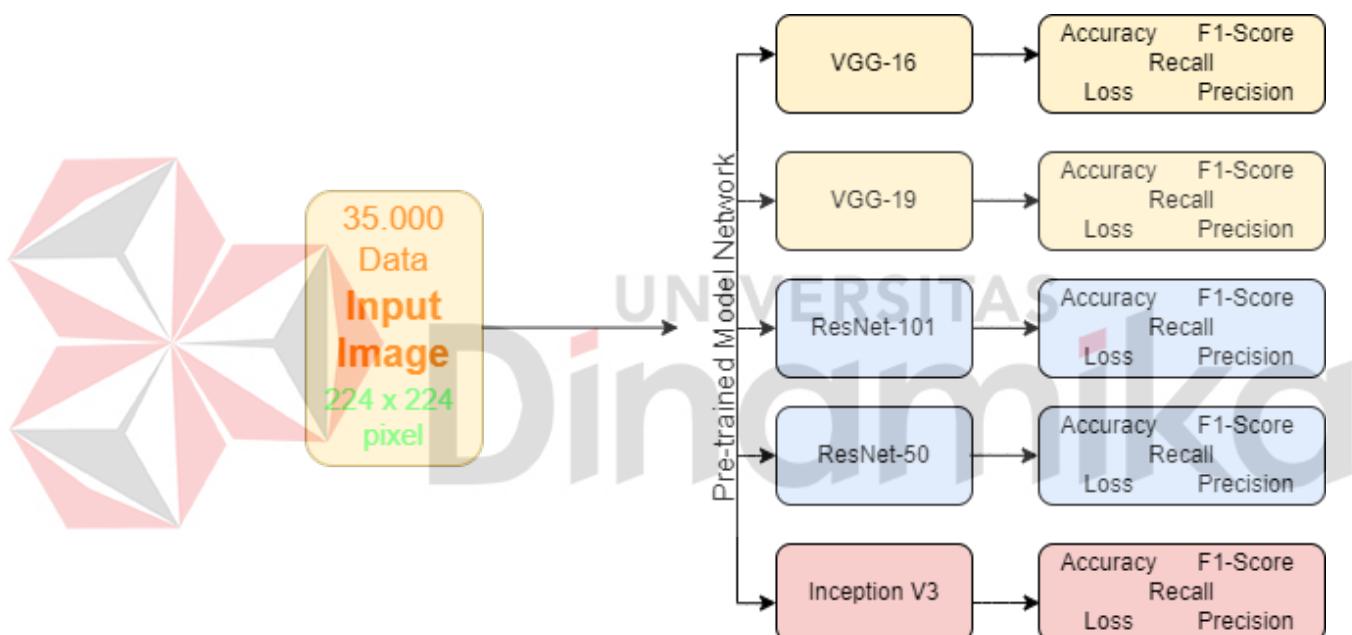


## BAB III

### METODOLOGI PENELITIAN

#### 3.1 Metode Klasifikasi Citra Fundus *Diabetic Retinopathy* Menggunakan *Pre-trained Model Network*

Bagan metode klasifikasi citra fundus *diabetic retinopathy* menggunakan *pre-trained model network* dapat dilihat pada Gambar 3.1 dibawah ini :



Gambar 3.1 Metode Klasifikasi Citra Fundus Diabetic Retinopathy

##### 3.1.1 Dataset

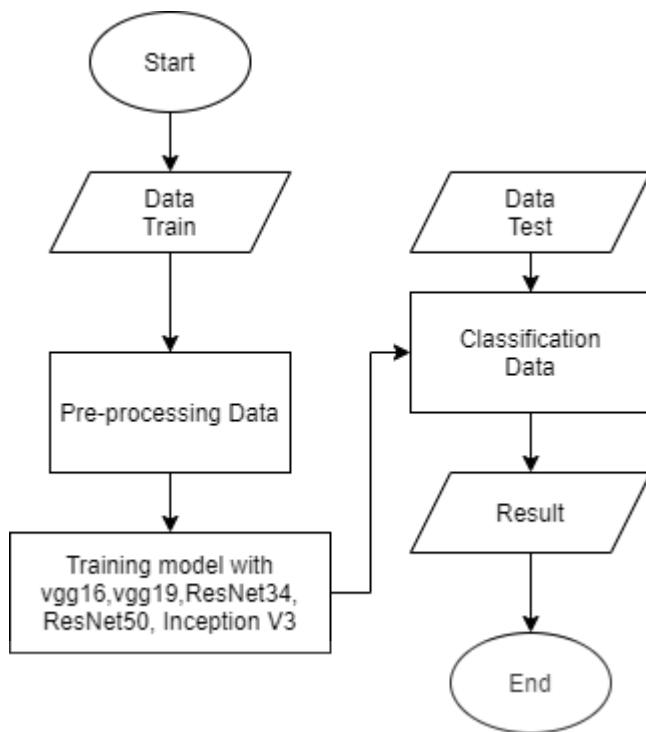
Dataset yang digunakan dalam klasifikasi citra fundus Diabetic Retinopathy ini merupakan dataset yang dapat diakses oleh public melalui website Kaggle.com. Dataset ini memiliki jumlah image fundus sebanyak 35.100 gambar. Dataset ini merupakan dataset yang telah dikurangi ukuran masing-masing file gambar fundus *diabetic retinopathy* dan juga telah dilengkapi dengan file .csv yang dapat digunakan untuk memilah data tingkat

stadium dari fundus *diabetic retinopathy* (*Non Diabetic Retinopathy, Mild, Moderate, Severe, Proliferative Diabetic Retinopathy*). Dataset ini diperoleh dari website bernama Kaggle.com. Dataset ini merupakan dataset yang telah diproses dengan cara mengecilkan ukuran dataset yang awalnya berukuran 89GB menjadi 8GB.

Pada dataset ini terdapat beberapa klas untuk mengkategorikan tingkat stadium dari fundus *diabetic retinopathy*. Klas 0 merupakan tingkat *non diabetic retinopathy* dengan jumlah data pada klas 0 sebanyak 25.810 gambar. Klas 1 adalah tingkat *mild* yang memiliki total jumlah data sebanyak 2443 gambar. Klas 2 adalah *moderate* dengan total jumlah data sebanyak 5292 gambar. Klas 3 adalah *severe* yang memiliki total jumlah data sebanyak 873 gambar. Dan klas 4 adalah *proliferative diabetic retinopathy* yang memiliki jumlah data sebanyak 708 gambar.

### 3.1.2 Arsitektur *Pre-trained Model Network*

Setelah menyiapkan dataset yang diperlukan, konten yang berada pada dataset dimasukkan kedalam beberapa model arsitektur yang telah disiapkan. pada penelitian ini model arsitektur yang akan digunakan untuk perbandingan diantaranya adalah, model arsitektur VGG-16, VGG-19, ResNet-101, ResNet-50 serta Inception V3. Masing-masing Pre-trained model memiliki perbedaan dalam banyaknya layer konvolusi yang terdapat dalam jaringan arsitekturnya. Sesuai dengan namanya, VGG-16 memiliki 16 layer konvolusi, VGG-19 memiliki 19 layer konvolusi, ResNet-34 memiliki 101 layer konvolusi, ResNet-50 memiliki jumlah layer sebanyak 50. Serta Inception V3 memiliki jumlah layer sebanyak 48 layer konvolusi.



Gambar 3.2 Diagram alir proses training

Gambar 3.2 menjelaskan tentang diagram alir proses training pada masing-masing arsitektur. Algoritma dimulai dari mengumpulkan dataset pada suatu lokasi penyimpanan. Setelah itu dataset tersebut akan dilakukan proses *preprocessing* untuk mengatur agar dataset dapat dimasukkan pada masing-masing arsitektur. Proses *preprocessing* ini meliputi pemotongan gambar, mengubah gambar menjadi black-white, dll. Setelah itu dataset tersebut akan masuk pada proses training.

Proses training data yang terdapat pada tiap arsitektur adalah berawal dari memberikan sebuah Data Input (citra) ke lapisan konvolusi. Data citra yang ditraining memiliki sebuah ukuran yaitu  $224 \times 224$  pixel. Gambar diproses melalui tumpukan lapisan konvolusi, dimana filter yang digunakan berukuran kecil, yaitu  $3 \times 3$  pixel. Dalam salah satu konfigurasi, juga

menggunakan filter konvolusi  $1 \times 1$ , yang dapat dilihat sebagai transformasi linier dari saluran input (diikuti oleh non-linier). *Pooling* spasial dijalankan oleh 5 lapisan *max-pooling*, yang mengikuti beberapa lapisan konvolusi (tidak semua lapisan konvolusi. diikuti oleh *max-pooling*). *Max-pooling* dilakukan melalui pixel berukuran  $2 \times 2$ . Lalu tiga lapisan Fully-Connected (FC) mengikuti lapisan convolutional (yang memiliki kedalaman yang berbeda-beda dalam arsitektur yang berbeda pula) (neurohive, 2018).

### 3.1.3 *Accuracy*

Setelah masing-masing *pre-trained model* dijalankan secara utuh, tiap-tiap model arsitektur akan menampilkan nilai akurasinya. Nilai ini akan diolah agar dapat menentukan nilai akurasi rata-rata dari masing-masing *pre-trained model* yang telah dijalankan. Nilai *accuracy* dapat ditemukan dengan menggunakan rumus berikut ini:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \quad (1)$$

### 3.1.4 *Loss*

*Loss* dapat diperoleh dengan menggunakan fungsi *evaluate*. Fungsi ini dapat membandingkan besarnya akurasi serta besarnya error/loss yang dapat terjadi pada sebuah arsitektur *pre-trained model*. Dengan mengevaluasi kembali hasil training yang sudah dilakukan, diharapkan dapat meningkatkan akurasi pada penelitian-penelitian selanjutnya. Nilai *Loss* didapatkan menggunakan rumus berikut ini:

$$L = \frac{1}{n} \sum_{t=1}^n e_t^2 \quad (2)$$

### 3.1.5 *Precision*

*Precision* merupakan rasio jumlah *positive class* yang telah diklasifikasikan dengan benar dan dibagi oleh jumlah dari TP (*True Positive*) dan FP (*False Positive*). Hasil pembagian tersebut dapat memberikan nilai *precision*. Nilai *precision* dapat didapatkan dengan menggunakan rumus berikut:



$$\text{Precision} = \frac{TP}{TP+FP} \quad (3)$$

Semakin tinggi nilai dari *precision* semakin bagus pula hasil klasifikasi tersebut.

### 3.1.6 *Recall*

*Recall* merupakan rasio prediksi *True Positive* (TP), dibandingkan dengan penjumlahan antara TP (*True Positive*) dan FN (*False Negative*). Hasil dari perbandingan tersebut dapat diperoleh nilai dari *recall*. Nilai dari *recall* harus tinggi, idealnya adalah bernilai 1. Nilai *recall* didapatkan dengan menggunakan rumus sebagai berikut:

$$\text{recall} = \frac{TP}{TP+FN} \quad (4)$$

### 3.1.7 ***F1-Score***

*F1-Score* adalah sebuah nilai hasil dari kombinasi antara nilai *precision* dan juga nilai dari *recall*. Nilai *F1-Score* dapat didapatkan dengan menggunakan rumus sebagai berikut :

$$F1score = 2 * \frac{(Recall * Precision)}{(Recall + Precision)} \quad (5)$$

*F1-Score* sebaiknya menghasilkan nilai yang tinggi. Idealnya adalah bernilai 1.



## BAB IV

### HASIL DAN PEMBAHASAN

#### 4.1 Pengujian Komparasi 5 *Pre-trained Model Network*

##### 4.1.1 Tujuan Pengujian Komparasi 5 *Pre-trained Model Network*

Tujuan pengujian ini adalah untuk mengkomparasi hasil akurasi dari tiap-tiap *pre-trained model network*, serta menentukan model *pre-trained* mana yang memiliki tingkat akurasi paling tinggi dengan tingkat komputasi yang rendah.

##### 4.1.2 Prosedur Pengujian Komparasi 5 *Pre-trained Model Network*

1. Memberikan *input data* yang sama pada tiap-tiap *pre-trained model network* yang digunakan.
2. Menjalankan proses *training* pada tiap-tiap *pre-trained model network*.
3. Mengambil data akurasi serta waktu komputasi pada *epochs* ke-10,15,20.

##### 4.1.3 Hasil Pengujian Komparasi 5 *Pre-trained Model Network*

Tabel 4.1 Hasil Pengujian Komparasi 5 Pre-trained model network

Epochs.	Akurasi Klasifikasi Model (%)				
	VGG-16	VGG-19	ResNet-101	ResNet-50	Inception V3
10.	73.76	74.60	73.49	73.76	73.44
15.	74.08	75.32	73.67	74.63	77.99
20.	74.02	75.61	73.67	73.43	73.40

#### 4.2 Hasil Perolehan Nilai 5 Parameter Uji

##### 4.2.1 Tujuan Memperoleh Nilai Hasil 5 Parameter Uji

Tujuan untuk memperoleh nilai hasil 5 parameter uji adalah untuk mengevaluasi hasil training pada sebuah *pre-trained model network*.

##### 4.2.2 Prosedur Memperoleh Nilai Hasil 5 Parameter Uji

1. Menjalankan program *training* pada tiap-tiap model arsitektur.

2. Menjalankan program *confusion matrix* untuk mendapatkan nilai parameter uji.
3. Menampilkan nilai parameter uji.

#### 4.2.3 Hasil Nilai 5 Parameter Uji

Terdapat 5 parameter uji yang digunakan, yaitu; *Loss*, *Accuracy*, *Precision*, *Recall*, dan *F1-score*. Setiap pre-trained model network yang digunakan akan dijalankan fungsi yang akan memberikan ke-5 parameter diatas.

##### A. VGG-16

Tabel 4.2 Hasil Nilai Parameter VGG-16

VGG-16					
Epoch	Loss	Accurac y (%)	Precision	F1-Score	Recall
5	0.7805	74.49%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.16 Klas 3 : 0.00 Klas 4 : 0.03	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.01 Klas 3 : 0.00 Klas 4 : 0.02	Klas 0 : 0.98 Klas 1 : 0.00 Klas 2 : 0.01 Klas 3 : 0.00 Klas 4 : 0.01
10	0.7448	74.49%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.12 Klas 3 : 0.00 Klas 4 : 0.04	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.02 Klas 3 : 0.00 Klas 4 : 0.03	Klas 0 : 0.98 Klas 1 : 0.00 Klas 2 : 0.01 Klas 3 : 0.00 Klas 4 : 0.02
15	0.7919	73.83%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.15 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.02 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.98 Klas 1 : 0.00 Klas 2 : 0.01 Klas 3 : 0.00 Klas 4 : 0.00
20	0.7844	74.22%	Klas 0 : 0.74	Klas 0 : 0.84	Klas 0 : 0.98

			Klas 1 : 0.00	Klas 1 : 0.00	Klas 1 : 0.00
			Klas 2 : 0.21	Klas 2 : 0.02	Klas 2 : 0.01
			Klas 3 : 0.00	Klas 3 : 0.00	Klas 3 : 0.00
			Klas 4 : 0.00	Klas 4 : 0.00	Klas 4 : 0.00

Berdasarkan tabel diatas, terlihat bahwa nilai *accuracy* pada *epochs* 5, 10, 15, maupun 20 memiliki hasil yang serupa, yaitu berada pada *range* 73%-74%. Sedangkan nilai *loss* berkisar antara *range* 0.74 hingga 0.79. Pada parameter *precision*, *recall*, dan *F1-score* terdapat hasil uji yang memiliki nilai 0.00, hal ini terjadi karena dataset yang digunakan merupakan *imbalanced* dataset. Sedangkan model arsitektur pada *deep learning* dibangun dengan asumsi bahwa dataset yang digunakan merupakan *balanced* dataset. Oleh karena itu, model akan lebih banyak memprediksi Klas yang memiliki jumlah data terbanyak, dalam penilitian ini klas yang memiliki data terbanyak adalah klas 0 (*Non-Diabetic Retinopathy*).

### B. VGG-19

Tabel 4.3 Hasil Nilai Parameter VGG-19

VGG-19					
<i>Epoch</i>	<i>Loss</i>	<i>Accurac y (%)</i>	<i>Precision</i>	<i>F1-Score</i>	<i>Recall</i>
5	0.7699	74.00%	Klas 0 : 0.73	Klas 0 : 0.82	Klas 0 : 0.94
			Klas 1 : 0.00	Klas 1 : 0.00	Klas 1 : 0.00
			Klas 2 : 0.20	Klas 2 : 0.07	Klas 2 : 0.04
			Klas 3 : 0.05	Klas 3 : 0.04	Klas 3 : 0.03
			Klas 4 : 0.04	Klas 4 : 0.03	Klas 4 : 0.02
10	0.7589	74.13%	Klas 0 : 0.73	Klas 0 : 0.84	Klas 0 : 0.98
			Klas 1 : 0.00	Klas 1 : 0.00	Klas 1 : 0.00
			Klas 2 : 0.22	Klas 2 : 0.03	Klas 2 : 0.02
			Klas 3 : 0.00	Klas 3 : 0.00	Klas 3 : 0.00

			Klas 4 : 0.02	Klas 4 : 0.01	Klas 4 : 0.01
15	0.7568	74.08%	Klas 0 : 0.73	Klas 0 : 0.80	Klas 0 : 0.90
			Klas 1 : 0.00	Klas 1 : 0.00	Klas 1 : 0.00
			Klas 2 : 0.16	Klas 2 : 0.12	Klas 2 : 0.10
			Klas 3 : 0.03	Klas 3 : 0.01	Klas 3 : 0.01
			Klas 4 : 0.04	Klas 4 : 0.03	Klas 4 : 0.02
20	0.7603	74.86%	Klas 0 : 0.72	Klas 0 : 0.82	Klas 0 : 0.95
			Klas 1 : 0.00	Klas 1 : 0.00	Klas 1 : 0.00
			Klas 2 : 0.16	Klas 2 : 0.06	Klas 2 : 0.04
			Klas 3 : 0.05	Klas 3 : 0.02	Klas 3 : 0.01
			Klas 4 : 0.05	Klas 4 : 0.03	Klas 4 : 0.03

Berdasarkan tabel diatas, terlihat bahwa nilai *accuracy* pada *epochs* 5, 10, 15, maupun 20 memiliki hasil yang serupa, yaitu berada pada *range* 74%-74.86%. Sedangkan nilai *loss* berkisar antara *range* 0.75 hingga 0.76. Pada *parameter precision, recall*, dan *F1-score* terdapat hasil uji yang memiliki nilai 0.00, hal ini terjadi karena dataset yang digunakan merupakan *imbalanced* dataset. Sedangkan model arsitektur pada *deep learning* dibangun dengan asumsi bahwa dataset yang digunakan merupakan *balanced* dataset. Oleh karena itu, model akan lebih banyak memprediksi Klas yang memiliki jumlah data terbanyak, dalam penilitian ini klas yang memiliki data terbanyak adalah klas 0 (*Non-Diabetic Retinopathy*).

### C. ResNet-101

Tabel 4.4 Hasil Nilai Parameter ResNet 101

ResNet-101					
Epoch	Loss	Accuracy (%)	Precision	F1-Score	Recall
5	0.8757	73.72%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.85 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00
10	0.8692	73.48%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.85 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00
15	1.0553	74.15%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.14 Klas 3 : 0.05 Klas 4 : 0.00	Klas 0 : 0.77 Klas 1 : 0.00 Klas 2 : 0.14 Klas 3 : 0.05 Klas 4 : 0.00	Klas 0 : 0.81 Klas 1 : 0.00 Klas 2 : 0.14 Klas 3 : 0.06 Klas 4 : 0.00
20	0.8818	73.67%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00

Hasil yang di dapat dari pengujian pada arsitektur ResNet-101, menghasilkan nilai akurasi dengan rentang nilai dari 72% hingga 74%. Sedangkan hasil yang didapatkan pada nilai *loss* adalah pada rentang nilai 0,86 hingga 1,05. Terdapat nilai 0,00 yang tampil pada *class* 1, 2, 3, 4. Sedangkan pada *class* 0, selalu didapatkan nilai yang mendekati dengan 1,0.

Hal ini dikarenakan oleh jumlah data yang terdapat pada *class 1, 2, 3, 4* jauh lebih sedikit daripada data yang terdapat pada *class 0*.

#### D. ResNet-50

Tabel 4.5 Hasil Nilai Parameter ResNet-50

ResNet-50					
<i>Epoch</i>	<i>Loss</i>	<i>Accurac y (%)</i>	<i>Precision</i>	<i>F1-Score</i>	<i>Recall</i>
5	0.8754	73.03%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00
10	0.7806	72.75%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00
15	0.7859	73.86%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.14 Klas 3 : 0.01 Klas 4 : 0.00	Klas 0 : 0.82 Klas 1 : 0.00 Klas 2 : 0.07 Klas 3 : 0.01 Klas 4 : 0.00	Klas 0 : 0.93 Klas 1 : 0.00 Klas 2 : 0.05 Klas 3 : 0.01 Klas 4 : 0.00
20	0.8329	73.84%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.85 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00

Hasil yang di dapat dari pengujian pada arsitektur ResNet-50, menghasilkan nilai akurasi dengan rentang nilai dari 72% hingga 74%. Sedangkan hasil yang didapatkan pada nilai loss adalah pada rentang nilai 0,86 hingga 1,05. Terdapat nilai 0,00 yang tampil pada *class* 1, 2, 3, 4. Sedangkan pada *class* 0, selalu didapati nilai yang mendekati dengan 1,0. Hal ini disebabkan oleh jumlah data yang terdapat di *class* 0 lebih besar dibandingkan dengan jumlah data yang terdapat pada *class* lainnya, oleh karena itu, model akan cenderung memprediksi *class* dengan jumlah data terbanyak, dalam hal ini adalah *class* 0.

### E. Inception V3

Tabel 4.6 Hasil Nilai Parameter Inception V3

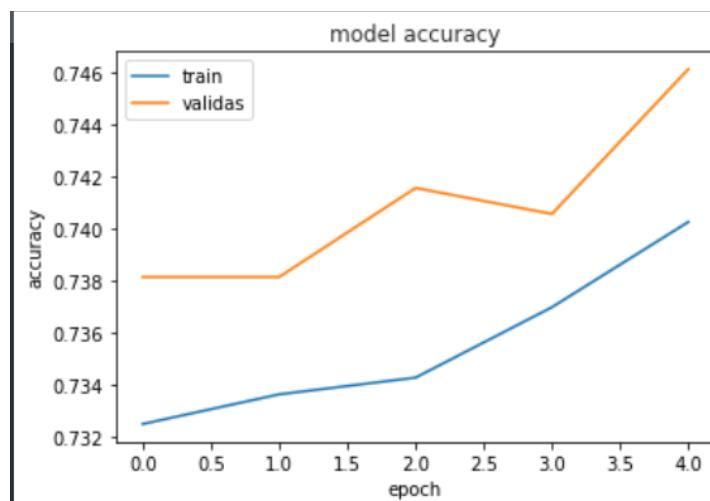
<i>Inception V3</i>					
<i>Epoch</i>	<i>Loss</i>	<i>Accurac y (%)</i>	<i>Precision</i>	<i>F1-Score</i>	<i>Recall</i>
5	0.7756	73.11%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.84 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.99 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00
10	0.8755	73.44%	Klas 0 : 0.74 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 0.85 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00	Klas 0 : 1.00 Klas 1 : 0.00 Klas 2 : 0.00 Klas 3 : 0.00 Klas 4 : 0.00
15	0.6567	78.39%	Klas 0 : 0.73 Klas 1 : 0.00 Klas 2 : 0.15 Klas 3 : 0.03 Klas 4 : 0.00	Klas 0 : 0.81 Klas 1 : 0.00 Klas 2 : 0.10 Klas 3 : 0.01 Klas 4 : 0.00	Klas 0 : 0.91 Klas 1 : 0.00 Klas 2 : 0.08 Klas 3 : 0.01 Klas 4 : 0.00

20	0.8712	73.83%	Klas 0 : 0.74	Klas 0 : 0.85	Klas 0 : 1.00
			Klas 1 : 0.00	Klas 1 : 0.00	Klas 1 : 0.00
			Klas 2 : 0.00	Klas 2 : 0.00	Klas 2 : 0.00
			Klas 3 : 0.00	Klas 3 : 0.00	Klas 3 : 0.00
			Klas 4 : 0.00	Klas 4 : 0.00	Klas 4 : 0.00

Pada arsitektur *Inception V3*, didapatkan nilai akurasi tertinggi dari kelima arsitektur yang digunakan, yaitu pada epochs ke-15 dengan hasil akurasi mencapai 78.39%. Serta didapati nilai *precision*, *recall*, dan *F1-score* dengan nilai tertinggi yang mendekati 1.00 ada pada *class 0*. Model cenderung memprediksi *class 0* karena jumlah data yang berada pada *class* tersebut lebih banyak dibandingkan dengan keempat *class* lainnya. Sehingga nilai *precision*, *recall*, dan *F1-score* mendekati nilai 1,00 pada *class 0*.

### 4.3 Hasil evaluasi terhadap nilai dari akurasi dan loss

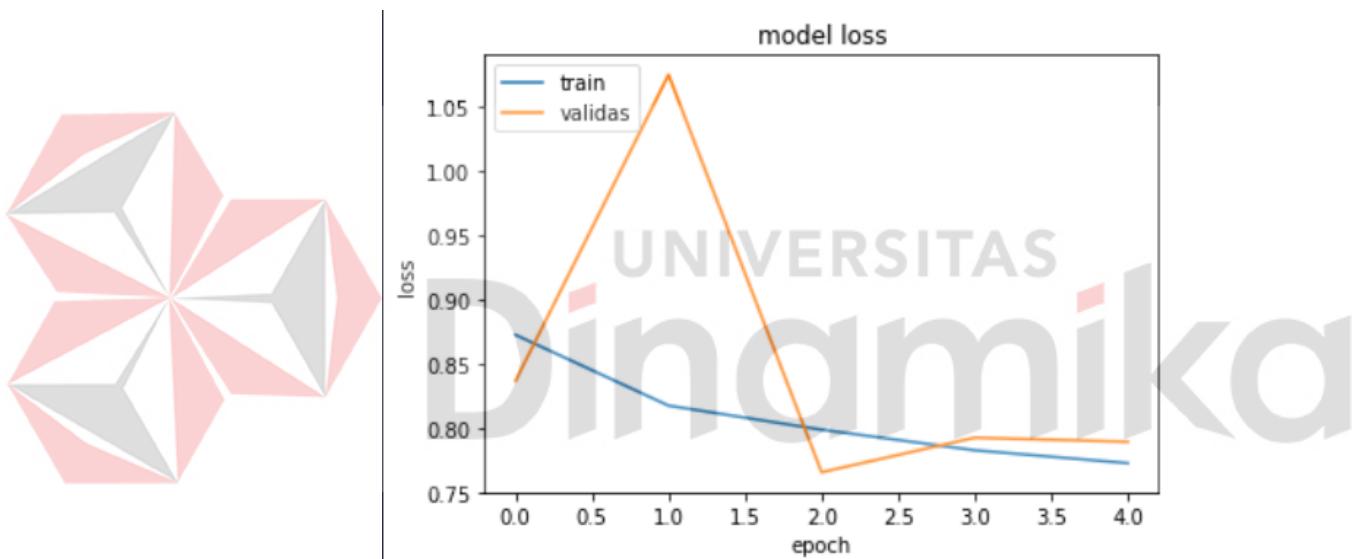
Berdasarkan dari hasil evaluasi terhadap nilai akurasi dan juga nilai dari *loss*, didapatkan sebuah grafik yang dapat memberikan informasi sebagai berikut:



Gambar 4.1 Grafik model accuracy

Dari Gambar 4.1 diatas dapat disimpulkan bahwa pada model yang dijalankan, terlihat bahwa nilai *accuracy* mengalami peningkatan sejajar dengan bertambahnya jumlah *epochs* yang diberikan pada model yang

sedang di *training*. Model yang digunakan pada Gambar 4.1 merupakan *pre-trained model VGG-19*. Semakin banyak *epoch* yang digunakan, maka semakin baik pula *accuracy* yang didapatkan, namun tidak selalu sebuah model dengan *accuracy* tinggi merupakan hasil yang terbaik, karena apabila sebuah model memiliki *accuracy* tinggi dengan hasil *loss* yang tinggi pula, maka dapat disebut *overfitting*. *Overfitting* adalah keadaan dimana pada data *training* mendapatkan akurasi yang tinggi, namun memiliki akurasi yang rendah pada prediksinya. Adapula *underfitting*, merupakan keadaan dimana model pelatihan tidak memiliki data pelatihan yang mencukupi, atau dengan kata lain kekurangan *training data*.



Gambar 4.2 Grafik model Loss

Dari grafik diatas dapat disimpulkan bahwa pada model yang telah di *training* mengalami lonjakan pada bagian validasi pada saat *epochs* rendah. Validasi *Loss*, atau *val loss* terdeteksi stabil pada *epochs* tinggi. Nilai *loss* yang tinggi dapat dikarenakan oleh beberapa faktor, salah satunya adalah ketidakmampuan sebuah model untuk menganalisa sebuah dataset yang diberikan, hal ini disebabkan oleh kurang baiknya kualitas dataset yang digunakan.

#### 4.4 Ranking Pre-Trained Model Network

Berikut ini adalah hasil dari pengurutan nilai akurasi dalam persen dan juga waktu komputasi dalam detik pada setiap *pre-trained model network* yang digunakan, yaitu: *VGG-16*, *VGG-19*, *ResNet-101*, *ResNet-50*, dan *Inception V3*.

Tabel 4.7 Ranking hasil komparasi nilai akurasi

Ranking	Pre-trained model	Akurasi (%)
1	<i>Inception V3</i>	77.99
2	<i>VGG-19</i>	75.32
3	<i>ResNet-50</i>	74.63
4	<i>VGG-16</i>	74.08
5	<i>ResNet-101</i>	73.67

Dari Tabel 4.7, dapat disimpulkan bahwa nilai akurasi tertinggi diperoleh dengan menggunakan *pre-trained model Inception V3* dengan hasil akurasi mencapai 77.99%. Dan hasil nilai akurasi terendah diperoleh pada *pre-trained model ResNet-101* dengan akurasi hingga 73.67%.

Tabel 4.8 Ranking hasil komparasi waktu komputasi

Ranking	Pre-trained model	Total komputasi
1	<i>VGG-16</i>	13.291 detik
2	<i>VGG-19</i>	13.300 detik
3	<i>ResNet-50</i>	13.884 detik
4	<i>ResNet-101</i>	14.987 detik
5	<i>Inception V3</i>	22.870 detik

Dari Tabel 4.8 , dapat disimpulkan bahwa waktu komputasi tercepat dapat dicapai dengan menggunakan *pre-trained model VGG-16*. Sedangkan waktu komputasi terlama dihasilkan pada *pre-trained model Inception V3*.

## BAB V

### PENUTUPAN

#### 5.1 Kesimpulan

Berdasarkan hasil analisa dan pembahasan pada bab 4, dapat disimpulkan sebagai berikut:

1. Dari hasil percobaan yang telah dilakukan, dapat diambil kesimpulan bahwasannya kelima *pre-trained model network* yang digunakan dapat berjalan dengan baik dan sesuai harapan.
2. Hasil training dari arsitektur *Inception V3* menghasilkan nilai akurasi paling tinggi dibandingkan dengan keempat arsitektur lainnya yang memiliki hasil akurasi hingga 77,99% pada *epoch* ke-15, dengan total waktu komputasi selama 15.594 detik.
3. Penggunaan arsitektur *Inception V3* membutuhkan waktu komputasi yang paling lama, hingga mencapai 1 detik pada tiap *steps* dan 1.030 hingga 1.166 detik pada setiap *epoch* yang dijalankan.
4. Penggunaan dataset yang memiliki ketidakseimbangan (*unbalanced*) data, menghasilkan nilai *precision*, *recall*, serta *F1-score* yang tidak bisa maksimal pada beberapa *pre-trained model network*.
5. Pada *class* 0, yaitu pada data dengan label 0 (*NO-Diabetic Retinopathy*) selalu mendapatkan nilai *precision*, *recall*, maupun *F1-score* yang tinggi, bahkan dapat mencapai nilai 1.00. Hal ini disebabkan oleh data yang terdapat dalam kategori 0, memiliki data terbanyak dalam dataset yang digunakan.
6. Hasil komparasi nilai akurasi dari 5 *pre-trained model network* menunjukkan bahwa nilai tertinggi diperoleh dengan menggunakan *pre-trained model Inception V3* dengan hasil akurasi mencapai 77.99%. Dan hasil nilai akurasi terendah diperoleh pada *pre-trained model ResNet-101* dengan akurasi hingga 73.67%

7. Hasil komparasi waktu komputasi dari 5 *pre-trained model network* menunjukkan bahwa waktu komputasi tercepat dapat dicapai dengan menggunakan *pre-trained model VGG-16*. Sedangkan waktu komputasi terlama dihasilkan pada *pre-trained model Inception V3*.

## 5.2 Saran

Pada penelitian selanjutnya diharapkan dapat menyempurnakan kekurangan dari Tugas Akhir ini dan mengimplementasikan beberapa petunjuk yang diberikan penulis

1. Mencoba untuk menjalankan atau mengimplementasikan *pre-processing image* yang lebih kompleks, misalkan: menggunakan *heatmapping* dan *saliency*.
2. Pemilihan dataset yang memiliki keseimbangan jumlah data yang sama antara satu kelas dengan kelas yang lain (*balanced data*).
3. Penggunaan *loss function* yang berbeda, disarankan untuk mencoba *categorical cross entropy*.



## DAFTAR PUSTAKA

- Arsal, M., Agus Wardijono, B., & Anggraini, D. (2020). Face Recognition Untuk Akses Pegawai Bank Menggunakan Deep Learning Dengan Metode CNN. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 6(1), 55–63. <https://doi.org/10.25077/teknosi.v6i1.2020.55-63>
- Monika., & Yuke Lusiani, S. (2019). *Retinopati Diabetik - Penyebab, Gejala, dan Penanganannya*. KMN Eye Care. <https://www.klinikmatanusantara.com/id/ketahui-lebih-lanjut/info-kesehatan-mata-dari-kmn-eyecare/artikel/retinopati-diabetik-penyebab-gejala-dan-penanganannya/>
- Muneeb, ul Hasan. (2018). *VGG16 - Convolutional Network for Classification and Detection*. Neurohive. <https://neurohive.io/en/popular-networks/vgg16/>
- Simonyan, K., & Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, 1–14.
- Tjin, Willy. (2019). *Retinopati Diabetik - Gejala, Penyebab Dan Mengobati - Alodokter*. Alodokter. <https://www.alodokter.com/retinopati-diabetik>
- Yani, D. U. R. (2017). *KLASIFIKASI TINGKAT KEPARAHAN NON - PROLIFERATIVE DIABETIC RETINOPATHY*.