

**IMPLEMENTASI ARSITEKTUR *INCEPTION RESNET-V2*  
DALAM DIAGNOSIS PENYAKIT PARKINSON  
BERBASIS ANDROID**

**SKRIPSI**



**AL WAASIU**

**H071181007**

**PROGRAM STUDI SISTEM INFORMASI**

**DEPARTEMEN MATEMATIKA**

**FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM**

**UNIVERSITAS HASANUDDIN**

**MAKASSAR**

**2023**

**IMPLEMENTASI ARSITEKTUR *INCEPTION RESNET-V2*  
DALAM DIAGNOSIS PENYAKIT PARKINSON  
BERBASIS ANDROID**

**SKRIPSI**

**Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Sains  
pada Program Studi Sistem Informasi Departemen Matematika Fakultas  
Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin**

**UNIVERSITAS HASANUDDIN**

**AL WAASIU**

**H071181007**

**PROGRAM STUDI SISTEM INFORMASI  
DEPARTEMEN MATEMATIKA  
FAKULTAS MATEMATIKA DAN ILMU PENGETAHUAN ALAM  
UNIVERSITAS HASANUDDIN  
MAKASSAR**

**2023**

## PERNYATAAN KEASLIAN

Saya yang bertanda tangan di bawah ini :

Nama : Al Waasiu  
Nim : H071181007  
Program Studi : Sistem Informasi  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulis saya dengan judul :

**Implementasi arsitektur *Inception ResNet-V2* dalam diagnosis penyakit parkinson berbasis android**

adalah benar hasil karya sendiri, bukan hasil plagiat dan belum pernah dipublikasikan dalam bentuk apapun.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Makassar, 4 Juni 2023

  
10000  
METERAI  
TEMPEL  
F6030AKX478727031  
Al Waasiu  
H071181007

**IMPLEMENTASI ARSITEKTUR *INCEPTION RESNET-V2*  
DALAM DIAGNOSIS PENYAKIT PARKINSON  
BERBASIS ANDROID**

**Disusun dan diajukan oleh :**

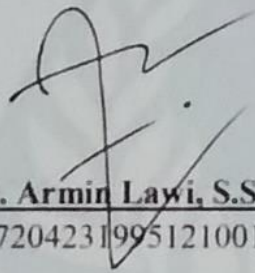
**Al Waasiu**

**H071181007**

Telah berhasil dipertahankan di hadapan Dewan Penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Komputer pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin

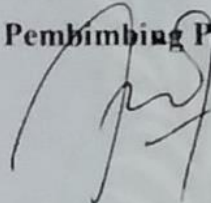
**Menyetujui ,**

**Pembimbing Utama,**



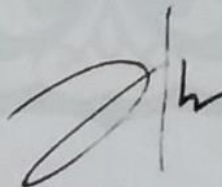
**Dr. Eng. Armin Lawi, S.Si., M.Eng.**  
NIP. 197204231995121001

**Pembimbing Pertama,**



**A. Muh. Amil Siddik, S.Si., M.Si**  
NIP. 199110032019031015

**Ketua Program Studi,**



**Dr. Hendra, S.Si., M.Kom.**  
NIP. 197601022002121001



## HALAMAN PENGESAHAN

Skripsi ini diajukan oleh:

Nama : Al Waastu

NIM : H071181007

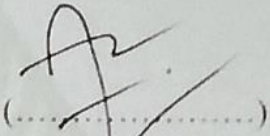
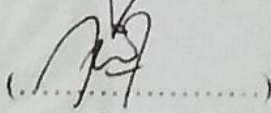
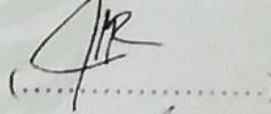
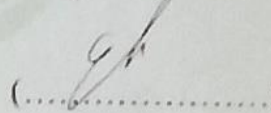
Program Studi : Sistem Informasi

Judul Skripsi : Implementasi Arsitektur *Inception ResNet-V2* Dalam Diagnosis Penyakit Parkinson Berbasis Android

Telah berhasil mempertahankan di hadapan dewan penguji dan diterima sebagai bagian persyaratan yang diperlukan untuk memperoleh gelar Sarjana Sains pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

### DEWAN PENGUJI

#### Tanda Tangan

1. Ketua : Dr.Eng. Armin Lawi, S.Si., M.Eng. 
2. Sekretaris : A. Muh. Amil Siddik, S.Si., M.Si 
3. Anggota : Dr. Andi Muhammad Anwar, S.Si., M.Si 
4. Anggota : Ir. Eliyah Acantha Manapa Sampetoding, S.Kom., M.Kom. 

Ditetapkan di : Makassar

Tanggal : 14 Juni 2023



## KATA PENGANTAR

Segala puji dan syukur kehadirat Allah SWT atas berkah, rahamat dan hidayah-Nya yang senantiasa dilimpahkan kepada penulis, sehingga bisa menyelesaikan skripsi dengan judul “Implementasi Arsitektur *Inception ResNet-V2* Dalam Diagnosis Penyakit Parkinson Berbasis Android” sebagai syarat untuk menyelesaikan gelar Sarjana Sains pada Program Studi Sistem Informasi Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.

Dalam penyusunan skripsi ini banyak hambatan serta rintangan yang penulis hadapi namun pada akhirnya dapat melaluinya berkat adanya bimbingan dan bantuan dari berbagai pihak baik secara moral maupun spiritual. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih yang tak terhingga kepada orang tua penulis, Ibunda **Wa Irani** dan Ayahanda **La Ode Rasale**, sebagai tempat kembali setelah pergi, dan tempat terlelap dikala lelah, terima kasih atas kasih sayang, doa, dan nasihat yang tulus sebagai bekal kehidupan. Rasa terima kasih juga penulis tujukan kepada **Malikul Mulki**, **Al Mughni** dan **Al Waduudu** sebagai adik, motivator, dan rival dalam membanggakan kedua orang tua, terima kasih atas dukungan yang penulis dapatkan selama ini.

Penghargaan dan ucapan terima kasih dengan penuh ketulusan juga penulis ucapkan kepada:

1. Bapak Rektor **Prof. Dr. Ir. Jamaluddin Jompa, M.Sc** selaku Rektor Universitas Hasanuddin beserta jajarannya.
2. Bapak **Dr. Eng. Amiruddin, M.Si** selaku Dekan Fakultas Matematika dan Ilmu Pengetahuan Alam Universitas Hasanuddin.
3. Bapak **Dr. Nurdin, S.Si., M.Si** selaku ketua departemen matematika fakultas matematika dan ilmu pengetahuan alam universitas hasanuddin, dosen-dosen pengajar, dan staf Departemen Matematika atas ilmu dan bantuan yang selama ini telah diberikan.
4. Bapak **Dr. Hendra, S.Si, M.Kom.** selaku ketua program studi sistem informasi universitas hasanuddin atas semua ilmu yang telah diberikan selama menjadi mahasiswa Sistem Informasi Unhas.

5. Bapak **Dr. Eng. Armin Lawi, S.Si., M.Eng.** selaku dosen pembimbing utama sekaligus ketua tim penguji, atas semua ilmu yang telah diberikan selama proses perkuliahan dan bimbingan, serta senantiasa memotivasi penulis dalam penulisan skripsi ini.
6. Bapak **A. Muh. Amil Siddik, S.Si., M.Si** selaku dosen pembimbing pertama sekaligus sekretaris tim penguji atas ilmu yang diberikan selama proses perkuliahan dan bimbingan, serta segala bentuk bantuan yang telah diberikan dalam penyusunan skripsi ini.
7. Bapak **Andi Muhammad Anwar, S.Si., M.Si** selaku anggota tim penguji atas segala ilmu yang telah diberikan selama proses perkuliahan, serta telah bersedia meluangkan waktunya untuk memberikan masukan dan kritik yang membangun dalam proses penyusunan skripsi ini.
8. Bapak **Ir. Eliyah Acantha Manapa Sampetoding, S.Kom., M.Kom.** sebagai anggota tim penguji atas segala kritikan dan masukan yang membangun dalam penyusunan skripsi ini.
9. Saudariku **Hera Ariska, Rahmayati, Nurul Fitri Hasari Ramadhani, Nur Aviva Iriawan**, teman-teman **Sistem Informasi Uh 2018**, Sahabat Ramsis (**Jumariah, St. Khadijah, Fitriana, Mega karunia Sari**), serta teman yang tidak dapat disebutkan satu persatu, yang telah sama-sama berjuang, senantiasa memberikan banyak dukungan serta motivasi kepada penulis, selalu ada baik dalam suka maupun duka yang dialami oleh penulis, serta selalu ada untuk memberikan solusi dalam permasalahan yang dialami oleh penulis. Semoga persahabatan dan kebersamaan kita tidak hilang ditelan waktu.
10. Keluarga besar **Sistem Informasi 2014, 2015, 2016, 2017, 2019, 2020,2021**, maupun Keluarga Besar **FMIPA Unhas** atas segala bentuk dukungan dan bantuan selama proses perkuliahan. Semoga kesuksesan selalu kita dapatkan dalam setiap langkah-langkah kita.
11. Seluruh pihak yang tidak dapat disebutkan satu per satu atas segala bentuk kontribusi, partisipasi, serta motivasi yang diberikan kepada penulis selama ini. Semoga apa yang kita berikan, dilipatgandakan oleh Tuhan Yang Maha Kaya.

Penulis menyadari bahwa masih banyak kekurangan dalam skripsi ini dikarenakan terbatasnya pengalaman dan pengetahuan yang dimiliki oleh penulis. Untuk itu, dengan segala kerendahan hati penulis mohon maaf sebesar-besarnya serta mengharapkan segala bentuk saran serta masukan bahkan kritik yang membangun dari segala pihak. Akhir kata, saya berharap Allah SWT berkenan membalas segala kebaikan semua pihak yang telah membantu. Semoga skripsi ini dapat bermanfaat bagi siapapun yang membacanya, terutama untuk pengembangan ilmu pengetahuan.

Makassar, 14 Juni 2023



Al Waasiu



**PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK  
KEPENTINGAN AKADEMIS**

---

Sebagai sivitas akademik Universitas Hasanuddin, saya yang bertanda tangan di bawah ini:

Nama : Al Waasiu  
Nim : H071181007  
Program Studi : Sistem Informasi  
Departemen : Matematika  
Fakultas : Matematika dan Ilmu Pengetahuan Alam  
Jenis Karya : Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Hasanuddin **Hak Bebas Royalti Non Eksklusif (*Non-exclusive Royalty-Free Right*)** atas karya ilmiah saya yang berjudul :

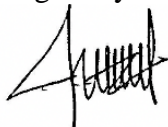
**“Implementasi Arsitektur *Inception Resnet-V2* Dalam Diagnosis Penyakit  
Parkinson Berbasis Android”**

beserta perangkat yang ada (jika diperlukan). Terkait dengan hal di atas, maka pihak universitas berhak menyimpan, mengalih-media/format-kan, mengelola dalam bentuk pangkalan data (*database*), merawat, dan mempublikasikan tugas akhir saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikian pernyataan ini saya buat dengan sebenarnya.

Dibuat di Makassar pada tanggal 14 Juni 2023

Yang menyatakan



(Al Waasiu)

## ABSTRAK

Penyakit parkinson atau *parkinson disease* merupakan salah satu penyakit gangguan *neurodegeneratif* kronis kedua setelah penyakit alzheimer yang hingga saat ini belum ada obat yang dapat menyembuhkan penyakitnya sehingga penatalaksanaan ditujukan hanya untuk menghambat progresivitas penyakit dan memperbaiki gejala. Penyakit parkinson ditandai dengan hilangnya neuron dopaminergik terutama di *pars compacta substantia nigra (SNc)*. Penyebab pasti kematian saraf ini masih belum diketahui hingga saat ini. Dokter dan peneliti yang berasal dari *Unified Parkinson's Disease Rating Scale (UPDRS)* sebelumnya telah mengusulkan menggunakan *Composite Index of Speed and Pen-pressure (CISP)* dari pembuatan sketsa sebagai fitur untuk menganalisis tingkat keparahan penyakit parkinson dengan sukses di tahap pertama. Oleh karena itu, penelitian ini bertujuan untuk membuat aplikasi android pengklasifikasi penyakit parkinson agar orang-orang lebih mudah mencari informasi mendiagnosa sedini mungkin menggunakan gambar sketsa penyakit parkinson. Klasifikasi gambar penyakit parkinson dapat dilakukan secara akurat dengan menggunakan algoritma *Convolutional Neural Network. Inception* merupakan pengembangan dari *Convolutional Neural Network (CNN)* salah satunya adalah arsitektur *Inception-Resnet-V2* yang menghasilkan performa yang sangat baik dengan komputasi yang relatif rendah. Model yang dihasilkan dari *training* gambar sebanyak 50 epoch memiliki performa sebesar 91,38% dengan presisi 85% dan *recall* 80 % serta *F1-score* 82%, dimana label negatif menyatakan pasien sehat dan label positif menyatakan penyakit parkinson.

Kata kunci : *Convolutional Neural Network (CNN), Inception ResNet-V2, Penyakit parkinson.*

**ABSTRACT**

*Parkinson's disease, is one of the most common chronic neurodegenerative disorders, second only to Alzheimer's disease. To date, there is no cure for the disease, and management is aimed at slowing the progression of the disease and improving symptoms. Parkinson's disease is characterized by the loss of dopaminergic neurons, particularly in the pars compacta of the substantia nigra (SNc). The exact cause of the death of these neurons is still unknown. Doctors and researchers from the Unified Parkinson's Disease Rating Scale (UPDRS) have previously proposed using the Composite Index of Speed and Pen-pressure (CISP) from sketching as a feature to analyze the severity of Parkinson's disease, with success in the first stage. Therefore, this study aims to create an Android application for Parkinson's disease classification so that people can easily search for information and diagnose the disease as early as possible using sketches of Parkinson's disease. The classification of Parkinson's disease images can be accurately done using the Convolutional Neural Network algorithm. Inception is a development of the Convolutional Neural Network (CNN) algorithm, and one of its architectures, the Inception-Resnet-V2, produces excellent performance with relatively low computation. The model produced from training 50 epochs of images has a performance of 91.38% with precision of 85%, recall of 80%, and F1-score of 82%, where the negative label represents healthy patients and the positive label represents Parkinson's disease.*

*Keywords: Convolutional Neural Network (CNN), Inception ResNet-V2, Parkinson's disease.*

## DAFTAR ISI

HALAMAN JUDUL.....	ii
PERNYATAAN KEASLIAN.....	iii
KATA PENGANTAR .....	vi
PERNYATAAN PERSETUJUAN PUBLIKASI TUGAS AKHIR UNTUK KEPENTINGAN AKADEMIS .....	ix
ABSTRAK .....	x
ABSTRACT.....	xi
DAFTAR ISI .....	xxi
DAFTAR GAMBAR .....	xv
DAFTAR TABEL.....	xvii
BAB 1 PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	4
1.3 Tujuan Penelitian .....	4
1.4 Batasan Masalah.....	5
1.5 Manfaat Penelitian .....	5
BAB II TINJAUAN PUSTAKA.....	6
2.1 Penelitian Terkait .....	6
2.2 Landasan Teori.....	7
2.2.1 Penyakit Parkinson.....	7
2.2.2 Citra digital.....	8
2.2.3 Deep Learning.....	10
2.2.4 Augmentasi Data.....	11
2.2.5 Convolutional Neural Network.....	12
2.2.5.1 Konvolusi .....	13
2.2.5.2 Padding.....	16
2.2.5.3 Stride .....	17
2.2.5.4 Fungsi Aktivasi .....	18
2.2.5.5 Pooling Layer .....	19
2.2.5.6 Flatten.....	19

2.2.5.7 Fully Connected Layer .....	20
2.2.6 Dropout .....	20
2.2.7 Tensorflow .....	21
2.2.8 Batch Size and Epoch.....	22
2.2.9 Categorical Crossentropy .....	22
2.2.10 Inception-Resnet .....	22
2.2.11 Confusion Matrix .....	27
<b>BAB III METODE PENELITIAN.....</b>	<b>30</b>
3.1 Waktu dan Tempat .....	30
3.2 Sumber Data.....	30
3.3 Tahapan Penelitian .....	30
3.3.1 Preprocessing data.....	30
3.3.2 Pembagian data .....	30
3.3.3 Training Data .....	30
3.3.4 Evaluasi model.....	31
3.3.5 Performansi dan Analisis Kinerja .....	31
3.3.6 Konversi Model.....	31
3.3.7 Build Aplikasi .....	31
3.4 Timeline Penelitian. ....	32
3.5 Rancangan Aplikasi Android .....	33
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>	<b>35</b>
4.1 Deskripsi Data.....	35
4.2 Pembagian Data .....	35
4.3 Implementasi Inception ResNet-V2.....	36
4.3.1 Preprocessing Data .....	36
4.4 Model Initialization & Training .....	37
4.5 Analisis Kinerja.....	40
4.5.1 Confusion Matrix pada Inception ResNet-V2.....	40
4.5.2 Presisi, Recall, dan F1-Score .....	41
4.6 Deploy Model.....	42
4.6.1 Save Model.....	43
4.6.2 Convert dan Save Model .....	43

4.7 Aplikasi Android.....	44
4.7.1 Desain Layout Aplikasi.....	44
4.7.2 Add TensorFlow Lite Model.....	45
4.7.3 Coding Activity.....	45
4.7.4 Visualisasi Aplikasi Android .....	47
4.7.5 Pengujian Aplikasi .....	48
BAB V.....	49
KESIMPULAN DAN SARAN.....	49
5.1 Kesimpulan .....	49
5.2 Saran.....	49
DAFTAR PUSTAKA .....	50
LAMPIRAN .....	53

## DAFTAR GAMBAR

Gambar 2.1. Spiral dan wave sehat.....	8
Gambar 2.2. Parkinson spiral dan wave.....	8
Gambar 2.3. Citra Biner.....	9
Gambar 2.4. Citra Grayscale.....	10
Gambar 2.5. Citra RGB.....	10
Gambar 2.6. Augmentasi dengan perputaran gambar maksimal sebesar $90^\circ$ .....	11
Gambar 2.7. Arsitektur CNN.....	12
Gambar 2.8. Proses konvolusi.....	14
Gambar 2.9. Contoh operasi konvolusi matriks citra yang memiliki 3 channel..	15
Gambar 2.10. Contoh operasi konvolusi multiple kernel.....	16
Gambar 2.11. Penambahan padding pada input.....	17
Gambar 2.12. Stride 1 yang bergeser secara horizontal.....	17
Gambar 2.13. Stride 2 yang bergeser secara horizontal.....	18
Gambar 2.14. Contoh operasi max pooling dan average pooling.....	19
Gambar 2.15. Proses flattening.....	20
Gambar 2.16 Skema untuk arsitektur Inception ResNet-V2.....	23
Gambar 2.17. Stem Inception Resnet-V2.....	25
Gambar 2.18. Modul A Inception ResNet-V2.....	26
Gambar 2.19. Modul B Inception ResNet-V2.....	26
Gambar 2.20. Modul C Inception ResNet-V2.....	26
Gambar 2.21. Reduction block A Inception ResNet-V2.....	27
Gambar 2.22. Reduction block B Inception ResNet-V2.....	27
Gambar 3.1. Diagram alur penelitian.....	32
Gambar 3.2. Timeline alur penelitian.....	33
Gambar 3.3. Rancangan dari aplikasi android.....	34
Gambar 4.1. Spiral dan wave sehat.....	35
Gambar 4.2. Parkinson spiral dan wave.....	35
Gambar 4.3. Citra spiral dan wave sehat sebelum dan setelah resize.....	36
Gambar 4.4. Zoom in & zoom out sketsa spiral.....	37
Gambar 4.5. Zoom in & zoom out sketsa wave (gelombang).....	37

Gambar 4.6. Plot Training Accuracy pada arsitektur Inception ResNet-V2.....	39
Gambar 4.7. Plot Training Loss pada arsitektur Inception ResNet-V2 .....	39
Gambar 4.8. Kurva ROC Inception ResNet V-2 pada epoch ke-50 .....	41
Gambar 4.9. Merupakan diagram batang Inception ResNet-V2.....	42
Gambar 4.10. Layout activity_main2.xml .....	44
Gambar 4.11. Halaman activity_main2.xml .....	45
Gambar 4.12. Model.tflite android studio.....	45
Gambar 4.13. State machine diagram .....	47
Gambar 4.14. Use case diagram.....	47
Gambar 4.15. Visualisasi aplikasi diagnosis penyakit parkinson .....	47



**DAFTAR TABEL**

Tabel 4.1. Pembagian dataset.....	36
Tabel 4.2. Hasil training accuracy pada modul Inception ResNet-V2.....	37
Tabel 4.3. Bentuk confusion matrix pada Inception ResNet-V2 .....	40
Tabel 4.4. Precison, recall, dan F1-score pada Inception ResNet-V2.....	42
Tabel 4.5. Testing Aplikasi .....	48

## BAB 1 PENDAHULUAN

### 1.1 Latar Belakang

Gangguan *neurodegeneratif* merupakan suatu istilah keadaan dimana terjadi kehilangan progresif terhadap struktur atau fungsi sel neuron, termasuk kematian sel *neuron*. Beberapa contoh gangguan *neurodegeneratif* yang banyak terjadi pada usia lanjut baik ringan maupun gangguan berat seperti penyakit parkinson, penyakit alzheimer dan penyakit huntington (Tiwari, 2016).

Penyakit parkinson atau *parkinson disease* merupakan salah satu penyakit gangguan *neurodegeneratif* kronis kedua setelah penyakit alzheimer yang hingga saat ini belum ada obat yang dapat menyembuhkan penyakitnya sehingga penatalaksanaan ditujukan hanya untuk menghambat progresivitas penyakit dan memperbaiki gejala (Salmanpour dkk., 2019). Penyakit parkinson ditandai dengan hilangnya *neuron dopaminergik* terutama di *pars compacta substansia nigra* (SNc). Penyebab pasti kematian saraf ini masih belum diketahui hingga saat ini. Konsekuensi langsungnya adalah kadar dopamin di daerah striatal otak turun tajam (Armañanzas dkk., 2013). Pada tahap awal penyakit parkinson menyebabkan ekspresi wajah yang datar, lengan tidak berayun saat berjalan, suara menjadi lebih kecil hingga gangguan gerak tubuh dimana tangan bergetar (*tremor*) saat tidak di gerakkan, kaku saat bergerak (*rigiditas*), kesulitan menulis serta gerakan yang melambat.

Menurut data dari *Global Burden of Disease, Injuries, and Risk Factors Study* (GBD, 2015), parkinson merupakan penyakit dengan tingkat penyebaran disabilitas dan kematian yang peningkatannya paling cepat diantara penyakit lainnya seperti alzheimer. Menurut studi analisis sistematis menyatakan bahwa pada tahun 2016 terdapat 6,1 juta orang dengan penyakit parkinson di seluruh dunia. Jumlah penderita parkinson sebanyak 6,1 juta mengalami peningkatan sebanyak 2,4 kali lipat dibandingkan tahun 1990 dimana penderita berjumlah hanya 2,5 juta orang diseluruh dunia.

Penderita parkinson yang berjenis kelamin pria lebih banyak dibandingkan wanita, penderita pria berjumlah 3,2 juta orang dan wanita berjumlah 2,9 juta orang.

Di antara seluruh jumlah penderita parkinson pada tahun 2016, 2,1 juta berasal dari negara dengan indeks sosiodemografik tinggi, 3,1 juta berasal dari negara dengan indeks sosiodemografik menengah, dan 0,9 juta berasal dari negara indeks sosiodemografik rendah. Tingkat penyebaran penyakit parkinson meningkat seiring pertambahan umur setelah umur 50 tahun, dengan puncak yaitu umur 85-89 tahun dan menurun setelah umur 89 tahun. Pemeriksaan penyakit parkinson dapat di lihat dari hasil gambar pasien untuk memberikan pencegahan maupun rehabilitasi/terapi terjadinya perkembangan penyakit pneumonia aspirasi menjadi penyakit parkinson. Untuk menghindari dampak negatif yang signifikan pada pasien penyakit parkinson, identifikasi penyakit parkinson pada tahap utama adalah wajib.

Dokter dan peneliti yang berasal dari *Unified Parkinson's Disease Rating Scale* (UPDRS) sebelumnya telah mengusulkan menggunakan *Composite Index of Speed and Pen-pressure* (CISP) dari pembuatan sketsa sebagai fitur untuk menganalisis tingkat keparahan penyakit parkinson dengan sukses di tahap pertama. Sketsa *spiral* dan *wave* (gelombang), dan tulisan tangan dapat dengan mudah dibedakan dari orang sehat (*healthy*) dengan orang yang terkena penyakit parkinson (*parkinson disease*) dan pengukuran sketsa dan tulisan tangan tersebut *non-invasif*. Gejala penyakit parkinson secara luas dibagi menjadi dua kelompok. Salah satunya adalah gejala motorik dan gejala non-motorik. Gejala motorik seperti *tremor* (gerakan kaki/tangan yang tidak disengaja), kekakuan (kesulitan menggerakkan anggota tubuh), kelambatan dalam aktivitas sehari-hari, gangguan keseimbangan, gaya berjalan terseok-seok. Di sisi lain, gejala non-motorik seperti kesulitan dengan memori, kelambatan berpikir, kecemasan dan depresi, insomnia dan kelelahan, masalah penglihatan, halusinasi dan delusi, masalah bicara dan menelan. Penyakit parkinson dengan gejala motorik menyebabkan tiga perubahan utama: dalam tulisan, ukuran tulisan (*micrographia*), tekanan pena dan kinematika. Gejala motorik berupa tremor dengan mengenali gambar antara subjek sehat dan parkinson. Oleh karena itu, klasifikasi biner ini dikembangkan untuk mendeteksi penyakit parkinson dengan memanfaatkan *Convolutional Neural Network* (Jahan dkk., 2021).

Saat ini sistem keputusan dan diagnosis berbasis komputer yang disebut sistem berbantuan komputer telah menjadi populer dengan akurasi tinggi serta hasil yang

konsisten dan efisien. Seperti berbagai aplikasi biomedis, diagnosis penyakit parkinson merupakan masalah klasifikasi yang penting. Metode yang di gunakan dapat menghasilkan hasil yang berbeda sesuai dengan data yang diatur. Dalam hal ini, berbagai algoritma *Machine Learning* harus diuji untuk menemukan metode yang berguna untuk data gambar penyakit parkinson. Proses klasifikasi gambar penyakit parkinson dapat dilakukan secara akurat dengan menggunakan algoritma *Convolutional Neural Network*. *Convolutional Neural Network* merupakan cabang dari *Deep Learning* yang biasa digunakan pada data *image*. CNN digunakan karena dapat menangani masalah yang kompleks dan memiliki kinerja yang baik dari hasil penelitian terkait.

CNN pertama kali dikembangkan dengan nama NeoCognitron oleh Kunihiko Fukushima, seorang peneliti dari NHK *Broadcasting Science Research Laboratories*, Kinuta, Setagaya, Tokyo, Jepang. Konsep tersebut kemudian dimatangkan oleh Yann LeChun, seorang peneliti dari *AT&T Bell Laboratories* di Holmdel, New Jersey, USA. Model CNN dengan nama *LeNet* berhasil diterapkan oleh LeChun pada penelitiannya mengenai pengenalan angka dan tulisan tangan. Pada tahun 2012, Alex Krizhevsky dengan penerapan CNN miliknya berhasil menjuarai kompetisi *ImageNet Large Scale Visual Recognition Challenge 2012*. Prestasi tersebut menjadi momen pembuktian bahwa metode *Deep Learning*, khususnya CNN. Metode CNN terbukti berhasil mengungguli metode *Machine Learning* lainnya seperti SVM pada kasus klasifikasi objek pada citra (Suartika E. P, I Wayan dkk., 2016).

*Inception* merupakan pengembangan dari *Convolutional Neural Network* (CNN) yang pertama kali diperkenalkan oleh Szegedy, dkk., pada tahun 2014 dalam paper berjudul “*Going Deeper with Convolutions*”. *Very deep convolutional networks* telah menjadi pusat pengembangan dalam performa *image recognition* belakangan ini. Contohnya adalah arsitektur *Inception* yang menghasilkan performa yang sangat baik dengan komputasi yang relatif rendah. *Inception-Resnet-V2* adalah variasi dari model *Inception V3*, dan ini jauh lebih dalam dari *Inception V3* sebelumnya. Pada *block* konvolusi pada *Inception Resnet-V2* memiliki jaringan yang sama di mana blok sisa (*residual*) yang berulang telah dikompresi. Kesuksesan *residual block* yang awal mula di ciptakan pada

arsitektur *ResNet* yang menjadi juara LSVRC 2015 (*ImageNet Large Scale Visual Recognition Challenge*) yang mengatasi *Vanishing Gradient* yang terjadi karena banyaknya layer konvolusi. Oleh karena itu diperlukan koneksi *residual* melatih arsitektur yang sangat dalam. *Inception ResNet-V2* adalah arsitektur CNN yang dibangun di atas varian arsitektur *Inception* tetapi menggabungkan koneksi *residual* (menggantikan tahap rangkaian filter dari arsitektur *Inception*). *Inception-ResNet-V2* mendapat semua manfaat dari pendekatan *residual* sambil mempertahankan efisiensi komputasinya

Berdasarkan uraian permasalahan yang telah dijelaskan di atas, maka dari itu penulis tertarik untuk melakukan penelitian dengan menggunakan arsitektur *Inception ResNet-V2* sebab belum ada penelitian terkait arsitektur tersebut dalam mengklasifikasikan dataset penyakit parkinson. Sehingga peneliti memutuskan untuk membuat penelitian yang berjudul “Implementasi Arsitektur *Inception ResNet-V2* Dalam Diagnosis Penyakit Parkinson Berbasis Android”.

## 1.2 Rumusan Masalah

Berdasarkan latar belakang masalah yang sudah diuraikan di atas, rumusan masalah dalam penelitian adalah sebagai berikut.

1. Bagaimana performa klasifikasi menggunakan arsitektur *Inception ResNet-V2* pada dataset penyakit parkinson dengan menggunakan sketsa gambar *spiral* dan *wave* (gelombang)?
2. Bagaimana men-*deploy* dataset penyakit parkinson kedalam aplikasi berbasis android ?

## 1.3 Tujuan Penelitian

Berdasarkan rumusan masalah yang sudah diuraikan di atas, maka tujuan penelitian adalah sebagai berikut.

1. Mengetahui performa klasifikasi menggunakan arsitektur *Inception ResNet-V2* pada dataset penyakit parkinson dengan menggunakan sketsa gambar *spiral* dan *wave* (gelombang).
2. Dapat men-*deploy* dataset penyakit parkinson kedalam aplikasi berbasis android.

#### 1.4 Batasan Masalah

1. Algoritma yang di gunakan merupakan *Convolutional Neural Network* (CNN) dengan arsitektur *Inception ResNet-V2*.
2. Dataset yang di gunakan berasal dari situs Kaggle dengan dataset penyakit parkinson.

#### 1.5 Manfaat Penelitian

Hasil penelitian ini diharapkan dapat bermanfaat :

1. Sebagai informasi mengenai performa penggunaan *convolutional neural network* dari arsitektur *Inception ResNet-V2*.
2. Menyediakan aplikasi berbasis android yang mampu dalam mengklasifikasikan penyakit parkinson menggunakan sketsa gambar *spiral* dan *wave* (gelombang).

## BAB II

### TINJAUAN PUSTAKA

#### 2.1 Penelitian Terkait

Dalam menghindari unsur plagiasi dengan penelitian sebelumnya, penulis menjadikan penelitian sebelumnya sebagai acuan. Berikut ini adalah penelitian-penelitian tentang penyakit parkinson.

Pada penelitian (Jahan dkk., 2021), terdapat judul “*Parkinson's Disease Detection Using ResNet50 with Transfer Learning*” peneliti ini telah dipelajari gejala motorik halus berupa sketsa. Data dari dataset Kaggle yaitu kumpulan sketsa spiral dan gelombang dari 55 subjek dimana 28 subjek dari kelompok kontrol sehat (tanpa PD) dan 27 subjek dari kelompok parkinson. Dataset berisi 102 gambar sketsa spiral dan 102 gambar sketsa gelombang. Sebuah tablet, kertas ukuran A3, dan pena digunakan untuk merekam sketsa. Peneliti mengusulkan sebuah sistem yang dapat menentukan sketsa dan laporan apakah sketsa pasien PD atau tidak. Algoritma *Deep Learning* dapat menangani solusi jaringan saraf generalisasi otak yang berbeda dengan desain yang sama. Jadi, kami menerapkan *Convolutional Neural Network* (CNN) untuk mengklasifikasikan gambar sketsa untuk membedakan atau mengidentifikasi pasien yang terkena Penyakit Parkinson (PD) dari kelompok kontrol sehat (tanpa PD) biasa. Eksperimen dilakukan pada model CNN yang berbeda dengan metode *transfer learning* dan menerapkan data sketsa Spiral dan Gelombang. Sistem yang diusulkan mencapai akurasi 96,67% pada model ResNet50 dengan sketsa spiral.

Pada penelitian (Chakraborty dkk., 2020), terdapat judul “*Parkinson's Disease Detection from Spiral and Wave Drawings using Convolutional Neural Networks: A Multistage Classifier Approach*” peneliti mengusulkan metode CNN dengan beberapa *Multiple Machine Learning Classifier*. Penulis bekerja dengan kumpulan data gambar spiral dan gelombang yang dikumpulkan dari penyimpanan data Kaggle. Penulis mengusulkan model CNN yang dijalankan di atas dua kumpulan data yang berbeda ini, kumpulan data sketsa spiral dan gelombang. Penulis menghitung probabilitas prediksi dan kemudian dilatih pada *Ensemble Classifier* RFC (*Random Forest Classifier*) dan LR (*Linear Regression*). Kemudian penulis

menggunakan voting rata-rata tertimbang untuk menggabungkan hasil pengujian sketsa spiral dan gelombang. Penulis mencapai akurasi 93,3%.

Pada penelitian (Gil-Martín dkk., 2019), terdapat judul “*Parkinson’s Disease Detection from Drawing Movements Using Convolutional Neural Networks*” peneliti mendeteksi penyakit Parkinson berdasarkan arah yang berbeda selama gerakan menggambar. Ada yang menggunakan transformasi *Fast Fourier* pada input dan diterapkan pada metode CNN. Peneliti membagi model menjadi dua bagian dimana bagian pertama terdiri dari lapisan konvolusi 2-lapisan untuk ekstraksi fitur dan bagian kedua terdiri dari lapisan 3-lapisan yang terhubung penuh untuk tujuan klasifikasi. Peneliti memperoleh akurasi 96,5%, skor F1 97,7%.

## 2.2 Landasan Teori

### 2.2.1 Penyakit Parkinson

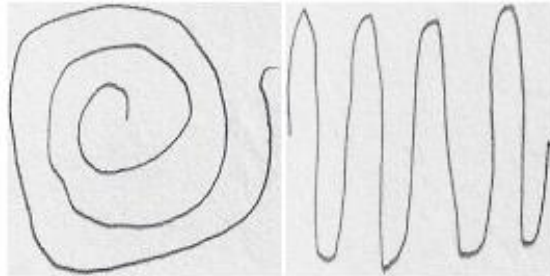
Penyakit parkinson merupakan gangguan fungsi otak yang disebabkan oleh proses *degenerasi ganglia basalis* pada sel *substansia nigra pars compacta* (SNc) dan ditandai dengan karakteristik seperti *tremor* saat istirahat, kekakuan otot dan sendi (*rigidity*), kelambanan gerak dan bicara (*bradikinesia*) serta instabilitas posisi tegak (*postural instability*). Penyakit ini adalah penyakit *neurodegeneratif* yang paling sering terjadi setelah Alzheimer dan lebih sering terjadi pada laki-laki dibandingkan perempuan. Penyakit Parkinson dimulai perlahan, tidak disadari, berangsur-angsur memburuk dan mempengaruhi kualitas hidup (Hanriko & Anzani, 2018). Penyakit Parkinson memiliki etiologi yang belum diketahui secara pasti, namun dapat disebabkan oleh faktor-faktor seperti faktor genetik, faktor lingkungan, umur, ras, cedera kranioserebral dan stress emosional.

Berdasarkan penelitian epidemiologi, umur dan ras juga berpengaruh terhadap terjadinya penyakit parkinson. Proses menua dapat menjadi faktor risiko yang mempermudah terjadinya proses degenerasi di SNc dan angka kejadian penyakit parkinson ditemukan lebih tinggi pada ras kulit putih dibandingkan kulit berwarna.

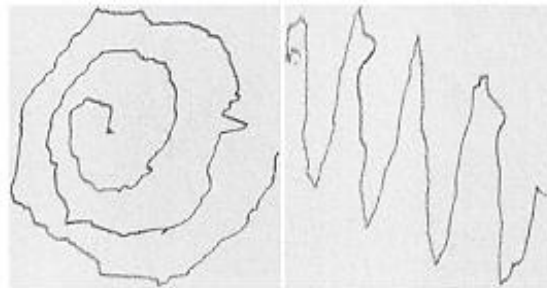
Dokter dan peneliti sebelumnya telah menggunakan tulisan tangan dan sketsa *spiral* dan *wave* untuk mengidentifikasi pasien Penyakit Parkinson dengan sukses di tahap utama . Sketsa *spiral* dan *wave*, dan tulisan tangan dapat dengan mudah dibedakan dari orang sehat (*healthy*) dengan orang yang terkena penyakit parkinson



(*parkinson disease*) dan pengukuran sketsa & tulisan tangan tersebut *non-invasif* (Jahan dkk., 2021). Pada gambar 2.1 menunjukkan contoh gambar sketsa *spiral* dan *wave* dari orang sehat sedangkan pada gambar 2.2 menunjukkan contoh gambar sketsa *spiral* dan *wave* (gelombang) dari pasien Parkinson.



Gambar 2.1. Spiral dan wave sehat



Gambar 2.2. Parkinson spiral dan wave

( sumber : [kaggle.com](https://www.kaggle.com) )

Pada Penyakit Parkinson, tidak ada diet khusus yang direkomendasikan, namun dianjurkan untuk selalu memakan makanan sehat seperti buah dan sayur untuk meningkatkan asupan serat dan membantu meringankan konstipasi. Dianjurkan pula mengkonsumsi bahan makanan yang mengandung antioksidan untuk memperlambat progresivitas Penyakit Parkinson. Antioksidan akan menghilangkan radikal bebas sehingga mengurangi stres oksidatif yang berhubungan dengan kematian sel dan penuaan. Bahan makanan yang mengandung antioksidan tinggi antara lain yaitu anggur, sayuran hijau seperti brokoli, wortel, teh hijau, kentang, kacang, ikan seperti tuna dan salmon.

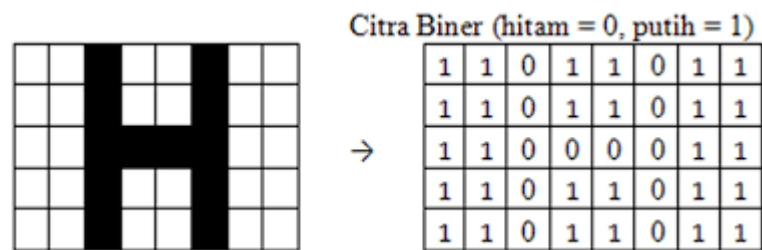
### 2.2.2 Citra digital

Citra adalah representasi objek dua dimensi dari dunia visual, menyangkut berbagai macam disiplin ilmu yang mencakup seni, human vision, astronomi,

teknik, dan sebagainya. Merupakan suatu kumpulan piksel-piksel atau titik-titik yang berwarna yang berbentuk dua dimensi (Jumadi dkk., 2021).

Citra digital adalah gambar dua dimensi yang bisa ditampilkan pada layar komputer sebagai himpunan/ diskrit nilai digital yang disebut pixel/ *picture elements*. Sebuah citra digital dapat mewakili oleh sebuah matriks yang terdiri dari M kolom dan N baris, dimana perpotongan antara kolom dan baris disebut piksel. Pixel yaitu elemen terkecil dari sebuah citra. Piksel mempunyai dua parameter, yaitu koordinat dan intensitas atau warna.

Beberapa jenis citra digital yang sering digunakan adalah citra biner, citra grayscale, dan citra warna. Citra biner (*binary image*) merupakan salah satu cara dalam merepresentasikan citra digital dimana citra ini menggunakan dua jenis warna saja, yaitu hitam dan putih. Citra biner disebut juga dengan citra W&B (*White&Black*) atau citra monokrom. Kedua warna ini masing – masing diwakili oleh angka – angka biner (0 dan 1).

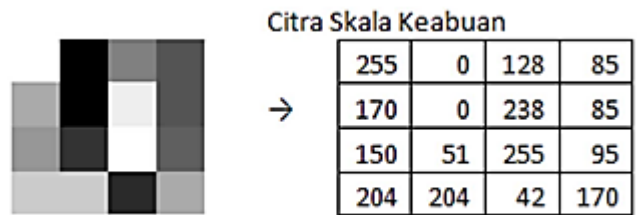


Gambar 2.3. Citra biner (*binary image*)

( sumber : [wordpress.com](http://wordpress.com) )

Pada gambar 2.3 ditunjukkan sebuah citra biner dengan representasi citra memiliki nilai 0 dan 1. Warna hitam representasi citra ini bernilai 0 dan putih bernilai 1.

Citra *grayscale* adalah citra digital yang hanya memiliki satu nilai channel pada pixelnya. Dengan kata lain, nilai bagian *red*, *green*, dan *blue* memiliki warna yang sama, yaitu warna hitam, keabuan, dan putih. Nilai-nilai tersebut digunakan untuk menunjukkan intensitas warna. Citra yang ditampilkan dari citra jenis ini terdiri atas warna abu-abu, bervariasi pada warna hitam pada bagian yang intensitas terlemah dan warna putih pada intensitas terkuat. Citra *grayscale* mempunyai kemungkinan warna hitam untuk nilai minimal dan warna putih untuk nilai maksimal.

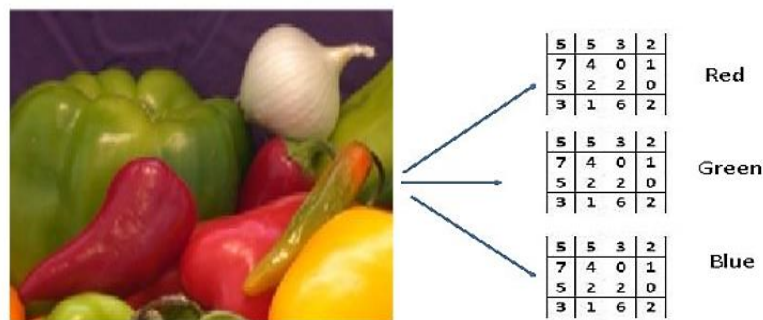


Gambar 2.4. Citra *Grayscale*

( sumber : [embeddednesia.com](http://embeddednesia.com) )

Pada gambar 2.4 ditunjukkan sebuah citra yang memiliki 2 warna yaitu hitam dan keabuan. Setiap piksel mewakili derajat keabuan dengan nilai antara 0 (hitam) sampai 255.

Citra warna merupakan metode dalam merepresentasikan suatu citra secara digital, dimana metode ini menggunakan kombinasi dari tiga warna primer (merah, hijau dan biru = RGB) untuk membentuk suatu citra. Adapun setiap titik pada citra mewakili kombinasi dari ketiga warna ini. Setiap warna dasar memiliki intensitas tersendiri dengan nilai minimum nol (0) dan nilai maksimum 255. Red memiliki warna minimum putih dan warna maksimum merah, Green memiliki warna minimum putih maksimum hijau, dan Blue memiliki warna minimum putih dan maksimum biru.



Gambar 2.5. Citra RGB

( sumber : [pemrogramanmatlab.com](http://pemrogramanmatlab.com) )

### 2.2.3 Deep Learning

Belakangan ini *Deep Learning* menjadi sorotan dalam pengembangan *Machine Learning*. Alasannya karena *Deep Learning* telah mencapai hasil yang luar biasa dalam visi computer. *Deep Learning* merupakan cabang dari *Machine Learning*

yang terinspirasi dari otak manusia dengan menerapkan jaringan syaraf buatan yang memiliki banyak hidden layer. *Convolutional Neural Network* (CNN) merupakan salah satu metode dalam *Deep Learning* yang dibuat untuk menutupi kelemahan dari metode sebelumnya.

Terdapat beberapa kelemahan dalam metode sebelumnya, tetapi dengan model ini sejumlah parameter bebas dapat dikurangi dan deformasi gambar input seperti translasi, rotasi dan skala dapat ditangani. Seiring dengan banyaknya pengembangan dan riset tentang *Deep Learning*, banyak *library* yang bermunculan dengan fokus mempelajari tentang jaringan syaraf tiruan. Salah satu contohnya adalah keras (Santoso & Ariyanto, 2018). Keras merupakan library jaringan syaraf tiruan tingkat tinggi yang ditulis dengan bahasa python dan mampu berjalan di atas *TensorFlow*, *CNTK*, atau *Theano*. *Library* ini menyediakan fitur yang digunakan dengan fokus mempermudah pengembangan lebih dalam tentang *Deep Learning*.

#### 2.2.4 Augmentasi Data

Augmentasi data adalah suatu proses dalam pengolahan data gambar, augmentasi merupakan proses mengubah atau memodifikasi gambar sedemikian rupa sehingga komputer akan mendeteksi bahwa gambar yang diubah adalah gambar yang berbeda, namun manusia masih dapat mengetahui bahwa gambar yang diubah tersebut adalah gambar yang sama. Augmentasi dapat meningkatkan akurasi dari model CNN yang dilatih karena dengan augmentasi model mendapatkan data-data tambahan yang dapat berguna untuk membuat model yang dapat melakukan generalisasi dengan lebih baik. Augmentasi yang dilakukan pada penelitian ini adalah membalikan gambar secara horizontal, melakukan zoom-in secara acak, dengan maksimal zoom sebesar 50% dari besar gambar, dan juga melakukan rotasi gambar secara acak dengan derajat maksimal  $90^\circ$  (Mahmud dkk., 2019).



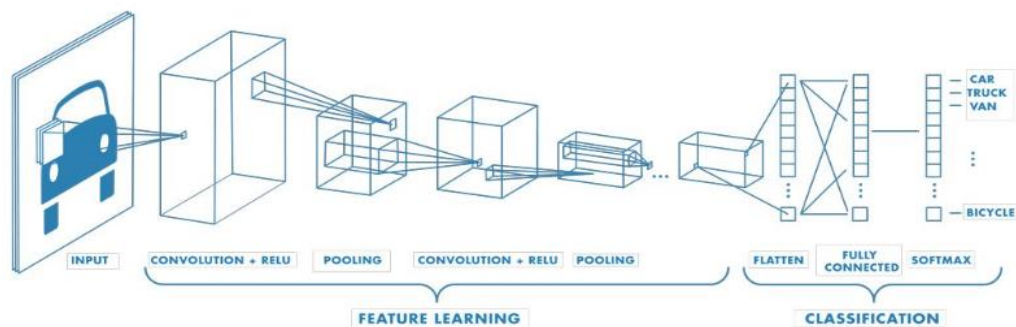
Gambar 2.6. Augmentasi dengan perputaran gambar maksimal sebesar  $90^\circ$

( sumber : [semanticscholar.org](https://www.semanticscholar.org) )

Pada gambar 2.6 ditunjukkan salah satu jenis augmentasi yang umum dilakukan adalah dengan melakukan perputaran gambar dengan besar tertentu.

### 2.2.5 Convolutional Neural Network

*Convolutional Neural Network* (CNN) adalah salah satu algoritma dari deep learning yang merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang dirancang untuk mengolah data dalam bentuk dua dimensi, misalnya gambar atau suara. CNN digunakan untuk mengklasifikasi data yang terlabel dengan menggunakan metode supervised learning. Cara kerja dari supervised learning adalah terdapat data yang dilatih dan terdapat variabel yang ditargetkan sehingga tujuan dari metode ini adalah mengelompokkan suatu data ke data yang sudah ada. Lapisan-lapisan CNN memiliki susunan neuron 3 dimensi (lebar, tinggi, kedalaman). Lebar dan tinggi merupakan ukuran lapisan, sedangkan kedalaman mengacu pada jumlah lapisan. Sebuah CNN dapat memiliki puluhan hingga ratusan lapisan yang masing-masing belajar mendeteksi berbagai gambar. Pengolahan citra diterapkan pada setiap citra latih pada resolusi yang berbeda, dan output dari masing-masing gambar yang diolah dan digunakan sebagai input ke lapisan berikutnya. Pengolahan citra dapat dimulai sebagai fitur yang sangat sederhana seperti kecerahan dan tepi atau meningkatkan kompleksitas pada fitur yang secara unik menentukan objek sesuai ketebalan lapisan (Santoso & Ariyanto, 2018). Berikut merupakan jaringan arsitektur *Convolutional Neural Network* :



Gambar 2.7. Arsitektur CNN

( sumber : [towardsdatascience.com](https://towardsdatascience.com) )

Pada gambar 2.7 diatas saat memasukkan input maka arsitektur dibagi menjadi 2 yaitu *feature learning* dan *classification*. Di *feature learning* ada 3 lapisan proses

dari ekstraksi fitur, lapisan ini akan melakukan proses untuk membentuk data agar mendapatkan suatu pola secara spesifik. Pada *classification*, model akan digunakan untuk melakukan klasifikasi objek berdasarkan kelasnya.

### 2.2.5.1 Konvolusi

Konvolusi adalah suatu istilah matematis yang berarti mengaplikasikan sebuah fungsi pada output fungsi lain secara berulang. Dalam pengolahan citra, konvolusi memiliki arti mengaplikasikan sebuah kernel pada citra. Kernel adalah sebuah matriks kecil dengan tinggi dan lebarnya lebih kecil dari matriks citra yang akan dikonvolusi. Kernel biasanya juga di sebut filter (Pohrel, 2019).

Dalam *machine learning*, input citra berbentuk array dua dimensi dan kernel merupakan parameter berbentuk array multidimensi yang disesuaikan dengan model algoritma. Konvolusi dapat digunakan pada lebih dari satu dimensi. Misalnya jika menggunakan gambar dua dimensi I sebagai input, maka kernel K juga berbentuk dua dimensi:

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(m, n)K(i - m, j - n) \quad (1)$$

Keterangan :

S (i,j) = Fungsi hasil konvolusi

I = Input

K = Kernel atau Filter

i,j = Pixel input

m,n = Pixel filter

Karena konvolusi bersifat komutatif maka berarti persamaan (1) dapat ditulis sebagai berikut:

$$S(i, j) = (K * I)(i, j) = \sum_m \sum_n I(i - m, j - n)K(m, n) \quad (2)$$

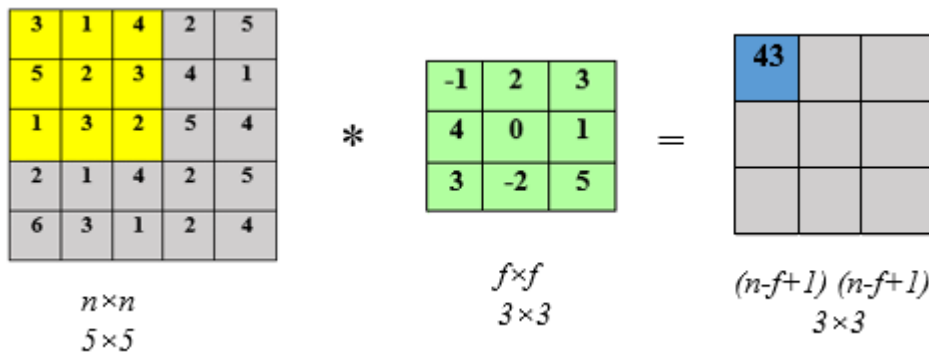
Sifat komutatif muncul karena telah dibalik relatif terhadap input. Sifat komutatif berguna untuk menulis bukti (proof) tapi sifat ini bukan merupakan sifat yang essensial pada implementasi jaringan syaraf tiruan. Terdapat fungsi serupa yang disebut *cross correlation* yang sama seperti konvolusi namun tanpa membalik kernel :

(3)

$$S(i, j) = (I * K)(i, j) = \sum_m \sum_n I(i + m, j + n)K(m, n)$$

Banyak *library machine learning* yang menerapkan *cross correlation* namun menyebutnya konvolusi. Pada penelitian ini, fungsi tersebut juga disebut konvolusi (Goodfellow dkk., 2016).

Tujuan melakukan konvolusi pada data citra adalah untuk mengekstraksi fitur dari citra input. Pada gambar 2.8 diilustrasikan sebuah matriks input berukuran 5 × 5 dikonvolusikan dengan kernel/filter berukuran 3 × 3, dengan stride atau perpindahannya sebanyak 1, dan zero padding. Hasil dari konvolusi tersebut ditunjukkan pada matriks result atau biasa disebut sebagai Feature Map (Albawi, 2017).



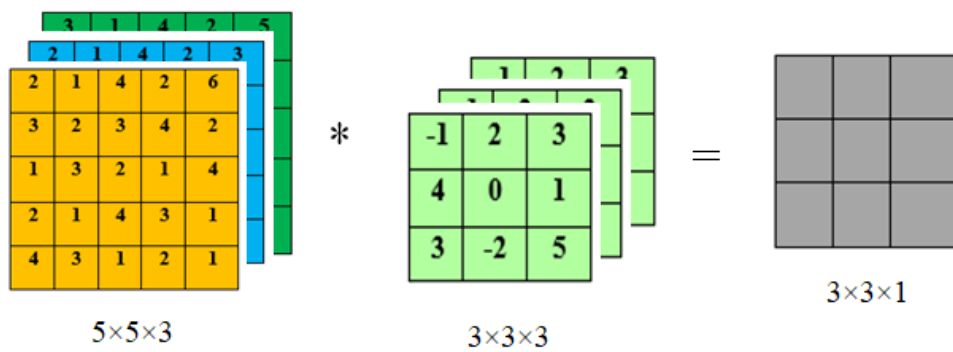
cara hitung *result* =  
 $(3 * -1) + (1 * 2) + (4 * 3) +$   
 $(5 * 4) + (2 * 0) + (3 * 1) +$   
 $(1 * 3) + (3 * -2) + (2 * 5) = 43$

Gambar 2.8. Proses konvolusi

Contoh ilustrasi pada gambar 2.8 adalah untuk citra yang hanya memiliki satu channel saja. Dalam melakukan konvolusi dengan citra RGB yang memiliki 3 channel, maka channel pada filter juga harus memiliki jumlah yang sama. Dengan kata lain dalam konvolusi, jumlah channel dalam input harus sama dengan jumlah channel pada kernel. Jika input citra berukuran  $n \times n \times n_c$  dengan  $n_c$  adalah jumlah channel atau layer pada input citra, maka persamaan (3) akan menjadi seperti berikut :

$$S(i, j) = (I * K)(i, j) = \sum_c^{n_c} \sum_m \sum_n I(i + m, j + n, c)K(m, n, c) \quad (4)$$

dengan  $c = 1 \dots n_c$



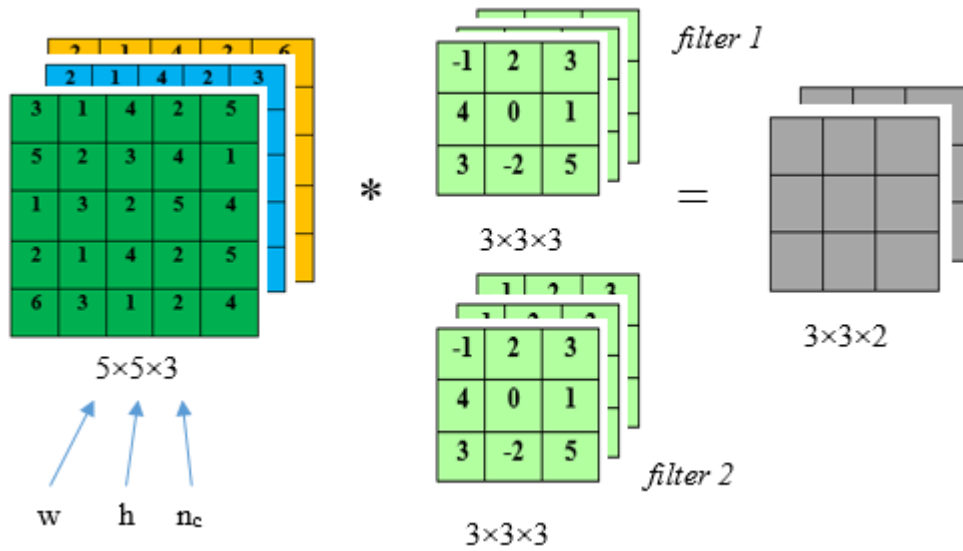
Gambar 2.9. Contoh operasi konvolusi matriks citra yang memiliki 3 channel

Pada gambar 2.9 diilustrasikan proses konvolusi pada input yang memiliki 3 channel. Hasil dari konvolusi ketiga channel ini hanya akan menghasilkan satu channel matriks saja. Masing-masing channel matriks akan dikonvolusikan dengan filter. Lalu hasil konvolusi dari masing-masing channel dijumlahkan untuk mendapatkan hasil dari matriks konvolusi atau *result*.

Dalam CNN juga terdapat operasi konvolusi yang menggunakan filter lebih dari 1. Biasanya operasi jenis ini dinamakan multiple kernel. Operasi untuk multiple kernel ini sama dengan operasi konvolusi sebelumnya. Hanya saja, hasil dari konvolusi multiple kernel ini memiliki jumlah channel yang sama dengan jumlah filternya. Jika  $nf$  adalah jumlah filter yang digunakan, dan  $k = 1 \dots nf$  maka persamaan (4) akan menjadi seperti berikut :

$$S(i, j) = (I * Kk)(i, j) = \sum_c^{nc} \sum_m \sum_n I(i + m, j + n, c) K k(m, n, c)$$





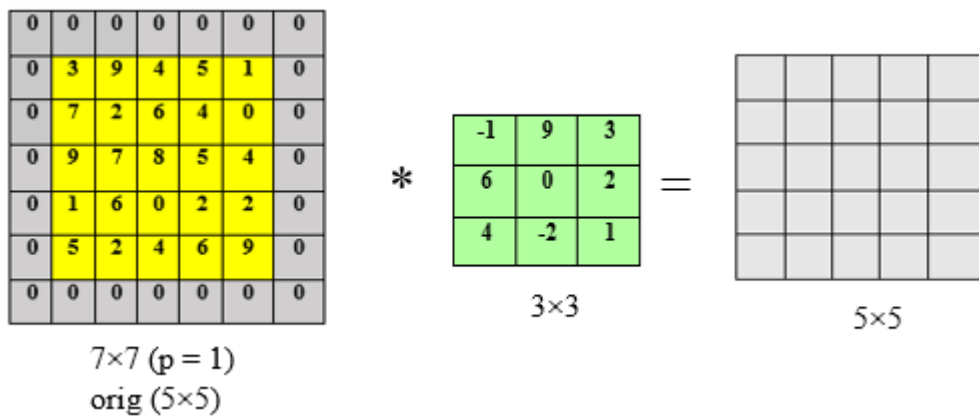
Gambar 2.10. Contoh operasi konvolusi multiple kernel

Misalkan pada gambar 2.10 memiliki matriks input berukuran  $5 \times 5$  dengan channel sebanyak 3, dikonvolusikan dengan 2 filter yang masing-masing memiliki channel sebanyak 3. Sehingga hasil konvolusi memiliki matriks dengan ukuran  $3 \times 3$  dengan channel sebanyak 2 (Priyono, 2018).

### 2.2.5.2 Padding

Padding atau Zero Padding adalah parameter yang menentukan jumlah piksel (berisi nilai 0) yang akan ditambahkan ke setiap sisi input. Ini digunakan untuk memanipulasi output dimensi lapisan konvolusi (*Feature Map*).

Tujuan penggunaan *padding layer* adalah dimensi *output* dari convolution layer akan selalu lebih kecil dari input (kecuali penggunaan filter  $1 \times 1$  dengan langkah 1) sehingga lebih banyak informasi yang terbuang yang tidak diperlukan saat proses konvolusi sedang berjalan (Dzulqarnain dkk., 2019). Selain itu, padding nol akan mengatur dimensi lapisan keluaran agar tetap sama dengan dimensi masukan atau setidaknya tidak berkurang drastis.



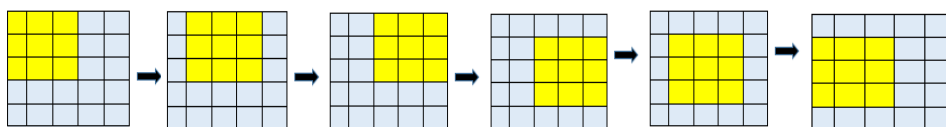
Gambar 2.11. Penambahan padding pada input

Pada gambar 2.11 diilustrasikan sebuah matriks input dengan ukuran asli sebesar  $5 \times 5$  dengan filter sebesar  $3 \times 3$ . Input ini kemudian diberi padding sebanyak satu sehingga membuat ukuran matriks input menjadi  $7 \times 7$ . Matriks yang telah diberi padding kemudian dikonvolusikan dengan filter sehingga menghasilkan feature map dengan ukuran yang sama dengan ukuran input aslinya yaitu  $5 \times 5$ .

### 2.2.5.3 Stride

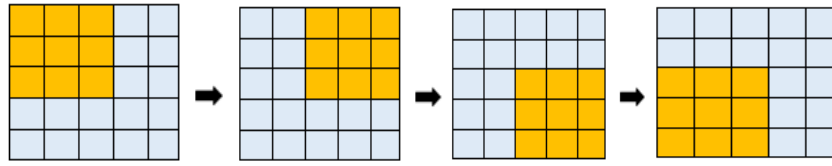
Stride adalah parameter yang menentukan jumlah pergeseran filter dalam piksel gambar. Jika stride nilainya 1, filter konvolusi akan bergeser 1 piksel secara horizontal dan vertikal. Lebih kecil nilai stride, model akan menangkap informasi yang lebih rinci dari gambar input, tetapi membutuhkan lebih banyak perhitungan jika dibandingkan dengan langkah besar.

Nilai stride kecil tidak selalu menghasilkan detail informasi piksel yang lebih baik, tetapi dengan nilai langkah kecil mencegah penumpukan informasi piksel yang tidak digunakan.



Gambar 2.12. Stride 1 yang bergeser secara horizontal

Pada gambar 2.12 diilustrasikan sebuah filter yang diaplikasikan pada input. Pergeseran filter ini ditentukan oleh besarnya ukuran stride. Nilai stride pada gambar 2.12 adalah 1, maka pergeseran kernel pada input hanya sebanyak 1 piksel saja.



Gambar 2.13. Stride 2 yang bergeser secara horizontal

Pada gambar 2.13 diilustrasikan sebuah filter yang diaplikasikan pada input. Pergeseran filter ini ditentukan oleh besarnya ukuran stride. Nilai stride pada gambar 2.13 adalah 2, maka pergeseran kernel pada input sebanyak 2 piksel.

#### 2.2.5.4 Fungsi Aktivasi

Fungsi aktivasi dihitung setelah operasi konvolusi. Fungsi aktivasi yang sering digunakan dalam jaringan saraf convolutional termasuk  $\tanh()$ , ReLU (*Rectified Linear Unit*), sigmoid, dan softmax. Softmax merupakan fungsi aktivasi yang digunakan pada layer output. Layer output memiliki banyak kesamaan dengan *fully-connected layer*, yang membedakan kedua *layer* ini adalah penggunaan fungsi aktivasi softmax pada layer output dan fungsi aktivasi ReLU pada *fully-connected layer* (Achmad dkk., 2019).

Bentuk fungsi ReLU :

$$f(x) = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases} \quad (6)$$

Berdasarkan fungsi dari ReLU, dapat diketahui bahwa nilai output dari nilai neuron akan bernilai  $x$  jika nilai inputnya lebih dari atau sama dengan 0. Dengan kata lain jika nilai input aktivasi bernilai lebih dari atau sama dengan 0, maka output dari neuron adalah nilai input aktivasi itu sendiri. Sebaliknya, jika nilai output dari nilai neuron dinyatakan sebagai 0 jika inputnya bernilai kurang dari 0, dalam hal ini bernilai negatif.

Fungsi softmax menghitung probabilitas dari setiap kelas target atas semua kelas target yang memungkinkan dan akan membantu untuk menentukan kelas target untuk input yang diberikan (Sofia, 2018).

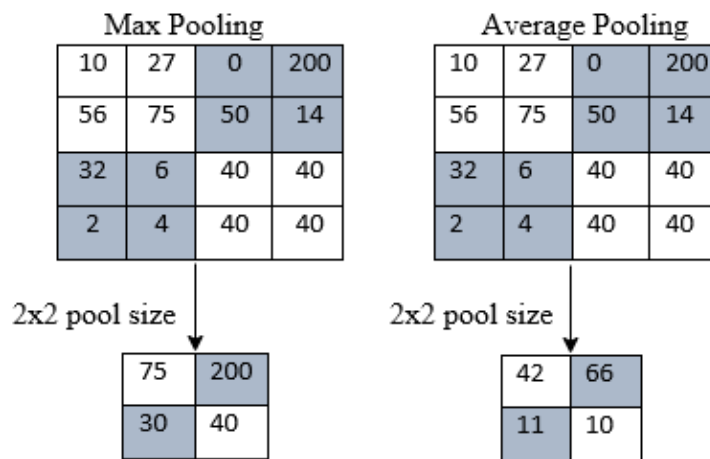
Bentuk fungsi Softmax :

$$S(x_i) = \frac{e^{x_i}}{\sum_{j=1}^n e^{x_j}} \quad (7)$$

Berdasarkan fungsi dari softmax, dapat diketahui bahwa setiap probabilitas dalam hasil berada dalam rentang 0...1, dan jumlah probabilitasnya adalah 1.

**2.2.5.5 Pooling Layer**

Pooling layer adalah lapisan yang berfungsi untuk mengurangi ukuran spasial dari fitur konvolusi sehingga dapat mengurangi sumber daya komputasi yang dibutuhkan untuk memproses data melalui pengurangan dimensi dari *feature map* (*downsampling*) sehingga mempercepat komputasi karena paramter yang diperbarui semakin sedikit. Selain itu, berguna untuk mengekstraksi fitur dominan sehingga proses pelatihan model lebih efektif. Ada dua jenis pooling layer, yaitu *max pooling* dan *average pooling*. *Max pooling* mengembalikan nilai maksimum dari bagian gambar yang dicakup oleh kernel sedangkan *average pooling* mengembalikan nilai rata – rata dari bagian gambar yang dicakup oleh kernel. Bentuk lapisan pooling yang paling umum adalah dengan menggunakan filter berukuran  $2 \times 2$  yang diaplikasikan dengan langkah sebanyak 2 dan kemudian beroperasi pada setiap irisan dari input. Contoh operasi max pooling dan average pooling ditunjukkan dalam gambar 2.14.



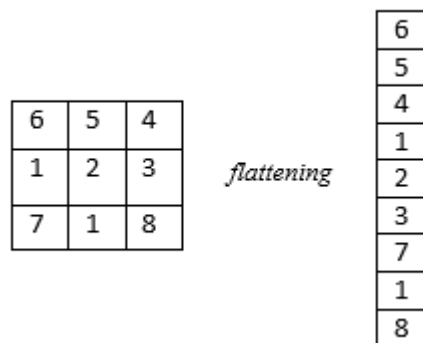
Gambar 2.14. Contoh operasi max pooling dan average pooling

**2.2.5.6 Flatten**

Flattening merupakan operasi yang mengubah matriks menjadi vektor satu dimensi. Proses flattening mengubah feature map yang telah diperoleh dari layer sebelumnya menjadi vektor satu dimensi agar feature map tersebut dapat

diklasifikasikan dengan fully-connected layer dan softmax (Achmad dkk., 2019). Proses flatten diilustrasikan pada Gambar 2.15.

Pada tahap backpropagation layer ini mengubah vektor satu dimensi kembali menjadi matriks dengan dimensi seperti semula untuk selanjutnya dapat dilakukan proses perubahan bobot filter.



Gambar 2.15. Proses flattening

### 2.2.5.7 Fully Connected Layer

Layer tersebut adalah layer yang biasanya digunakan dalam penerapan MLP dan bertujuan untuk melakukan transformasi pada dimensi data agar data dapat diklasifikasikan secara linear. Setiap neuron pada convolution layer perlu ditransformasi menjadi data satu dimensi terlebih dahulu sebelum dapat dimasukkan ke dalam sebuah fully connected layer. Karena hal tersebut menyebabkan data kehilangan informasi spasialnya dan tidak reversibel, fully connected layer hanya dapat diimplementasikan di akhir jaringan. Convolution layer dengan ukuran kernel  $1 \times 1$  melakukan fungsi yang sama dengan sebuah *fully connected layer* namun dengan tetap mempertahankan karakter spasial dari data. Hal tersebut membuat penggunaan *fully connected layer* pada CNN sekarang tidak banyak dipakai.

### 2.2.6 Dropout

Dropout adalah teknik regularisasi jaringan syaraf dimana beberapa neuron akan dipilih secara acak dan tidak dipakai selama pelatihan. Dropout merupakan proses mencegah terjadinya overfitting dan juga mempercepat proses learning. Dropout mengacu kepada menghilangkan neuron yang berupa hidden maupun *layer*

yang berupa *visible* di dalam jaringan. Dengan menghilangkan satu neuron, berarti menghilangkannya sementara dari jaringan yang ada. Neuron yang akan dihilangkan akan dipilih secara acak. Setiap neuron akan diberikan probabilitas yang bernilai antara 0 dan 1 (Lina, 2019).

### **2.2.7 Tensorflow**

*Framework Tensorflow* dikembangkan oleh *Google Brain Team* pada tahun 2015 untuk perhitungan numerik, dan sekarang banyak digunakan oleh perusahaan besar untuk pengembangan aplikasi AI, seperti klasifikasi citra, penyematan kata, dan pengembangan *chatbot*. *Tensorflow* menyediakan antarmuka yang dapat mengekspresikan algoritma *Machine Learning* dan aplikasi untuk mengeksekusi algoritma. *Tensorflow* mendukung pemodelan *Recurrent Neural Network (RNN)*, *Restricted Boltzmann Machine (RBM)*, *Convolutional Neural Network (CNN)* dan *Dynamic Bayesian Network (DBN)*, serta eksekusi paralel. Nama *Tensorflow* didasarkan pada bagaimana tensor di seluruh jaringan. Tensor merupakan jenis array multidimensi (Wiranda dkk., 2020).

#### **2.2.7.1 Tensorflow lite**

*Tensorflow Lite* adalah seperangkat alat untuk membantu pengembang menjalankan model *Tensorflow* pada perangkat *mobile*, *embedded*, dan *IoT*. Inferensi pembelajaran mesin pada perangkat dengan latensi rendah dan ukuran biner kecil. *Tensorflow Lite* mendukung *platform* Android / *iOS* serta *platform* Linux (misalnya Raspberry Pi). Pada perangkat seperti Raspberry Pi, API Python membantu. Platform *Tensorflow Lite* juga mendukung model Core ML serta platform *iOS*. Untuk penggunaan *Tensorflow Lite* dapat menggunakan model yang sudah ada atau menyiapkan model sendiri dan melatih model tersebut. Setelah model sudah siap dan harus dikonversi ke dalam *Tensorflow Lite* format menggunakan konverter. Mengkonversi model ML menjadi model *Tensorflow Lite* dapat dilakukan hanya dalam satu baris kode dengan memanggil metode konversi (Louis, dkk., 2019). Berikut adalah perintah Python sederhana yang mengubah model yang ada menjadi format *Tensorflow Lite*. (tflite).

### 2.2.8 Batch Size and Epoch

Batch Size adalah jumlah sampel data yang disebarikan ke *Neural Network*. Sedangkan Epoch adalah ketika seluruh dataset sudah melalui proses training pada *Neural Network* sampai dikembalikan ke awal untuk sekali putaran, karena satu *Epoch* terlalu besar untuk dimasukkan (*feeding*) kedalam komputer maka dari itu kita perlu membaginya kedalam satuan kecil (*batches*) (Imam, 2018).

### 2.2.9 Categorical Crossentropy

*Categorical crossentropy* adalah *loss function* yang digunakan dalam multiclass classification. Dimana sebuah sampel hanya bisa menjadi salah satu dari banyak kategori yang mungkin. *Loss crossentropy* digunakan saat menyesuaikan bobot model selama *training*. Tujuannya adalah untuk meminimalkan kerugian, yaitu semakin kecil kerugian semakin baik modelnya. Model yang sempurna yaitu memiliki nilai *loss crossentropy* sebesar 0.

*Categorical crossentropy loss function* dapat dihitung dengan menggunakan persamaan :

$$Loss = - \sum_{i=1}^{n_o} y_i \cdot \log s_i \quad (8)$$

Dimana  $s_i$  adalah prediksi peluang pada kelas ke  $i$ , sedangkan  $y_i$  adalah nilai target yang sesuai, dan  $n_o$  adalah jumlah node pada output layer.

### 2.2.10 Inception-Resnet

Sejak kompetisi ImageNet 2012 memenangkan entri oleh Krizhevsky dkk, jaringan "AlexNet" telah berhasil diterapkan ke berbagai tugas visi komputer yang lebih besar, misalnya untuk deteksi objek, segmentasi, estimasi pose manusia, klasifikasi video, pelacakan objek, dan super resolusi (Szegedy dkk., 2016).

Dalam karya ini mempelajari kombinasi dari dua ide terbaru yaitu koneksi residual yang diperkenalkan oleh He dkk., dan versi revisi terbaru dari arsitektur Inception. Dalam membandingkan dua varian Inception antara *Inception-V3* dan *V4*, dengan versi *hybrid Inception-ResNet*. Kendala utamanya bahwa parameter dan kompleksitas komputasi model harus agak mirip dengan model *non-residual*. Dapat

dikatakan bahwa *Inception ResNet-V2* memiliki performa yang sama baiknya, melebihi performa single frame yang canggih pada dataset validasi ImageNet.

Versi residual Inception hanya dua di antaranya yang dirinci di sini. Yang pertama "*Inception-ResNet-V1*" kira-kira biaya komputasi *Inception-V3*, sementara "*Inception-ResNet-V2*" cocok dengan biaya mentah dari arsitektur Inception-V4 yang baru diperkenalkan. Lihat gambar 2.16 Skema untuk arsitektur *Inception ResNet-V2*.



Gambar 2.16. Skema untuk arsitektur *Inception ResNet-V2*.

( sumber : [arxiv.org](http://arxiv.org) )

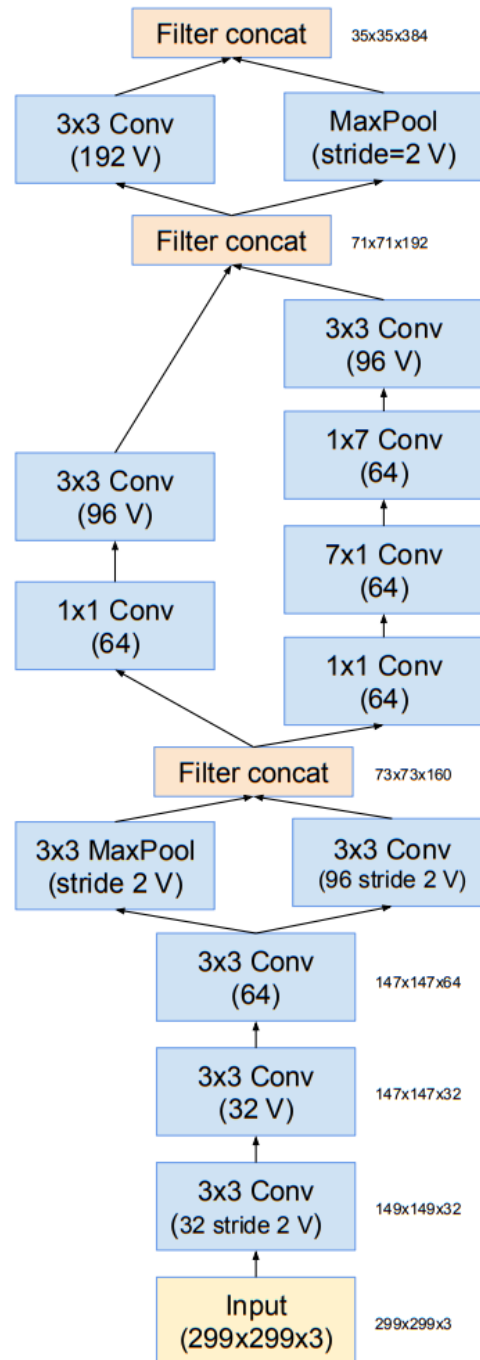


Pada tabel 2.1 merupakan arsitektur rinci dari *Inception ResNet-V2* yang dikelompokkan berdasarkan layer. Jumlah *channel input* pada masing-masing *layer* ditentukan oleh jumlah kernel pada *layer* sebelumnya yang dimana sudah ditentukan oleh arsitektur *Inception ResNet-V2* sendiri.

Table 2.1 Arsitektur *Inception ResNet-V2*

<i>Type</i>	<i>Kernel size/stride Or remarks</i>	<i>Input Size</i>
<i>Stem</i>		299×299×3
<i>5× Inception-resnet A</i>		35×35×256
<i>Reduction-A</i>		35×35×256
<i>10× Inception-resnet-B</i>		17×17×896
<i>Reduction-B</i>		17×17×896
<i>5× Inception-resnet-C</i>		8×8×1792
<i>Average Pooling</i>	8×8	8×8×1792
<i>Dropout (0,8)</i>		1×1×1792
<i>Softmax</i>		1×1×1000

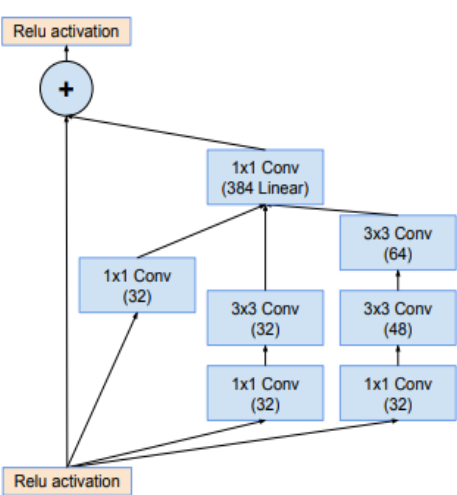
Gambar 2.16 menunjukkan arsitektur lengkap dari *Inception ResNet-V2* yang dimana diawali dari input ke stem. Stem, dalam hal ini mengacu pada operasi initial set yang dilakukan sebelum memperkenalkan blok-blok *Inception ResNet-V2*.



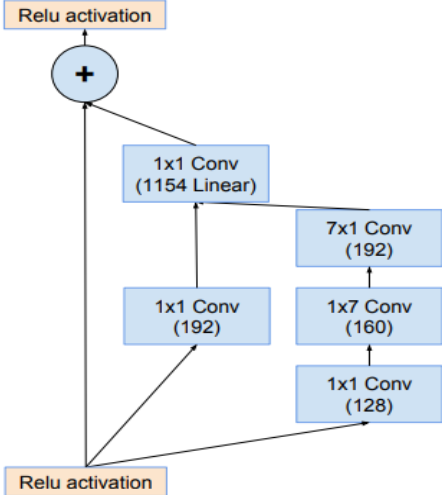
Gambar 2.17. Proses stem Inception ResNet-V2

Pada gambar 2.17, convolution yang ditandai dengan “V” menggunakan same padding sehingga dimensi output akan sama dengan dimensi input. Sebaliknya, convolution yang tidak ditandai dengan “V” menggunakan valid padding.

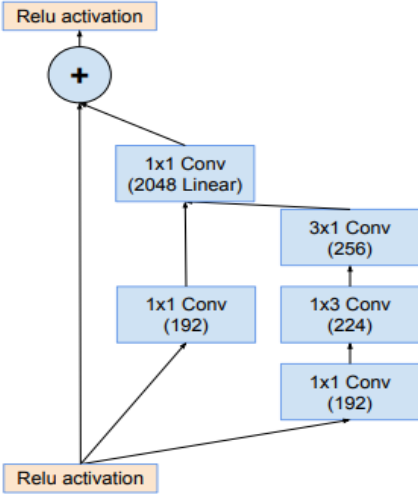
*Inception ResNet-V2* dibagi menjadi 3 modul, yaitu modul A, B, dan C yang terdapat pada gambar 2.18, 2.19, dan 2.20. modul-modul ini terlihat mirip dengan *Inception ResNet-V1*.



Gambar 2.18. Modul A *Inception Resnet-V2*

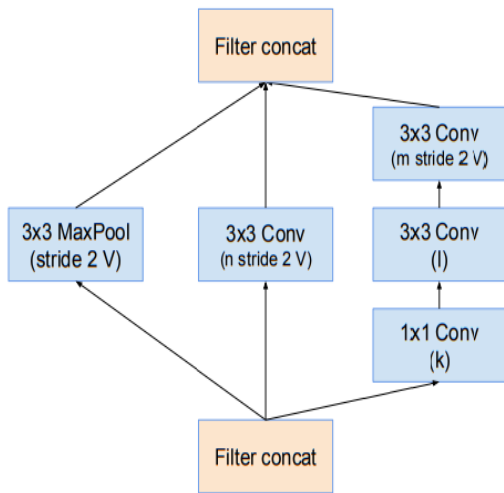


Gambar 2.19. Modul B *Inception Resnet-V2*

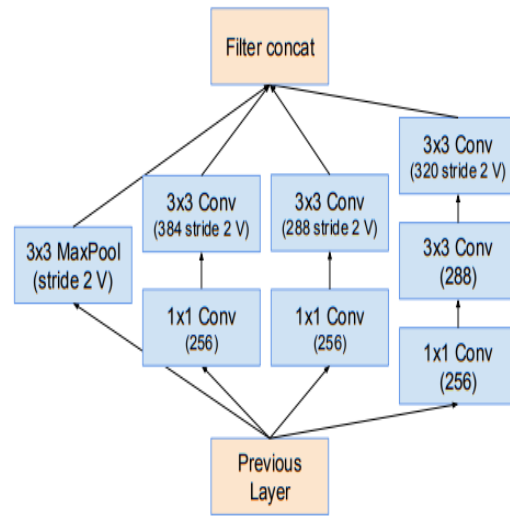


Gambar 2.20. Modul C *Inception Resnet-V2*

*Inception ResNet-V2* memperkenalkan *reduction blocks* khusus yang digunakan untuk mengubah lebar dan tinggi grid. Versi sebelumnya tidak secara eksplisit memiliki *reduction blocks*, tetapi fungsionalitas ini juga diterapkan. *Reduction blocks* bisa kita lihat pada gambar 2.21 dan 2.22.



Gambar 2.21. Reduction block A Inception Resnet-V2



Gambar 2.22. Reduction block B Inception Resnet-V2

### 2.2.11 Confusion Matrix

*Confusion matrix* Merupakan tabel yang menggambarkan performa dari sebuah model atau algoritma secara spesifik. Setiap baris dari matrix tersebut, merepresentasikan kelas aktual dari data, dan setiap kolom merepresentasikan kelas prediksi dari data (atau sebaliknya). Pada confusion matrix, terdapat 4 istilah representasi dari hasil proses klasifikasi. Empat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN).

Kelas klasifikasi positif yang diprediksi benar disebut *true positive*, kelas klasifikasi positif yang diprediksi negatif disebut *false negative* (salah). Kelas negatif yang diprediksi benar disebut *true negative*, dan kelas negatif yang diprediksi salah disebut *false negative*.

*Confusion matrix* dapat digunakan dalam menghitung berbagai performance metrics untuk mengukur kinerja model yang telah dibuat. Performance metrics yang sering digunakan adalah *accuracy*, *precision*, *recall*, dan *F1-Score*.

*Accuracy* menggambarkan seberapa akurat model dapat mengklasifikasikan dengan benar. Accuracy merupakan rasio prediksi benar dengan keseluruhan data. Dengan kata lain, accuracy merupakan tingkat kedekatan nilai prediksi dengan nilai *actual* (nilai sebenarnya).

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (9)$$

*Precision* menggambarkan tingkat kebenaran suatu pengklasifikasi dari seluruh kelas positif yang diprediksi. Maka, *precision* merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif.

$$precision = \frac{TP}{TP + FP} \quad (10)$$

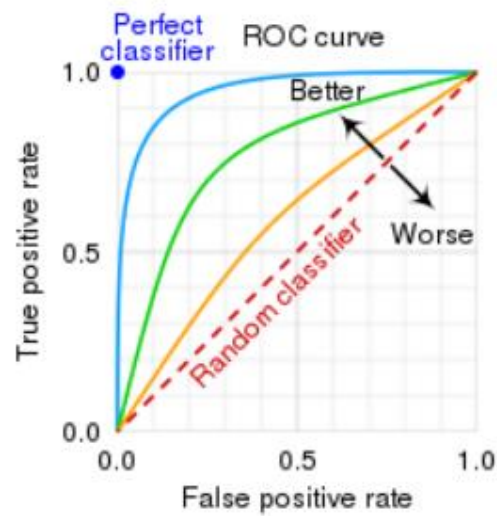
Recall (*Sensitivity*) menggambarkan seberapa baik suatu pengklasifikasi dalam mengklasifikasikan kelas positif. Recall merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan data yang benar positif.

$$recall = \frac{TP}{TP + FN} \quad (11)$$

F1-score merupakan gabungan antara *precision* dan *recall* yang bertujuan untuk melihat seberapa baik performa model dalam memprediksi suatu kelas.

$$F1-Score = 2 \times \frac{(Precision \times Recall)}{(Precision + Recall)} \quad (12)$$

Kurva ROC (Receiver Operating Characteristic) adalah kurva yang menyajikan ilustrasi performansi dari *binary classifier system* dalam menghasilkan sebuah prediksi. Kurva ROC didapatkan dengan membuat *plotting* dari *true positive rate* (TPR) terhadap *false positive rate* (FPR).



Gambar 2.23. Kurva ROC

Kurva ROC menunjukkan visualisasi antara *true positive rate* (TPR) dan false positive rate (FPR). *Classifier* yang memberikan kurva semakin mendekati ke sudut kiri atas (*perfect classifier*) menunjukkan kinerja yang semakin baik. Selanjutnya, sebagai dasar evaluasi kinerja model, dibuat *classifier* acak (*random classifier*) yang memberikan titik-titik yang terletak di sepanjang diagonal (FPR = TPR).

## BAB III METODE PENELITIAN

### 3.1 Waktu dan Tempat

Penelitian ini dilaksanakan dari bulan Juli 2022 sampai Oktober 2022. Lokasi penelitian dilakukan di Laboratorium Rekayasa Perangkat Lunak Program Studi Sistem Infomasi, Departemen Matematika, Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Hasanuddin.

### 3.2 Sumber Data

Data yang digunakan pada penelitian ini di ambil dari situs Kaggle: <https://www.kaggle.com/kmader/parkinsons-drawings> adalah data citra tulisan tangan, sketsa *spiral* dan *wave* (gelombang) untuk mengidentifikasi pasien penyakit parkinson dengan total gambar sebanyak 204 citra.

### 3.3 Tahapan Penelitian

#### 3.3.1 Preprocessing data

Pada tahap ini, data citra akan dipotong menjadi ukuran yang sama besar dengan tujuan agar jumlah piksel yang akan diolah menjadi sama. Pada citra kemudian diterapkan augmentasi data untuk memperbanyak variasi bentuk pada dataset.

#### 3.3.2 Pembagian data

Setelah preprocessing data, data citra kemudian dibagi menjadi dua yaitu data training dan data testing. Dari jumlah sebanyak 204 citra terbagi menjadi 70% untuk data training, dan 30% untuk data testing. 70% data training ini kemudian diproses menggunakan model klasifikasi *Convolutional Neural Network* dengan menggunakan arsitektur *Inception ResNet-V2*.

#### 3.3.3 Training Data

Pada tahap ini, data citra akan dilatih dengan menggunakan arsitektur *Inception ResNet-V2*.