PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN ARSITEKTUR GOOGLENET

SKRIPSI

JHON DAVID SIHOMBING

198160019

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2025

PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN ARSITEKTUR GOOGLENET

SKRIPSI

Diajukan Sebagai Salah Satu Syarat untuk Memperoleh Gelar Sarjana

di Fakultas Teknik

Universitas Medan Area

OLEH:

JHON DAVID SIHOMBING

198160019

PROGRAM STUDI TEKNIK INFORMATIKA

FAKULTAS TEKNIK

UNIVERSITAS MEDAN AREA

2025

LEMBAR PENGESAHAAN

Judul Skripsi	: Pengenalan Bahasa Isyarat Tangan Menggunakan Artsitektur Googlenet
Nama	: Jhon David Sihombing
Npm	: 198160019
Fakultas	: Teknik Informatika

Disetujui Oleh :

Pembimbing

Muhathir,ST., M.Kom

Diketahui :

Dekan Fakultas Teknik	Ketua Prodi. Teknik Informatika
Ir. Fatma Supriatno, S.T, MT,	**Rizki Muliono S.Kom, M.Kom**

Tanggal Lulus: 11 Maret 2025.

HALAMAN PERNYATAAN

Saya menyatakan bahwa skripsi yang saya susun, sebagai syarat memperolah gelar sarjana merupakan hasil karya tulis saya sendiri. Adapun bagian-bagian tertentu dalam penulisan skripsi ini yang saya kutip dari hasil karya orang lain telah dituliskan sumbernya secara jelas sesuai dengan norma, kaidah, dan etika penulisan ilmiah.

Saya bersedia menerima sanksi pencabutan gelar akademik yang saya peroleh dan sanksi-sanksi lainya dengan peraturan yang berlaku, apabila di kemudian hari ditemukan adanya plagiat dalam skripsi ini.

Medan, 11 Maret 2025

Jhon David Sihombing

198160019

HALAMAN PERNYATAAN PERSETUJUAN PUBLIKASI

TUGAS AKHIR/SKRIPSI/TESIS KEPENTINGAN AKADEMIS

Sebagai sivitas akademika Universitas Medan Area, saya yang bertanda tangan dibawah ini:

Nama	: Jhon David Sihombing
NPM	: 198160019
Program Studi	: Teknik Informatika
Fakultas	: Teknik
Jenis Karya	: Skripsi

Demi pengembangan ilmu pengetahuan, menyetujui untuk memberikan kepada Universitas Medan Area **Hak Bebas Royalti Noneksklusif** (***Non-exlusive RoyaltyFree Right***) atas karya ilmiah saya yang berjudul:

PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN ARSITEKTUR GOOGLENET.

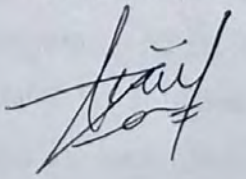
Beserta perangkat yang ada (jika diperlukan). Dengan Hak Bebas Royalti Noneklusif ini Universitas Medan Area berhak menyimpan, mengalih media/format-kan, mengelola dalam bentuk pengkalan data (Databases), merawat, dan mempublikasikan skripsi saya selama tetap mencantumkan nama saya sebagai penulis/pencipta dan sebagai pemilik Hak Cipta.

Demikan peryataan ini saya buat dengan sebenarnya.

Dibuat di Medan

Pada tanggal, 11 Maret 2025

Yang Menyatakan



(Jhon David Sihombing)

KATA PENGANTAR

Puji syukur penulis panjatkan kepada Tuhan Yang Maha Esa yang selalu menyertai kita semua, sehingga penulis dapat menyelesaikan skripsi dengan judul "**PENGENALAN BAHASA ISYARAT TANGAN MENGGUNAKAN ARSITEKTUR GOOGLENET**" sebagai salah satu syarat yang dilakukan untuk menyelesaikan program sarjana (S1) pada program sarjana Fakultas Teknik dan Jurusan Teknik Informatika di Universitas Medan Area.

Dengan kesempatan ini, penulis mengucapkan terimakasih terhadap pihak-pihak yang telah memberi banyak dukungan serta arahan sehingga penulis bisa menyelesaikan penelitian dengan baik, dan untuk itu penulis menyampaikan terimakasih kepada:

1. Tuhan Yang Maha Esa, karena berkat dan karunia-Nya penulis bisa menyelesaikan penelitian dan skripsi dengan baik.
2. Bapak Prof. Dr. Dadan Ramdan, M.Eng, M.Sc, selaku Rektor Universitas Medan Area.
3. Bapak Dr. Eng., Supriatno, S.T, M.T selaku Dekan Fakultas Teknik Universitas Medan Area.
4. Bapak Rizki Muliono S.Kom, M.Kom. selaku Ketua Kaprodi Informatika Universitas Medan Area.
5. Bapak Muhathir, S.T., M.Kom selaku Dosen Pembimbing saya yang telah bersedia meluangkan waktu untuk memberikan arahan dan bimbingan selama penyusunan skripsi saya ini.
6. Seluruh jajaran dosen dan staff pada Fakultas Teknik Universitas Medan Area.
7. Orang tua penulis, Alm. Bapak Limmer Sihombing dan Ibu Agusnita Pasaribu, abang saya Yunus Prandika Sihombing, serta adik-adik saya yang telah memberikan doa serta dukungan selama penyusunan skripsi ini.
8. Orang tua penulis, Bapak J. Purba dan Ibu S. Br. Sidabutar, serta adik saya Jesika Purba yang senantiasa memperhatikan dan mendoakan penulis.
9. Keluarga penulis, Op. Br. Panjaitan, Tulang E. Pasaribu, Tulang Pdt. J. Pasaribu, dan seluruh keluarga yang mendukung dan mendoakan penulis.

10. Adik saya, Ofi Panjaitan, S.Pd., yang banyak menolong dan senantiasa memberikan semangat dalam penyelesaian skripsi ini.
11. Teman-teman sepelayanan di Gereja Pentakosta Indonesia sidang Tobanauli yang telah memberikan kesan positif serta semangat kepada penulis.

Penulis dengan penuh kesadaran mengakui bahwa penelitian ini belum terlepas dari kekurangan tertentu. Oleh karena itu, penulis dengan rendah hati memohon maaf atas segala kesalahan yang mungkin terdapat dalam penelitian ini. Penulis sangat menghargai kritik dan saran yang konstruktif dari pembaca, karena hal tersebut akan membantu penulis untuk meningkatkan kualitas penelitian di masa mendatang.

Medan, 11 Maret 2025

Penulis

Jhon David Sihombing

198160019

ABSTRAK

Dalam beberapa tahun terakhir, penggunaan model pembelajaran mendalam, khususnya *convolutional neural networks (CNN)*, telah menjadi pendekatan dominan dalam tugas klasifikasi citra, termasuk pengenalan bahasa isyarat tangan. Pemilihan *optimizer* sangat memengaruhi performa model, namun masih sedikit studi yang memberikan evaluasi komparatif secara mendalam terhadap berbagai *optimizer* dengan menggunakan metrik yang kuat seperti *F1-score* pada arsitektur kompleks seperti *GoogleNet*. Penelitian ini bertujuan untuk mengevaluasi dan membandingkan kinerja berbagai algoritma optimisasi pada arsitektur *GoogleNet* untuk tugas klasifikasi bahasa isyarat tangan, dengan menggunakan *F1-score* sebagai metrik evaluasi utama. Pendekatan eksperimental kuantitatif digunakan dalam studi ini. Model *GoogleNet* dilatih menggunakan *dataset* citra bahasa isyarat tangan dengan tujuh *optimizer* yang berbeda: *RMSprop, Adamax, Nadam, Adadelta, Adam, Adagrad,* dan *SGD*. Semua model dilatih dalam kondisi yang identik, dan performanya dievaluasi menggunakan *F1-score* untuk mengukur keseimbangan antara presisi dan *recall*. *RMSprop* mencapai Akurasi 0.9917, Presisi 0.9917, *Recall* 0.9917 dan *F1-score* tertinggi sebesar 0.9917, menunjukkan adaptabilitas dan efisiensi yang unggul. Hasil penelitian ini menegaskan bahwa pemilihan *optimizer* sangat memengaruhi performa klasifikasi pada arsitektur *CNN Googlenet* yang kompleks. *RMSprop* terbukti paling efektif untuk tugas klasifikasi bahasa isyarat tangan. Studi selanjutnya disarankan untuk mengeksplorasi teknik optimisasi hibrida atau metode penjadwalan *learning rate* guna meningkatkan performa, khususnya pada *dataset* yang lebih besar dan beragam.

Kata Kunci: Bahasa Isyarat Tangan, *Googlenet*, *Optimizer,* Klasifikasi, *CNN*

ABSTRACT

In recent years, the use of deep learning models, especially convolutional neural networks (CNN), has become the dominant approach in image classification tasks, including hand sign language recognition. The choice of optimizer greatly affects model performance, yet there are few studies that provide a deep comparative evaluation of various optimizers using strong metrics such as the F1-score on complex architectures like GoogleNet. This research aimed to evaluate and compare the performance of various optimization algorithms on the GoogleNet architecture for the hand sign language classification task, using the F1-score as the primary evaluation metric. A quantitative experimental approach was used in this study. The GoogleNet model was trained using a hand sign image dataset with seven different optimizers: RMSprop, Adamax, Nadam, Adadelta, Adam, Adagrad, and SGD. All models were trained under identical conditions and their performance was evaluated using the F1-score to measure the balance between precision and recall. RMSprop achieved an Accuracy of 0.9917, Precision of 0.9917, Recall of 0.9917, and the highest F1-score of 0.9917, indicating superior adaptability and efficiency. The research confirmed that the choice of optimizer significantly affected classification performance on the complex CNN GoogleNet architecture. RMSprop proved to be the most effective for hand sign language classification. Future studies are suggested to explore hybrid optimization techniques or learning rate scheduling methods to improve performance, particularly on larger and more diverse datasets.

***Keywords**: Hand Sign Language, GoogleNet, Optimizer, Classification, CNN*

RIWAYAT HIDUP

Penulis lahir di Desa Medan Estate pada tanggal 29 Mei 2001 dari Ayah Alm Limmer Sihombing dan Ibu Agusnita br Pasaribu. Penulis adalah anak kedua dari 5 (lima) bersaudara.

Penulis pertama kali mengenyam pendidikan di bangku Sekolah Dasar Negeri 105293 Medan pada tahun 2006-2012, meneruskan pendidikan di Sekolah Menengah Pertama Negeri 29 Medan diselesaikan pada tahun 2012-2015, kemudian meneruskan pendidikan di Sekolah Menengah Kejuruan Swasta Teladan Medan pada tahun 2015-2018.

Pada tahun 2018, penulis lulus dari SMK Swasta Teladan Medan dan pada 2019 terdaftar sebagai mahasiswa Fakultas Teknik Prodi Teknik Informatika Universitas Medan Area. Penulis juga pernah mengikuti program Pertukaran Mahasiswa Merdeka selama satu semester di Institut Teknologi Kalimantan. Pada saat ini tahun 2025 penulis dinyatakan lulus dalam ujian akhir Sidang Skripsi.

DAFTAR ISI

LEMBAR PENGESAHAN ..... **i**
HALAMAN PERNYATAAN ..... **ii**
KATA PENGANTAR ..... **iv**
ABSTRAK ..... **vi**
RIWAYAT HIDUP ..... **viii**
DAFTAR ISI ..... **ix**
DAFTAR GAMBAR ..... **xi**
DAFTAR TABEL ..... **xii**
DAFTAR LAMPIRAN..... **xiii**

BAB 1 PENDAHULUAN **1**
1.1 Latar Belakang 1
1.2 Rumusan Masalah 3
1.3 Batasan Masalah 4
1.4 Tujuan Penelitian 6
1.5 Manfaat Penelitian 6
1.6 Sistematika Penulisan 6

BAB II LANDASAN TEORI **8**
2.1 Klasifikasi 8
2.2 *Deep Learning* 10
2.3 *Convolutional Neural Network* (CNN) 12
2.4 Arsitektur *Googlenet* 18
2.5 *Optimizer Adam* 21
2.6 *Optimizer Adamax* 22
2.7 *Optimizer AdaGrad* 23
2.8 *Optimizer AdaDelta* 24
2.9 *Optimizer Nadam* 25
2.10 *Optimizer SGD* 26
2.11 *Optimizer RMSprop* 28
2.12 *Confusion Matrix* 29
2.13 Penelitian Terdahulu 31

BAB III METODOLOGI PENELITIAN **32**
3.1 Prosedur Penelitian 32
3.2. Analisis Data 33
3.3 Perancangan Metode CNN 35
3.4 Pemodelan Arsitektur *Googlenet* 35
3.5 Arsitektur *Googlenet* 35
3.6 Uji Coba *Optimizer* 37
3.7 Confusion Matrix 38

3.8 Hasil Klasifikasi 39
3.9 Alat dan Bahan Penelitian 39

BAB IV HASIL DAN PEMBAHASAN 41
4.1 Hasil Implementasi 41
4.2 Pembahasan 65

BAB V KESIMPULAN DAN SARAN 68
5.1 Kesimpulan 68
5.2 Saran 68

DAFTAR PUSTAKA 70
LAMPIRAN 73

DAFTAR GAMBAR

Gambar 2.1 Bahasa Isyarat Tangan **9**
Gambar 2.2 Pengenalan *Deep Learning* **11**
Gambar 2.3 Struktur Dasar CNN..... **12**
Gambar 2.4 Arsitektur *GoogleNet* **18**
Gambar 2.5 Metode *Confusion Matrix* **29**
Gambar 3.1 Alur Diagram Penelitian..... **32**
Gambar 3.2 Jaringan Arsitektur CNN..... **35**
Gambar 3.3 Perhitungan *Confusion Matrix Presisi, Recal, F1-Score* **38**
Gambar 4.1 Hasil Visualisasi *Googlenet* dengan *Optimizer Adam* **44**
Gambar 4.2 Hasil *Confusion Matrix* dengan *Optimizer Adam* **45**
Gambar 4.3 Hasil Visualisasi *Googlenet* dengan *Optimizer Adamax*..... **47**
Gambar 4.4 Hasil *Confusion Matrix* dengan *Optimizer Adamax* **48**
Gambar 4.5 Visualisasi Arsitektur *Googlenet* dengan *Optimizer Nadam*..... **50**
Gambar 4.6 Hasil *Confusion Matrix Optimizer Nadam* **51**
Gambar 4.7 Visualisasi Arsitektur *Googlenet* dengan *Optimizer Adadelta*.... **53**
Gambar 4.8 Hasil *Confusion Matrix Optimizer Adadelta* **54**
Gambar 4.9 Visualisasi Arsitektur *Googlenet* dengan *Optimizer Adagrad* **56**
Gambar 4.10 Hasil *Confusion Matrix Optimizer Adagrad*..... **57**
Gambar 4.11 Visualisasi Arsitektur SGD..... **59**
Gambar 4.12 Hasil *Confusion Matrix Optimizer SGD*……..……..**60**
Gambar 4.13 Visualisasi Arsitektur *Googlenet* dengan *Optimizer RMSprop* . **62**
Gambar 4.14 Hasil C*onfusion Matrix Optimizer RMSprop*..... **64**

DAFTAR TABEL

Tabel 2.1 Penelitian Terdahulu **31**
Tabel 3.1 Pembagian Dataset..... **33**
Tabel 3.2 Struktur Arsitektur *Googlenet*..... **36**
Tabel 3.3 Parameter *Googlenet* **38**
Tabel 3.4 *Hardware* yang Digunakan..... **39**
Tabel 3.5 *Software* yang Digunakan **40**
Tabel 4.1 Hasil Augmentasi Data **41**
Tabel 4.2 Clasificaion Report *Optimizer Adam* **45**
Tabel 4.3 *Clasification Report Optimizer Adamax* **48**
Tabel 4.4 *Clasification Report Optimizer Nadam* **51**
Tabel 4.5 *Clasification Report Optimizer Adadelta* **54**
Tabel 4.6 *Clasification Report Optimizer Adagrad* **57**
Tabel 4.7 *Clasification Report Optimizer SGD* **61**
Tabel 4.8 *Clasification Report Optimizer RMSprop* **64**
Tabel 4.9 Perbandingan *Optimizer* **66**
Tabel 4.6 Hasil Akurasi Penelitian Terdahulu dengan *GoogleNet* **67**

DAFTAR LAMPIRAN

Lampiran 1 Kode Program……**73**
Lampiran 2 Hasil Plagiasi……**75**
Lampiran 3 SK Pembimbing……..……..**76**
Lampiran 4 Surat Pengantar Riset……**77**
Lampiran 5 Surat Selesai Riset……**78**

BAB I

PENDAHULUAN

1.1 Latar Belakang

Teknologi kecerdasan buatan berkembang pesat dan digunakan di berbagai bidang seperti pengenalan pola dan klasifikasi gambar. Klasifikasi bahasa isyarat merupakan salah satu penerapan kecerdasan buatan yang membantu mengenali gerakan tangan sebagai salah satu jenis komunikasi nonverbal, terutama bagi penyandang tunarungu. Bahasa isyarat digunakan sebagai sarana komunikasi utama antara penderita gangguan pendengaran dengan orang disekitarnya. Di bidang teknologi, penting untuk memperkenalkan bahasa isyarat melalui gambar untuk membangun sistem yang dapat mengubah gerakan tangan menjadi informasi yang dapat dipahami masyarakat umum (Swapna dkk., 2020). Sedangkan komunitas tunarungu sangatlah penting menggunakan bahasa isyarat dalam komunikasi, pengertian masyarakat tentang pemahaman gestur dan gerakan dalam bahasa isyarat sangatlah terbatas yang menjadi ganguan saat ingin berkomunikasi menjadi hambatan dalam berkomunikasi (Yolanda dkk., 2020). *Convovulutional Neural Network* (CNN) adalah teknik pembelajaran mendalam yang sangat efektif untuk klasifikasi gambar. Arsitektur CNN yang berkinerja baik dalam klasifikasi gambar adalah *GoogLeNet* atau *Inception v3*, yang dirancang untuk meningkatkan efisiensi dan akurasi pemrosesan gambar. *GoogLeNet* telah terbukti memiliki kemampuan klasifikasi objek tingkat lanjut dengan menggunakan beberapa lapisan konvolusional dalam strukturnya, memungkinkan pengenalan pola pada gambar yang kompleks. Dalam penelitian ini, CNN dengan arsitektur *GoogLeNet*

digunakan untuk klasifikasi bahasa isyarat tangan.Dengan kemajuan serta perkembangan teknologi, ada cara untuk mempermudah hubungan komunikasi antara pengguna dan orang yang tidak bisa menggunakan bahasa isyarat melalui teknologi pengenalan isyarat tangan (Rizal dkk., 2020). Pengenalan bahasa isyarat tangan secara otomatis dapat memberikan tembusan kemajuan yang mudah dalam mengatasi hambatan komunikasi ini. dengan ada nya teknologi ini dapat digunakan dalam aplikasi penerjemah bahasa isyarat, alat bantu pendidikan, atau perangkat interaktif untuk meningkatkan mempermudah para tunarungu dalam berkomunikas begitu juga dangan orang normal dalam berkomunikasi (Dako & Ridwan, 2021).

GoogleNet pertama kali diluncurkan pada tahun yang merupakan salah satu arsitektur CNN yang dirancang dengan efesiensi dan kedalaman membuat arsitektur sangat cocok digunakan untuk klasifikasi pengenalan bahasa isyarat tangan.salah satu keunggulan utama *GoogleNet* adalah penggunaan *inception module*, yang memungkinkan model dalam mengeksekusi fitur multi skala dari bentuk dan gestur tangan dengan menentukan akurasi tinggi (Zakiah dkk., 2021).

Kemampuan ini sangat diperlukan dalam menangani variasi bentuk, ukuran, dan orientasi gestur tangan. dengan 22 lapisan, *GoogleNet* lebih efisien dalam hal jumlah parameter yang dipakai dibandingkan dengan model-model CNN lainya, disebabkan penggunaan *inception* module yang mengurangi beban komputasi parameter yang digunakan googlenet yang membantu mencengah terjadinya *overfitting* dan stabilitas serta memperbaiki akurasi yaitu *auxilivery*, terutama saat berhadapan dengan data beragam seperti bentuk objek (Thiodorus dkk., 2021). selain itu googlenet memiliki kemampuan yang signifikan dalam menggabungkan efesiensi komputasi dan akurasi serta generalisasi yang baik terhadap

gestur,gerakan dan gambar, membuat *GoogleNet* unggul dan ideal untuk aplikasi pengenalan bahasa isyarat tangan (Wicaksono, 2020). Pada penelitian sebelumnya yang menggunakan arsitektur Googlenet dengan judul penelitian "klasifikasi kualitas kelapa" mendapatkan akurasi yang cukup baik dalam mengklasifikasi citra *GoogleNet* mendapatkan hasil dengan rata-rata akurasi 84,89% pada setiap lapisan, disusul arsitektur *RestNet101* dengan akurasi rata-rata 78,41%, *RestNet50* dengan akurasi rata-rata 77,18%. (Mustamin dkk., 2021)

Namun, dalam melatih CNN, peran *optimizer* sangat penting dalam menentukan seberapa cepat dan akurat model dapat beradaptasi terhadap pola dalam data. *Optimizer* adalah metode yang digunakan untuk memperbarui bobot jaringan berdasarkan gradien error. Berbagai jenis *optimizer* memiliki karakteristik dan kecepatan konvergensi yang berbeda, yang dapat mempengaruhi hasil akhir dari pelatihan model CNN. Dalam penelitian ini, beberapa jenis *optimizer* akan diuji, yaitu *Adam, Adamax*, *Adadelta*, *Stochastic Gradient Descent* (SGD), *RMSprop, AdaGrad*, dan *Nadam.* Penggunaan berbagai *optimizer* ini bertujuan untuk menentukan jenis *optimizer* yang paling efektif dalam mengklasifikasikan bahasa isyarat tangan menggunakan arsitektur *GoogLeNet.* Maka penelitian ini Mencoba melakukan analisis klasifikasi dengan judul "Pengenalan Bahasa Isyarat Tangan Menggunakan *Googlenet*".

1.2 Rumusan Masalah

Permasalahan yang di identifikasi dalam penelitian ini, yaiutu bagaimana proses analasis yang akan dilakukan dalam pengujian dalam mengklasifikasi bahasa isyarat melalui citra gambar dengan menggunakan algoritma CNN melalui

model Arsitektur *Googlenet* dalam menentukan hasil *akurasi confusion matrix f-1score.*

1.3 Batasan Masalah

Adapun batasan masalah yang diambil dari kajian latar belakang diatas sebagai berikut ini:

1. Jangkauan Bahasa Isyarat

   Peneletian berfokus pada pengenalan bahasa isyarat tangan yang terbatas pada gerakan dan jari tanpa memperhitungkan ekspresi wajah atau posisi tubuh yang juga merupakan bagian yang penting digunakan dari komunikasi bahasa isyarat. Model yang disesuaikan hanya akan memproses gestur tangan sebagai *input* utama untuk diterjemahkan ke dalam teks.

2. Penerapan Bentuk Bahasa Isyarat

   Sistem yang dikembangkan akan dibatasi pada satu jenis bahasa, misalnya *American Sign Language* (ASL) atau *Indonesian Sign Language* (SIBI). Penegenalan bahasa isyarat lainnya akan memerlukan pengembangan dan penyesuaian pada model lebih lanjut.

3. Data dan Bentuk Gerakan

   Sistem yang akan diuji menggunakan dataset yang berisi gambar dengan variasi bentuk dan polatangan yang terbatas, contoh salah satu faktor yang dapat mempengaruhi yaitu seperti pencahayaan, sudut pandang, dan kecepatan gerakan yang tidak signifikan. Mungkin tidak diperhitungan sama dalam penelitian ini.

4. Arsitektur Jaringan yang Diuji

   Penelitian ini hanya akan berfokus dalam menguji akurasi pada arsitektur *GoogleNet* sebagai model CNN. Tidak mengekplorasi arsitektur lainnya ataupun membandingkan.

5. Bagaimana perbandingan performa berbagai jenis *optimizer (Adam, Adamax, Adadelta, SGD, RMSprop, AdaGrad, dan Nadam)* dalam proses pelatihan CNN untuk klasifikasi bahasa isyarat tangan?

6. *Optimizer* mana yang memberikan hasil terbaik dalam akurasi dan konvergensi untuk klasifikasi bahasa isyarat tangan menggunakan arsitektur *GoogleNet*?

7. Perangkat Komputer

   Penelitian ini akan dibatasi oleh ketersediaan sumber daya yang digunakan dalam pelatihan dalam pengujian model, maka dari itu peforma pada komputasi akan diuji secara *real-time* dan penggunaan yang terbatas seperti, pada perangkat *desktop* ataupun laptop, dan bukanpada perangkat seperti *smartphone*.

8. Pengukuran Keberhasilan

   Hasil daripada penelitian ini akan diukur dari tingkat akurasi yang dicapai melalui pengenalan gerakan pada gerakan dan pola pada *dataset* yang digunakan dan tidak mencakup faktor lain seperti penyederhanan daya atau kecepatan *realtime* pada aplikasi praktis yang lebih banyak.

1.4 Tujuan Penelitian

Adapun yang menjadi tujuan dalam penelitian sebagai berikut:

1. Menentukan seberapa baik *Goolgenet* dalam mengenali citra berbagai gestur tangan yang berbeda untuk menentukan hasil akurasi klasifikasi.
2. Melakukan pengujian terhadap berbagai jenis *optimizer* untuk mengetahui pengaruhnya terhadap performa klasifikasi.
3. Menentukan *optimizer* terbaik dalam proses klasifikasi bahasa isyarat tangan dengan menggunakan arsitektur *GoogleNet.*
4. Menerapkan metode *GoogleNet* dalam mengklasifikasi bahasa isyarat

1.5 Manfaat Penelitian

Adapun manfaat penelitian ini yaitu sebagai berikut :

1. Memperkaya keilmuan terkait *Deep Learning* dalam mengklasifikasi Bahasa Isyarat.
2. Menerapkan metode *googleNet* dalam mengklasifikasi Bahasa Isyarat.
3. Menghasilkan *Deep Learning* dengan metode *GoogleNet* dalam pengenalan Bahasa Isyarat.
4. Dapat menjadi referensi lain dalam melakukan penelitian lebih lanjut terkait model CNN dengan arsitektur Googlenet.

1.6 Sistematika Penulisan

Dalam penyusunan penulisan dalam penelitian ini dibentuklah sistematika penulisan yang terdiri dari beberapa bagian yang terpenting sebagai berikut.

BAB I : PENDAHULUAN

Pada bagian dari bab ini dijelaskn tentang mengenai awal dari penelitian yang disusun secara terstruktur diantaranya Latar Belakang, Tujuan Penelitian, Rumusan Masalah, Batasan Masalah, dan Manfaat Penelitian.

BAB II : TINJAUAN PUSTAKA

Dibagian bab ini dijelaskan secara terstruktur tentang teori yang ditemukan untuk sebagai acuan dalam mmperluas suatu informasi dalam melakukan klasifikasi bahasa Isyarat dengan citra rentina pada manusia menggunakan metode *GoogleNet.* Diliputi dengan dasar-dasar dari metode *Deep Learning* Arsitektur *GoogleNet.*

BAB III : METODE PENELITIAN

Pada bagian bab ini dijelaskan aluir dari metode penelitian ini yang disusun secara tersusun dan beskala. Agar mudah dipahami oleh pembaca.

BAB IV : HASIL DAN PEMBAHASAN

Pada bab ini berisi suatu hasil yang diperoleh pada saat melakukan pengujian dari metode *GoogleNet* dengan objek gerak tubuh.

BAB IV : HASIL DAN PEMBAHASAN

Bab ini berikan tentang kesimpulan dari keselurahan penelitian yang dilakukan dalam waktu pengujian dari metode *GoogleNet* menggunakan obejek gambar bentuk tangan dan bab ini juga berisi tentang saran tentang kekurangan dari penelitian ini gara untuk penelitian selanjutnya dapat melakukan suatu penelitian dengan pengembangan dari metode *GoogleNet*.

BAB II

LANDASAN TEORI

2.1 Klasifikasi

Klasifikasi citra adalah proses yang bertujuan untuk mengkategorikan objek dalam gambar berdasarkan fitur visual tertentu dengan menggunakan teknik *computer vision* dan *machine learning*. Proses ini melibatkan beberapa tahapan, seperti pengumpulan data citra, pra-pemrosesan untuk menyiapkan data, ekstraksi fitur untuk mengidentifikasi karakteristik utama, serta pelatihan dan evaluasi model (Fitrianingsih & Rodiah, 2020). Beberapa algoritma yang sering digunakan dalam klasifikasi citra meliputi *K-Nearest Neighbor* (KNN), Support Vector Machine (SVM), dan yang paling efektif, *Convolutional Neural Networks* (CNN). CNN unggul karena kemampuannya untuk mengekstraksi fitur visual secara otomatis melalui lapisan konvolusi, yang menganalisis gambar secara detail, termasuk garis tepi, bentuk, dan pola pada berbagai tingkat abstraksi. Model seperti GoogLeNet, yang berbasis CNN, memiliki kemampuan menangkap berbagai skala informasi, sehingga sangat cocok untuk tugas-tugas klasifikasi citra yang kompleks, seperti pengenalan gestur dan bentuk dalam bahasa isyarat (Rochmawanti dkk., 2021).

2.1.1 Bahasa Isyarat Tangan

Bahasa isyarat adalah sistem komunikasi yang menggunakan gerakan tangan, ekspresi wajah, dan gerakan tubuh untuk menyampaikan pesan. Bahasa ini digunakan terutama oleh komunitas tunarungu atau orang dengan gangguan pendengaran, meskipun juga dapat digunakan oleh orang lain dalam situasi tertentu. Setiap negara atau daerah biasanya memiliki bentuk bahasa isyarat yang berbeda-

beda, seperti *American Sign Language* (ASL) di Amerika Serikat atau Bahasa Isyarat Indonesia (BISINDO) di Indonesia (Nasha Hikmatia A.E. & Zul, 2021)

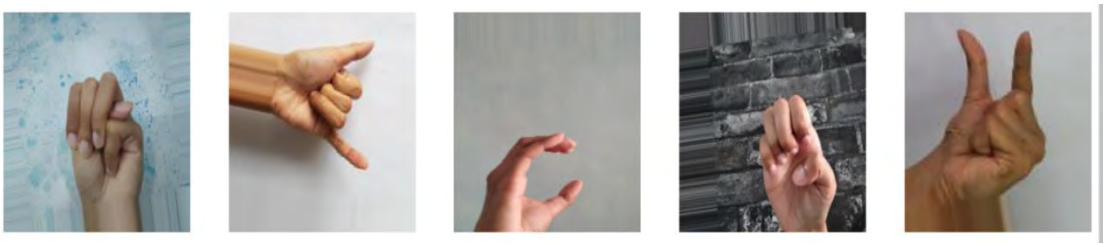
Karakteristik Utama Bahasa Isyarat:

1. Visual dan Gerakan: Bahasa isyarat bersifat visual, artinya informasi disampaikan melalui penglihatan, bukan pendengaran. Gerakan tangan, jari, dan wajah merupakan bagian penting dalam menyampaikan makna.
2. Tata Bahasa yang Unik: Bahasa isyarat memiliki tata bahasa dan aturan sintaksis yang unik, tidak hanya sekadar menerjemahkan kata dari bahasa lisan.
3. Ekspresi Wajah dan Tubuh: Ekspresi wajah, seperti mengerutkan dahi atau tersenyum, serta postur tubuh, juga memainkan peran penting dalam menekankan makna atau menyampaikan emosi.

Contoh Penggunaan:

1. Komunikasi antar komunitas tunarungu.
2. Komunikasi dalam situasi di mana suara tidak efektif, seperti di lingkungan yang sangat bising.
3. Pendidikan inklusif untuk anak-anak tunarungu atau dengan gangguan bicara.

Bahasa isyarat adalah bahasa yang lengkap dan kaya dengan struktur linguistik yang setara dengan bahasa lisan (Fauzi & Moenir, 2021).



Gambar 2.1 Bahasa Isyarat Tangan

2.2 *Deep Learning*

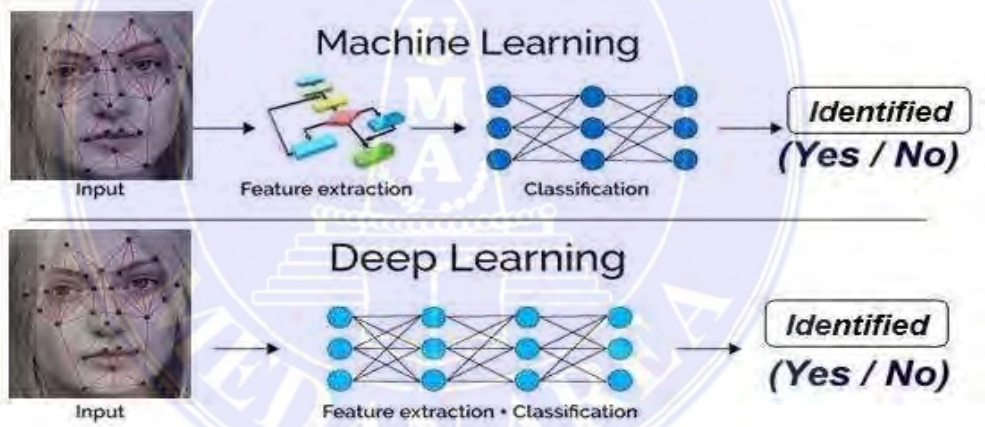
Deep Learning adalah salah satu cabang dari *machine learning* yang berfokus pada penggunaan jaringan saraf tiruan dengan banyak lapisan (dikenal sebagai *Neural networks*) untuk menganalisis berbagai jenis data. *Deep learning* memungkinkan model untuk belajar dari data yang tidak terstruktur, seperti gambar, suara, dan teks, dengan cara yang meniru cara manusia memahami informasi (F. Mustamin et al., 2021). Penggunaan *Deep Learning* digunakan dalam berbagai aplikasi, antara lain:

1. Pengolahan Citra: Klasifikasi gambar, deteksi objek, segmentasi citra, dan pengenalan wajah.
2. Pemrosesan Bahasa Alami (NLP): Terjemahan bahasa otomatis, analisis sentimen, dan *chatbots.*
3. Pengenalan Suara: Sistem pengenalan suara, asisten virtual, dan transkripsi otomatis.
4. Kendaraan Otonom: Menggunakan sensor dan citra untuk navigasi dan pengenalan rintangan.
5. Kesehatan: Diagnostik medis melalui analisis citra medis, deteksi penyakit, dan personalisasi pengobatan.

Manfaat *Deep Learning*

1. Akurasi Tinggi: Model *deep learning* seringkali menghasilkan akurasi yang lebih tinggi dibandingkan teknik machine learning tradisional, terutama dalam pengolahan data yang kompleks. Kemampuan Belajar dari Data Besar: *Deep learning* dapat menangani dan belajar dari volume data yang sangat besar, yang sulit dikelola oleh metode lain.

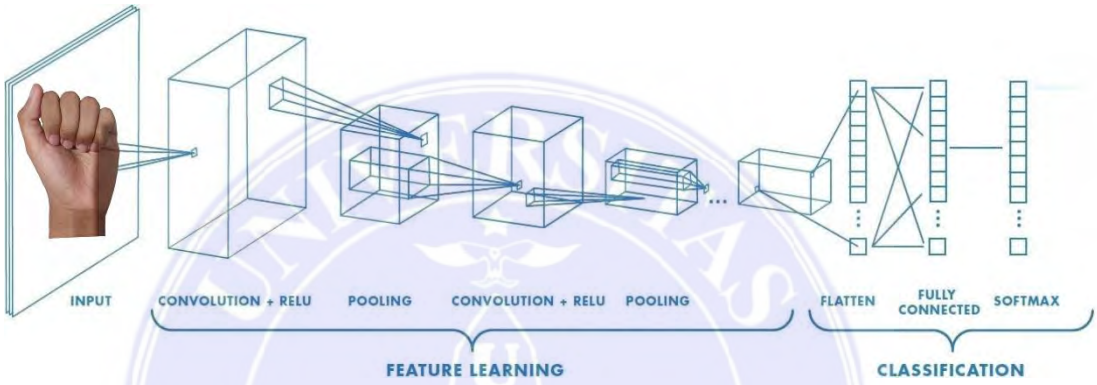
2. Otonomi dalam Pembelajaran: Model dapat belajar fitur secara otomatis dari data, mengurangi kebutuhan untuk ekstraksi fitur manual.
3. Penerapan Luas: *Deep learning* dapat diterapkan di berbagai bidang, termasuk teknologi, kesehatan, transportasi, dan hiburan.
4. Inovasi dan Penemuan Baru: Mendorong inovasi dalam pengembangan teknologi baru, seperti aplikasi AI yang lebih canggih dan otomatisasi proses.
5. Dengan kemajuan yang terus berlanjut dalam teknologi komputasi dan algoritma*, Deep learning* terus berkembang dan memainkan peran penting dalam kemajuan di berbagai industri (Queentinela & Triyani, 2021)



Gambar 2.2 Pengenalan Deep Learning

2.3 *Convolutional Neural Network* (CNN)

Convolutional neural network (CNN) adalah jenis arsitektur jaringan saraf yang sangat efektif dalam memproses data seperti *grid* seperti citra gambar. CNN dirancang untuk mengenali pola dan fitur dalam data visual, menjadikannya sangat populer untuk tugas-tugas seperti pengenalan gambar, deteksi objek, dan pengolahan video (Lestari & Aini, 2021)



Gambar 2.3 Struktur Dasar CNN

1. Struktur Dasar CNN

CNN biasanya terdiri dari beberapa lapisan yang berbeda, yang masing-masing memiliki fungsi spesifik:

a. Lapisan *Input*

Menerima *input* berupa citra. Misalnya, citra berwarna dengan dimensi 32 x 32 *piksel* akan memiliki ukuran *input* 32 x 32 x 3 (3 saluran untuk warna RGB).

Input citra berukuran

$$H \times W \times C \quad (2.1)$$

Keterangan;

H: Tinggi citra

W: Lebar citra

C: jumlah saluran (jalur RGB)

b. Lapisan Konvolusi

Menerapkan operasi konvolusi pada inIput untuk mengekstraksi fitur. Dalam lapisan ini, filter (atau kernel) bergerak di seluruh citra untuk mendeteksi pola seperti tepi, sudut, dan tekstur. Setiap filter menghasilkan peta fitur (*feature map*) yang menunjukkan lokasi dan kekuatan fitur yang terdeteksi (Hatta Fudholi et al., 2021).

$$Y(i,j,k) = \sum_{M=1}^{F_H} + \sum_{c=1}^{F_w} + \sum_{n=1}^{c} \mathrm{X}\,(i+m, j+n, c).K(m,n,c,k) + b_k$$

(2.2)

Keterangan:

X(i,j,c) : Nilai input posisi (*i,j*) dan saluran c

K(m,n,e,k): Nilai filter pada posisi (*m,n*) saluran c, dan filter ke-*k.*

b_k : Bias untuk filter *k.*

F_H, F_w : Ukuran filter (tinggi dan lebar).

C : Jumlah saluran (*channel*) pada *input*

$Y(i,j,k)$: *Output* dari konvolusi pada posisi (*i,j*) dan filter ke- *k*

c. Lapisan Aktivasi (*ReLU*)

Setelah proses konvolusi, fungsi aktivasi *ReLU* (*Rectified Linear Unit*) diterapkan untuk menambahkan *non-linearitas*. *ReLU* hanya mengijinkan nilai positif, dan mengubah semua nilai negatif menjadi nol.

$$f(x) = \max(0, x) \qquad (2.3)$$

Jika $x > 0$, maka *output* $f(x) = x$ (nilai x dipertahankan).

Jika $x \leq 0$, maka *output* $f(x) = 0$ (nilai x dipotong menjadi 0).

Keterangan:

$f(x)$: Ini adalah *output* dari fungsi aktivasi. Fungsi aktivasi digunakan untuk memperkenalkan *non-linearitas* dalam jaringan saraf. *Non-linearit*as ini memungkinkan jaringan untuk belajar berbagai pola yang kompleks.

x: Input yang diterima oleh fungsi aktivasi, biasanya merupakan hasil dari perkalian bobot dan *input*, serta penambahan bias, yaitu $W \times I + b$

$\max(0, x)$: Fungsi ini mengambil nilai maksimum antara 0 dan x, artinya:

Jika $x > 0$, maka *output* $f(x) = x$ (nilai x dipertahankan).

Jika $x \leq 0$, maka *output* $f(x) = 0$ (nilai x dipotong menjadi 0).

$x > 0$: Jika *input* x lebih besar dari 0, fungsi *ReLU* mengeluarkan *output* yang sama dengan *input* x. Dalam kasus ini, tidak ada modifikasi terhadap *input*.

$x \leq 0$: Jika *input* x lebih kecil dari atau sama dengan 0, maka *output* dari fungsi ReLU adalah 0. Ini berarti bahwa *input* negatif atau nol di-nonaktifkan, memberikan keluaran nol.

d. Lapisan *Pooling*

Mengurangi dimensi peta fitur dengan mengambil nilai maksimum (*Max pooling*) atau rata-rata (*Average pooling*) dalam suatu area tertentu. Ini membantu mengurangi jumlah parameter, mencegah *overfitting*, dan meningkatkan efisiensi komputasi.

$$O = \frac{I-F}{S} + 1 \qquad (2.4)$$

Keterangan:

O: Ukuran *output* dari *layer* konvolusi, biasanya dalam bentuk panjang atau lebar (*height/width*) dari *feature map* hasil setelah filter diterapkan. Nilai ini menunjukkan berapa banyak *neuron* atau piksel yang ada di *layer* hasil (*output*).

I: Ukuran dari *input*, yang biasanya adalah dimensi panjang atau lebar dari gambar *input* atau *layer* sebelumnya. Jika *input* adalah gambar, ini akan merujuk pada ukuran *height* atau *width* dari gambar tersebut.

F: Ukuran filter atau kernel yang digunakan dalam proses konvolusi. Filter ini adalah matriks kecil yang "digeser" melintasi gambar untuk mengekstrak fitur. Contoh filter berukuran 3×3 $berarti\ F = 3$

S: Langkah perpindahan filter saat melakukan konvolusi. *Stride* menentukan seberapa jauh filter akan bergerak setiap kali dalam proses konvolusi. Jika *stride* bernilai 1, filter akan bergeser satu piksel setiap kali. Jika *stride* bernilai 2, filter akan melompati satu piksel setiap kali.

$+1$: Bagian tambahan dalam rumus ini digunakan untuk memperhitungkan elemen *output* pertama dari konvolusi, karena meskipun filter di-*shift,* output harus dimulai dari indeks 1, bukan 0.

e. Lapisan *Fully Connected* (FC)

Setelah beberapa lapisan konvolusi dan *pooling, lapisan fully connected* menghubungkan semua neuron dari lapisan sebelumnya ke neuron di lapisan *output.* Ini berfungsi untuk menghasilkan prediksi akhir berdasarkan fitur yang telah diekstrak.

$$y_i = \sum_j w_{i,j} . x_j + b_i \quad (2.5)$$

Keterangan:

$w_{i,j}$: Bobot antara neuron *j* di lapisan sebelumnya dengan neuron *i* di lapisan *fully connected.*

x_j: Output dari neuron *j* di lapisan sebelumnya.

b_i: Bias neuron *i*.

y_i: *Output* dari neuron *i* di lapisan *fully connected.*

f. Lapisan *Output* dan *Softmax*

Memberikan hasil akhir, biasanya berupa probabilitas untuk setiap kelas dalam tugas klasifikasi, menggunakan fungsi aktivasi seperti *softmax.*

$$P(y = k) = \frac{e^{zk}}{\sum_{n=1}^{k} e^{zj}} \tag{2.6}$$

Keterangan:

Z_k: skor (logit) dari kelas *k.*

K: jumlah kelas

P: $(y = k)$*:* probabilitas bahwa *input* milik kelas K.

g. *Loss Function* (Fungsi Kerugian - *Cross-Entropy*)

Untuk mengukur seberapa baik model memprediksi, CNN biasanya menggunakan *cross-entropy loss* untuk klasifikasi. *Cross-entropy loss* menghitung perbedaan antara prediksi dan label sebenarnya.

$$L = \sum_{i=1}^{K} y_i \log(y_{i;}) \tag{2.7}$$

Keterangan:

y_i: label sebenarnya (1 jika benar, 0 jika salah).

$\hat{y}$: prediksi probabilitas untuk kelas *i.*

K: jumlah kelas

h. Backpropagation dan *Gradient Descent* (Optimisasi)

Untuk memperbarui bobot dalam CNN, digunakan algoritma *Gradient Descent* yang bekerja dengan menghitung turunan *(gradien)* dari fungsi kerugian dan memperbarui bobot.

Rumus pembaruan bobot dengan *gradient descent*:

$$\mathrm{w} := w - \eta . \frac{\partial L}{\partial w} \tag{2.8}$$

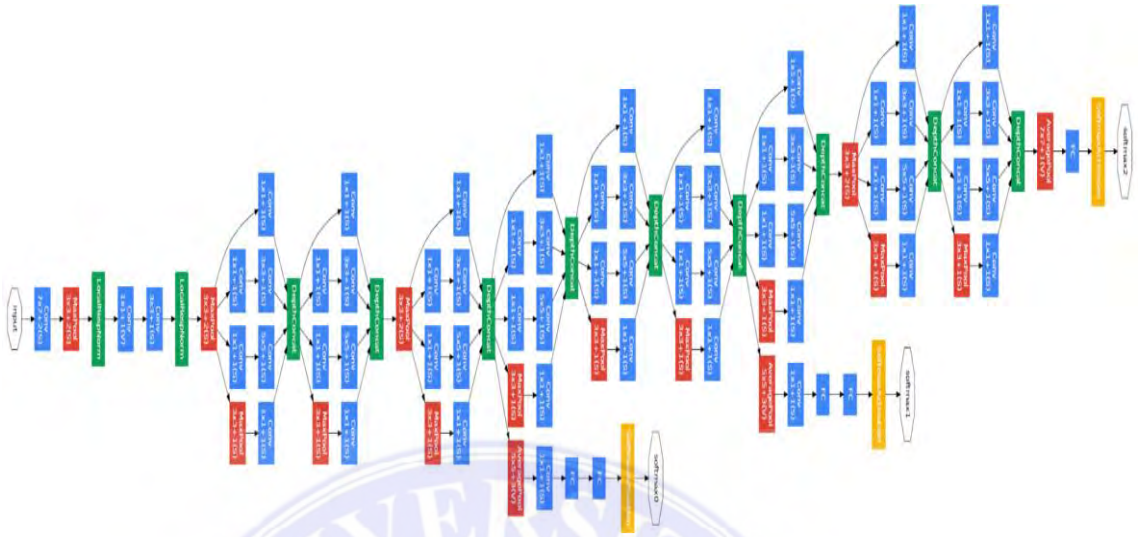
Keterangan:

η: *learning rate*, mengontrol sebearapa besar perubahan bobot setiap kali diperbaharui.

$\frac{\partial L}{\partial \omega}$: *gradien* dari fungsi kerugian *L* terhadap bobot *w*

w*:* bobot yang sedang diperbaharui.

2.4 Arsitektur *Googlenet*



Gambar 2.4 Arsitektur Googlenet

GoogleNet (Inception v1) adalah arsitektur jaringan konvolusi yang diperkenalkan oleh *Google* pada tahun 2014 dalam kompetisi ILSVRC (*ImageNet Large Scale Visual Recognition Challenge*). Arsitektur ini memperkenalkan konsep *Inception Module*, yang memungkinkan jaringan untuk melakukan konvolusi dengan filter yang berbeda-beda secara paralel, menangkap fitur di berbagai skala. *Efisiensi* Komputasi: Mengurangi jumlah parameter dengan menggunakan konvolusi 1×1 sehingga meminimalkan beban komputasi (Hatta Fudholi dkk., 2021).

1. *Inception Module*: Memperkenalkan berbagai ukuran filter (misalnya 1×1, 3×3 dan 5×5) yang bekerja secara paralel untuk menangkap fitur dari gambar dengan skala yang berbeda.

a. *Input Layer*

Input yang diberikan ke *GoogleNet* berupa gambar dengan dimensi tertentu. Misalkan *input* memiliki dimensi 224×224×3 (RGB). Tidak ada rumus yang digunakan di sini, karena ini adalah data mentah.

b. *Convolutional Layer*

GoogleNet menggunakan beberapa lapisan konvolusi untuk mengekstrak fitur dasar. Konvolusi dilakukan dengan filter berbeda, dan operasinya didefinisikan sebagai:

$$Y(i,j,k) = \sum_{M=1}^{F_H} + \sum_{c=1}^{F_w} + \sum_{n=1}^{C} \mathrm{X}\,(i+m, j+n, c).K(m,n,c,k) + b_k \quad (2.9)$$

Keterangan;

b_k : Bias untuk filter *k.*

F_H, F_w : Ukuran filter (tinggi dan lebar).

C : Jumlah saluran (*channel*) pada *input*

$Y(i,j,k)$: *Output* dari konvolusi pada posisi (*i,j*) dan filter ke- k

c. *Inception Module*

Inception Module adalah inti dari *GoogleNet,* di mana berbagai jenis operasi konvolusi (3x3, 5×5) dan max pooling dilakukan secara paralel dan digabungkan. Arsitektur ini mengkombinasikan fitur dari berbagai skala secara bersamaan.

Convolusi ke 1

$$y_{1x1} = \sum_{n=1}^{c} \mathrm{X}\,(i,j,c).K_{1x1}(1,1,c,k) + b_k \quad (2.10)$$

Convolusi ke 2

$$y_{3x3} = \sum_{M=1}^{3} + \sum_{c=1}^{3} + \sum_{n=1}^{c} \mathrm{X}\,(i+m, j+n, c).K_{3x3}(m,n,c,k) + b_k$$

(2.11)

Convolusi ke 3

$$y_{5x5} = \sum_{M=1}^{5} + \sum_{c=1}^{5} + \sum_{n=1}^{c} \mathrm{X}\,(i+m, j+n, c).K_{5x5}(m,n,c,k) + b_k$$

(2.12)

d. *Reduction Layers (*Lapisan Pengurangan Dimensi)

Konvolusi 1×1 digunakan secara intensif di seluruh jaringan, terutama untuk mengurangi dimensi sebelum operasi konvolusi yang lebih mahal, seperti 3×3 atau 5×5. Operasi ini ditujukan untuk mengurangi jumlah parameter dan komputasi, sambil tetap mempertahankan informasi penting.

$$y_{1x1} = \sum_{n=1}^{c} \mathrm{X}\,(i,j,c).K_{1x1}(1,1,c,k) + b_k \qquad (2.13)$$

e. *Fully Connected Layer* (Lapisan Sepenuhnya Terhubung)

Setelah melalui beberapa *Inception Module*, lapisan terakhir dari *GoogleNet* adalah *Fully Connected* Layer yang melakukan klasifikasi. Sama seperti CNN biasa, output dari lapisan *fully connected* dihitung sebagai:

$$y_i = \sum_j w_{i,j}.x_{,j} + b_i \qquad (2.14)$$

Keterangan:

$w_{i,j}$ = bobot antara neuron ke-*j* di lapisan sebelumnya dengan neuron ke-i.

$x_{,j}$ = *output* dari neuron ke-*j* di lapisan sebelumnya.

b_i= bias neuron ke-*i*.

y_i = *output* dari neuron ke-*i*.

6. ***Output Layer dan Softmax***

Pada tahap akhir, *output* dari lapisan *fully connected* diteruskan ke lapisan *Softmax* untuk menghasilkan probabilitas dari setiap kelas.

Rumus *Softmax*:

$$P(_{y,k}) = \frac{e^{zk}}{\sum_{n=1}^{k} e^{zj}} \qquad (2.15)$$

2.5 *Optimizer Adam*

Optimizer Adam (*Adaptive Moment Estimation*) adalah salah satu algoritma optimasi yang populer dalam pelatihan jaringan saraf *(Neural network*). Adam menggabungkan keuntungan dari optimizers lain *seperti AdaGrad dan RMSProp,* dengan menyesuaikan kecepatan pembelajaran (*learning rate*) dari setiap parameter berdasarkan estimasi momen pertama (rata-rata) dan momen kedua (variansi tak terpusat) dari gradien. Berikut adalah beberapa poin penting tentang Adam:

1. Momentum dan Kecepatan: *Adam* mengandung elemen momentum yang mempercepat pergerakan menuju minimum di arah yang relevan, serta elemen kecepatan untuk meredam osilasi saat mendekati nilai minimum.

2. Koreksi Bias: *Adam* menggunakan istilah koreksi bias dalam perhitungannya untuk mengatasi bias pada langkah awal. Ini memastikan estimasi lebih akurat selama tahap awal pelatihan.

3. *Hyperparameters*

   Learning Rate (α): Biasanya diset pada 0.001, meskipun perlu ditunin untuk hasil optimal.

   Beta 1 (β_1): Faktor peluruhan untuk momen pertama (biasanya 0.9).

   Beta 2 (β_2): Faktor peluruhan untuk momen kedua (biasanya 0.999).

   Epsilon (ε): Konstanta kecil (biasanya 10−810^{-8}10−8) untuk mencegah pembagian dengan nol.

2.6 *Optimizer Adamax*

Adamax adalah varian dari algoritma *Adam* yang menggunakan norma maksimum (atau normal tak hingga, ∞) sebagai gantinya, menjadikannya lebih stabil dalam situasi tertentu, terutama ketika gradien memiliki nilai yang sangat besar atau distribusi yang tidak biasa. *Adamax* merupakan singkatan dari *"Adam with-norm"*, dan formula utamanya serupa dengan *Adam* tetapi dengan sedikit perbedaan dalam skala gradiennya. Berikut adalah langkah-langkah utama dalam *Adamax*:

1. Memperbarui momen pertama: *Adamax* memperbarui nilai rata-rata eksponensial pertama mtm_tmt (estimasi rata-rata dari gradien).

$$mt = \beta 1 \cdot mt - 1 + (1 - \beta 1) \cdot gt$$

2. Memperbarui norma tak hingga: Daripada menggunakan momen kedua dalam bentuk norma L2 (seperti di *Adam*), *Adamax* memperbarui sebagai norma maksimum atau norma tak hingga dari gradien

$$ut = \max(\beta 2 \cdot ut - 1, | gt | 0)$$

3. Langkah pembaruan parameter: Parameter diperbarui dengan menggunakan laju pembelajaran α, dengan koreksi bias pada momen pertama mtm_tmt:

$$\theta t + 1 = \theta t - ut\alpha \cdot mt$$

2.7 *Optimizer AdaGrad*

Adagrad (*Adaptive Gradient Algorithm*) adalah algoritma optimasi yang secara adaptif menyesuaikan laju pembelajaran untuk setiap parameter selama proses pelatihan. Algoritma ini dikenal baik dalam menangani masalah *sparse data* (data jarang), misalnya dalam tugas *natural language processing* atau *computer vision*, karena mampu mempercepat *konvergensi* pada gradien yang jarang.

1. Penyesuaian Laju Pembelajaran untuk Setiap Parameter: *Adagrad* menyesuaikan laju pembelajaran secara khusus untuk setiap parameter berdasarkan *gradien* dari iterasi sebelumnya. Parameter yang jarang diperbarui akan memiliki laju pembelajaran yang lebih tinggi, sementara parameter yang sering diperbarui akan memiliki laju yang lebih rendah.
2. Akumulasi *Gradien* Kuadrat: Adagrad menyimpan akumulasi dari kuadrat gradien untuk setiap parameter. Laju pembelajaran dibagi dengan akar dari akumulasi ini, yang mencegah parameter besar melangkah terlalu jauh dan memperlambat pembaruan untuk parameter yang sering berubah.

Secara matematis, pembaruan parameter pada *Adagrad* dirumuskan sebagai:

$$\theta t + 1 = \theta t - \frac{\alpha}{\sqrt{Gt + e}} \cdot gt$$

2.8 *Optimizer AdaDelta*

Adadelta adalah algoritma optimasi adaptif yang dikembangkan untuk mengatasi salah satu kelemahan utama dari *Adagrad*, yaitu penurunan drastis pada laju pembelajaran seiring waktu. *Adadelta* tidak menyimpan akumulasi dari semua gradien sebelumnya, melainkan menggunakan pendekatan rata-rata eksponensial bergerak untuk menghitung perubahan gradien baru, sehingga lebih stabil dan tidak membutuhkan laju pembelajaran awal secara eksplisit. Berikut adalah karakteristik utama *Adadelta:*

1. Tidak Membutuhkan Laju Pembelajaran Awal: Berbeda dengan algoritma lainnya, *Adadelta* tidak memerlukan parameter laju pembelajaran eksplisit. Sebaliknya, ia menyesuaikan skala pembaruan parameter berdasarkan rata-rata eksponensial gradien sebelumnya, membuat pembaruan parameter lebih stabil.

2. Rata-rata Eksponensial dari *Gradien:* Adadelta menghitung rata-rata eksponensial bergerak dari kuadrat gradien (biasanya dilambangkan dengan $E[g\text{^}2]$ dan rata-rata *eksponensial* dari pembaruan parameter itu sendiri (biasanya dilambangkan dengan $E[\Delta\theta 2]$.

3. Langkah Pembaruan Parameter: Pembaruan parameter dilakukan dengan menghitung perubahan parameter ($\Delta\theta t$) berdasarkan kuadrat rata-rata eksponensial dari gradien dan dari pembaruan parameter sebelumnya:

$$\Delta\theta t = \frac{RMS[\Delta\theta]\mathrm{t} - 1}{2RMS[g]t} . gt$$

2.9 *Optimizer Nadam*

Nadam (Nesterov-accelerated Adaptive Moment Estimation) adalah varian dari algoritma *Adam* yang menggabungkan teknik *Nesterov momentum* untuk mencapai konvergensi yang lebih cepat dan stabil. Nadam menggabungkan kelebihan dari *Adam*—yang menggunakan momen pertama dan kedua dari gradien—dengan momen *Nesterov* yang membantu mempercepat optimasi pada arah yang tepat. Berikut adalah karakteristik utama *Nadam*:

1. Gabungan *Adam dan Nesterov Momentum*: Nadam memanfaatkan rata-rata bergerak dari gradien pertama dan gradien kuadrat, seperti pada *Adam*, namun menggunakan *momentum Nesterov*. Momentum *Nesterov* memperhitungkan gradien pada posisi yang diprediksi di masa depan, sehingga memungkinkan algoritma untuk "mengintip" posisi berikutnya dan melakukan penyesuaian lebih awal.

2. Pembaruan Momen Pertama dan Kedua: Nadam menggunakan dua momen rata-rata eksponensial—momen pertama untuk rata-rata gradien (mewakili arah) dan momen kedua untuk rata-rata kuadrat gradien (mewakili ukuran langkah). Momen pertama dihitung dengan menggunakan momentum *Nesterov*.

$$mt = \beta 1 \cdot mt - 1 + (1 - \beta 1) \cdot gt$$

3. Langkah Pembaruan Parameter dengan *Nesterov Momentum*: *Nadam* memperbarui parameter dengan menambahkan koreksi momentum *Nesterov*, yang memperhitungkan perkiraan posisi berikutnya untuk setiap parameter:

$$\theta t + 1 = \theta t - a.\frac{\beta 1 + (1 - \beta 1).\,\mathrm{gt}}{\sqrt{vt + e}}$$

4. Stabilitas dan Konvergensi Cepat: Karena menggabungkan *Adam* dengan momentum *Nesterov*, *Nadam* cenderung konvergen lebih cepat dan lebih stabil, terutama pada tugas-tugas yang kompleks dengan gradien yang berfluktuasi atau berubah cepat.

2.10 *Optimizer SGD*

Stochastic Gradient Descent (SGD) adalah salah satu algoritma optimasi yang paling sederhana dan paling umum digunakan dalam pelatihan model *machine learning* dan *deep learning*. Berbeda dengan *Gradient Descent* biasa, yang memperbarui parameter berdasarkan rata-rata gradien dari seluruh data pelatihan, *Stochastic Gradient Descent* memperbarui parameter berdasarkan satu contoh data acak atau satu mini-batch kecil data pada setiap langkah. Ini membuat SGD lebih efisien dan memungkinkan model untuk menangani dataset yang sangat besar.Gabungan *Adam dan Nesterov Momentum*: *Nadam* memanfaatkan rata-rata bergerak dari gradien pertama dan gradien kuadrat, seperti pada *Adam,* namun menggunakan momentum *Nesterov.* Momentum *Nesterov* memperhitungkan gradien pada posisi yang diprediksi di masa depan, sehingga memungkinkan

algoritma untuk "mengintip" posisi berikutnya dan melakukan penyesuaian lebih awal.

1. Pembaharuan Parameter Berdasarkan Sampel Acak: *SGD* memperbarui parameter setelah menghitung gradien dari satu sampel atau satu *batch* kecil data, bukan seluruh dataset. Ini memungkinkan pembaruan parameter yang lebih sering, mempercepat proses pelatihan, dan membuatnya lebih efisien untuk dataset besar.
2. Parameter: Pada setiap iterasi, parameter diperbarui berdasarkan *gradien* dari fungsi kerugian terhadap parameter saat ini:

$$\theta t + 1 = \theta t - \alpha \cdot gt$$

3. Kemampuan Generalisasi Lebih Baik: Karena pembaruan parameter dilakukan secara acak (*stochastic), SGD* sering kali lebih baik dalam menemukan minimum global dibandingkan gradient descent biasa, yang rentan terjebak pada minimum lokal. Variasi dalam pembaruan membantu menghindari jebakan minimum lokal dan dapat meningkatkan generalisasi model.
4. Kekurangan dan Penggunaan Momentum: *SGD* dapat mengalami fluktuasi atau osilasi pada arah yang tidak optimal saat mendekati minimum, yang membuat konvergensi menjadi lambat. Untuk mengatasi ini, sering kali ditambahkan *momentum*, yang membantu menstabilkan pembaruan dan mempercepat konvergensi. *SGD with Momentum* memperbarui parameter dengan mempertimbangkan laju dari pembaruan sebelumnya.

2.11 *Optimizer RMSprop*

RMSprop (Root Mean Square Propagation) adalah algoritma optimasi adaptif yang dikembangkan untuk mengatasi salah satu kelemahan utama dari *Adagrad*, yaitu penurunan laju pembelajaran yang terlalu cepat. *RMSprop* memperbaiki hal ini dengan menggunakan rata-rata bergerak eksponensial dari kuadrat gradien, sehingga memungkinkan pembaruan parameter yang stabil dan lebih efektif dalam proses pelatihan yang panjang. Berikut adalah karakteristik utama *RMSprop*:

1. Rata-rata Eksponensial dari Kuadrat Gradien: *RMSprop* menghitung rata-rata eksponensial dari kuadrat gradien alih-alih menyimpan akumulasi kuadrat gradien penuh, seperti pada *Adagrad*. Ini membuat skala pembaruan parameter tetap stabil meskipun melalui banyak iterasi, sehingga laju pembelajaran tidak mengalami penurunan drastis. Rumus untuk rata-rata eksponensial kuadrat gradien E[g^2]t
2. Langkah Pembaruan Parameter: Parameter diperbarui berdasarkan skala dari rata-rata kuadrat gradien, yang menyesuaikan pembaruan parameter sesuai dengan ukuran gradien. Pembaruan parameter pada *RMSprop* dituliskan sebagai:

$$\theta t + 1 = \theta t - \frac{a}{\sqrt{E[g^2]t + e}}$$

3. Adaptasi Laju Pembelajaran per Parameter: Dengan menggunakan rata-rata kuadrat gradien, *RMSprop* secara otomatis menyesuaikan laju pembelajaran untuk setiap parameter. Parameter dengan gradien besar akan mendapatkan langkah yang lebih kecil, sedangkan parameter dengan gradien kecil akan mendapatkan langkah yang lebih besar, sehingga proses pelatihan menjadi lebih seimbang.

4. Kinerja pada Data yang Berfluktuasi: *RMSprop* dirancang untuk bekerja dengan baik pada masalah yang memiliki gradien berfluktuasi, seperti pada jaringan saraf dalam (*deep networks).* Algoritma ini menjaga pembaruan parameter tetap stabil, terutama ketika gradien memiliki skala yang bervariasi.

2.12 *Confusion Matrix*

Evaluasi data *Confusion matrix* adalah proses yang digunakan untuk mengukur atau menghitung nilai akurasi pada image citra. Tabel pada proses evaluasi *confusion matrix* terdiri dari baris data yang akan uji diprediksi *true* (benar) dan *false* (tidak benar) suatu data yang di proses pada saat klasifikasi (Solikin, 2020). Seperti pada contoh gambar di bawah:

	Predicted Class	
True Class	***Positive***	***Negative***
Positive	TP	FN

Gambar 2.5 Metode Confusion Matrix

Keterangan :

TP (*True Positive*) : Memprediksi nilai positif dan hasilnya benar.

TN (*True Negative*) : Memprediksi nilai negative dan hasilnya benar.

FP (*False Positive*) : Memprediksi nilai positif dan hasilnya salah.

FN (*False Negative*) : Memprediksi nilai negatif dan hasilnya salah.

1. Akurasi

Akurasi mempresentasikan seberapa akurat sistem dapat mengklasifikasi data secara benar. Nilai akurasi adalah perbandingan antara data yang terklasifikasi benar dengan jumlah keseluruhan data (Solikin, 2020). Nilai akurasi dapat diperoleh dengan persamaan berikut.

$$Accuracy\ \frac{\text{TN+TP}}{\text{TN+FP+TP+FN}} \quad (2.16)$$

2. Presisi

Presisi adalah perbandingan nilai dari jumlah data kategori terklasifikasi benar dengan total keseluruhan data kategori terklasifikasi benar (Solikin, 2020). Untuk mencari nilai presisi dapat digunakan persamaan berikut

$$Presicion\ \frac{\text{TP}}{\text{TP+FP}} \quad (2.17)$$

3. *Recal*

Recall dilakukan untuk mengetahui perbandingan jumlah data kategori terklasifikasi benar oleh sistem dengan jumlah data kategori terklasifikasi benar dan salah (Solikin, 2020). Persamaan berikut dapat digunakan untuk memperoleh nilai *recall*.

$$Recall\ \frac{\text{TP}}{\text{TP+FN}} \quad (2.18)$$

4. F1-score

F1-Score merupakan penggabungan dari presisi dan *recal* l. Nilai *f1-score* dapat diperoleh dengan persamaan berikut.

$$F1\ \frac{2*Presicion*Recall}{\text{Presicion+Recall}} \quad (2.19)$$

2.13 Penelitian Terdahulu

Tabel 2.1 Penelitian Terdahulu

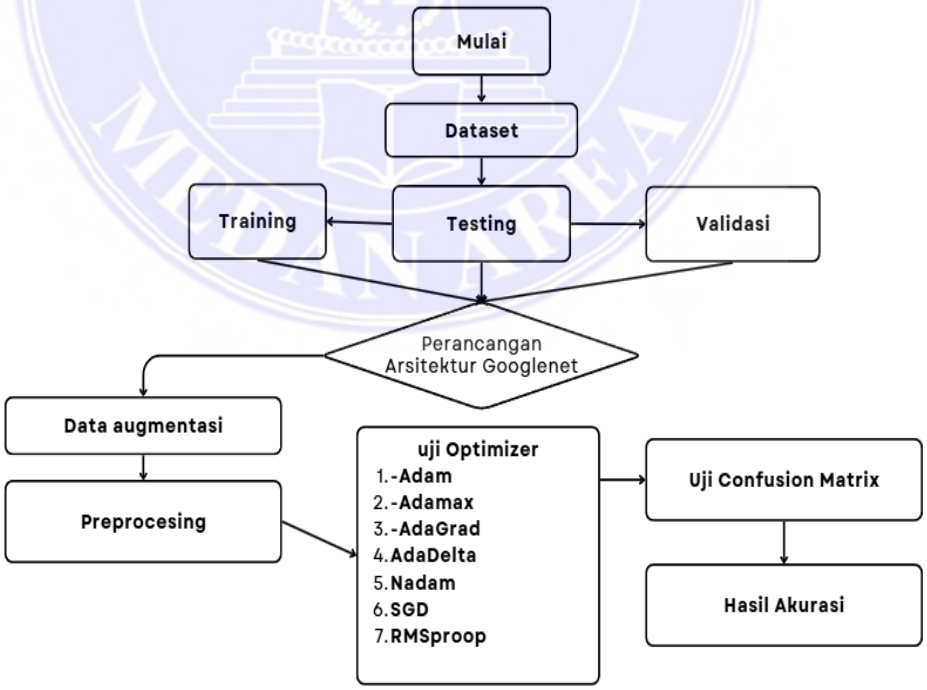
Penulis	Metode	Hasil
Mutiara Sholawati, Karina Auliasari, FX. Ariwibisono (2022)	*CNN*	Berdasarkan pengujian Bahasa isyarat menggunakan rumus *confusion matrix*, diperoleh nilai *accuracy, recall, specificity* dan *sensitivity* sebesar 80,76%.
Ariadi Retno Tri Hayati, Ririd, Yoppy Yunhasnawa, Yusliana Gadis Buata. (2018)	*LVQ*	Sistem ini dapat mengenali 24 huruf bahasa isyarat yang ditargetkan. Berdasarkan hasil dari perancangan dan pengujian sistem pada penelitian ini, persentase akurasi baca huruf terbaik adalah sebesar 88,75%.
Rohmat Indra Borman, Bentar Priyopradono. (2018)	*PCA*	Citra yang diuji total berjumlah 36 buah citra menghasilkan total rata-rata akurasi pengenalannya sebesar 97,8% dengan menggunakan metode ekstraksi *wavelet* dengan PCA.
Ayu Widya Agata, Wahyu S JSaputra, Chrystia AjiPutra (2024)	*CNN, SIFT*	Dalam penelitian, beberapa gerakan memerlukan percobaan berulang untuk deteksi yang akurat, mengingat kemiripan dengan gerakan bahasa isyarat lain. Meskipun demikian, akurasi tertinggi tercapai pada model dengan nilai *epoch* 55, mencapai 99.78%
Achmad Hasyim Nur'azizan, Abdul Riqza Ardiansyah, Rasio Fernandis. (2024)	*OPEN CV & MediaPipe*	Pada penelitian ini didapatkan hasil dari pengujian secara *real time* dihasilkan *f1 score* 0.9875, *recall* 0.9875, dan hasil *precision* 0.9875, dengan jumlah data sebanyak 400 dataset citra dan pengujian secara *real time*.

BAB III

METODOLOGI PENELITIAN

3.1 Prosedur Penelitian

Pada Gambar 3.1 merupakan diagram alur proses alur penelitian pengenalan bahasa isyarat menggunakan *Googlenet* yang meliputi dengan beberapa prosedur yang di awali dengan, persiapan dataset yang akan di uji kemudian data akan di bagi menjadi 3 data dengan *data training, testing*, dan *validasi* selanjutnya data akan dilakukan *preprocesing* setelah melalui *preprocesing* data akan di olah dengan menggunakan arsitektur CNN dengan rancangan *Googlenet* untuk menentukan menentukan akurasi menggunakan *Confusion Matrix* selanjutnya hasil dari pengujian akurasi arsitektur di analisa untuk menentukan akurasiyang dicapai.



Gambar 3.1 Alur Diagram Penelitian

3.2. Analisis Data

Data pada penelitian ini menggunakan data gambar yang di ambil melalui sumber *dataset kaggle* dengan format data gambar berbentuk PNG dengan bentuk citra bahas isyarat tangan dengan posisi tangan sesuai dengan abjad: a-z, Proses dalam penelitian ini menggunakan 3 proses yaitu proses *training testing*,dan *validasi* dalam proses *training,* dari data yang akan di analisis adapun model uji coba data yang digunakan sebnayak 2.040 data gambar. Model proses evaluasi digunakan untuk penelitian yaitu dengan membagi data menjadi 80% data dijadikan sebagai data training dan untuk 10% data dijadikan testing kemudian 10% untuk dijadikan data validasi. Penelitian ini menggunakan pendekatan eksperimental, yang bertujuan untuk menguji efektivitas arsitektur *Convolutional Neural Network* (CNN*) GoogLeNet* dalam mengklasifikasikan bahasa isyarat tangan. Eksperimen ini dilakukan dengan melatih model CNN menggunakan dataset citra bahasa isyarat tangan, yang kemudian dievaluasi performanya menggunakan berbagai *optimizer*, yaitu *Adam, Adamax, Adadelta, Stochastic Gradient Descent (SGD), RMSprop, AdaGrad*, dan *Nadam.* Untuk menganalisis kinerja model, model evaluasi yang digunakan adalah *confusion matrix dan F1-score.*

Tabel 3.1 Pembagian Dataset

Kelas	Data *Training*	Data *Testing*	Data Validasi	Jumlah
A	80	10	10	100
B	80	10	10	100
C	80	10	10	100
B	80	10	10	100
E	80	10	10	100
F	80	10	10	100
G	80	10	10	100
H	80	10	10	100
I	80	10	10	100

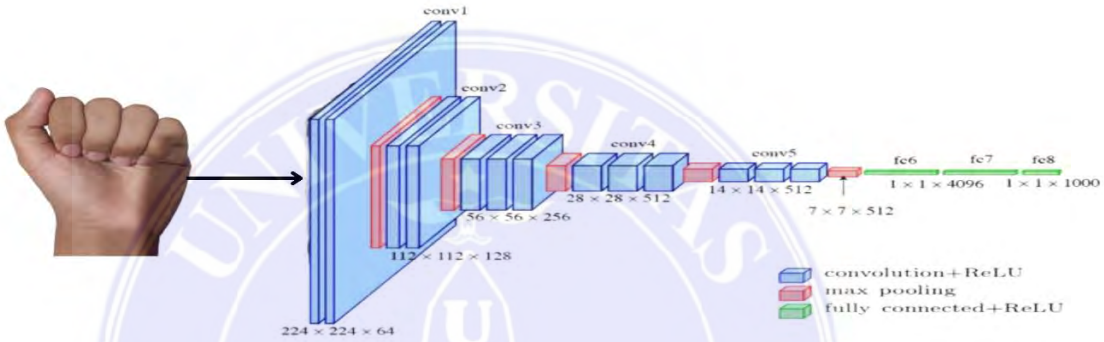
K	80	10	10	100
L	80	10	10	100
M	80	10	10	100
N	80	10	10	100
O	80	10	10	100
P	80	10	10	100
Q	80	10	10	100
R	80	10	10	100
S	80	10	10	100
T	80	10	10	100
U	80	10	10	100
V	80	10	10	100
W	80	10	10	100
X	80	10	10	100
Y	80	10	10	100
Total	1920	240	240	2.400
Persentase	80%	10%	10%	100%

3.3 *Preprocesing* Data

Pada model pelatihan penelitiaan ini menggunakan proses *preprocesing* proses dimna data akan di seleksi melalui beberapa tahap sebelum diproses mulai dari tahap, pembersihan data (*cleaning*) data yang sudah ada akan diseleksi kembali untuk mencengah kesalahpahaman data yang akan di proses penghapusan data yang tidak lengkap dan tidak relevan untuk mengatasi ketidaksalahpahaman data yang akan di proses ketika pengujian citra. Setelah data dalam tahap pembersihan dilanjutkan dengan tahapan data yang akan diolah akan disatukan dengan format yang sama tujuannya agar mempermudah proses analasis tahap terakhir mengurangi jumlah data dengan cara pengurangan dimensi, pengurangan jumlah, dan data kompresi data agar mempermudah proses pada tahap akhir.

3.3 Perancangan Metode CNN

Perancangan metode CNN pada penelitian ini dilakukan mulai dari tahapan seperti, menyusun proses pelatihan data yang akan digunakan dimulai dari menghitung jumlah lapisan yang digunakan, menentukan filter, terakhir menetapkan fungsi aktivasi dan menentukan ukuran *pooling*. Sebagai contoh ilustrasi jaringan CNN.



Gambar 3.2 Jaringan Arsitektur CNN

3.4 Pemodelan Arsitektur *Googlenet*

Pada proses penelitian ini Algoritma CNN dengan Arsitektur *Googlenet* digunkan untuk melakukan klasifikasi pengenalan pada bahasa isyarat tangan tujuan utamanya dalam pelatihan ini untuk menentukan seberapa baik arsitektur *Googlenet* dalam melakukan klasifikasi gambar bahasa isyarat tangan dengan menentukan hasil akurasi melalui uji evaluasi *confusion matrix F1-score.*

3.5 Arsitektur *Googlenet*

Metode *deep learning* CNN dengan tahapan proses pemodelan arsitektur Googlenet melakukan penginputan citra dengan menghitung akurasi yang akan dihasilkan. Pada penelitian ini berfokus untuk sepenuhnya menggunakan arsitektur *GoogleNet* menggunakan 57 lapisan *konvolusi.* Ini termasuk dalam

berbagai *Inception Module*, yang terdiri dari beberapa konvolusi paralel dengan filter yang berbeda-beda, Lapisan-lapisan pada konvolusi ini tersebar dari bagian awal (*feature extraction)* hingga modul *Inception* di bagian tengah jaringan. Namun, karena *GoogleNet* menggunakan pendekatan modular dengan banyak cabang paralel, jumlah lapisan konvolusi di setiap cabang berbeda-beda, sehingga total keseluruhan lapisan konvolusi tidak selalu terletak secara berurutan. Berikut ilustrasi berbentuk table dari struktur arsitektur *Googlenet*.

Tabel 3.2 Struktur Arsitektur *Googlenet*

No	Lapisan	Tipe operasi	Ukuran *input*	Ukuran *output*	Jumlah filter/ *poolingsize*	*Stride*	*Padding*
1	input	-	224x224x3	224x224x3	-	-	-
2	*Convoliutional layer*	7x7 *conv*	224x224x3	112x112x64	64	2	3
3	*Max pooling*	3x3 *maxpool*	112x112x64	56x56x64	3x3	2	0
4	*Convolutional layer*	1x1 *conv*	56x56x64	56x56x64	64	1	0
5	*Convolutional layer*	3x3 *conv*	56x56x64	56x56x192	3x3	2	0
6	*Max pooling*	3x3 *max pool*	56x56x192	28x28x256	*Multiple filters 64,*	1	*various*
7	*Inception module 1a*	1x1,3x3,5x5 *conv,*	28x28x92	28x28x256	*Multiple filters 64,*	1	*various*
8	*Inception module 1b*	1x1,3x3,5x5 *conv, maxpool*	28x28x256	28x28x480	*Multiple filters 128,192,96*	1	*various*
9	*max pooling*	3x3 *max pool*	28x28x480	14x14x512	*Multiple filters,192,208,48*	2	0
10	*Inception module 2a*	1x1,3x3,5x5 *conv, max pool*	28x28x256	28x28x480	*Multiple filters,128,192,96*	1	*various*
11	*Inception module 2b*	1x1,3x3,5x5*, conv,max pool*	14x14x512	14x14x512	*Multiple filters, 160,224,64*	1	*various*
12	*Inception module 2c*	1x1,3x3,5x5	14x14x512	14x14x512	*Multiple filters*	1	*various*

		conv,max pool			*128,256,64*		
13	*Inception module 2d*	1x1,3x3,5 x5 conv, max pool	14x14x 512	14x14x 528	*Multiple filters 112,288,64*	1	*various*
14	*Inception module 2e*	1x1,3x3,5 x5 *conv, max pool*	14x14x 528	14x14x 832	*Multiple filters 256,320,128*	1	*various*
15	*Max pooling*	3x3 *max pool*	14x14x 832	7x7x83 2	*3x3*	2	0
16	*Inception module*	1x1,3x3,5 x5 *conv,max pool*	7x7x83 2	7x7x83 2	*Multiple filters256,3 20,128*	1	*various*
17	*Inception module*	1x1,3x3,5 x5 *conv,max pool*	7x7x83 2	7x7x10 24	*Multiple filters384,3 284,128*	1	*various*
18	*Average pooling*	7x7 *Avg pool*	7x7x10 24	1x1x10 24	7x7	1	0
19	*dropout*	*Dropout* (40%)	1x1x10 24	1x1x10 24	-	-	-
20	*Fully connected (Fc)*	FC	1x1x10 24	1x1x10 00	1000	-	-
21	*softmax*	*softmax*	1x1x10 00	1x1x10 00	-	-	-

3.6 Uji Coba *Optimizer*

Optimizer adalah algoritma yang bertujuan untuk memperbarui bobot jaringan berdasarkan gradien error, agar model dapat belajar dari data latih dengan lebih efektif. *Optimizer* yang diuji dalam penelitian ini adalah:

1. Adam: Kombinasi dari *RMSprop* dan momentum, yang memiliki adaptasi laju pembelajaran untuk setiap parameter.
2. *Adamax*: Modifikasi dari *Adam* dengan penyesuaian norm L∞, cocok untuk jaringan dalam.
3. *Adadelta:* Berfokus pada akumulasi *gradien* yang memberikan pembelajaran lebih stabil untuk model.

4. *Stochastic Gradient Descent (SGD): Optimizer* dasar yang meng-update parameter secara sederhana.
5. *RMSprop:* Memiliki adaptasi terhadap laju pembelajaran per-parameter.
6. *AdaGrad:* Menggunakan penyesuaian laju pembelajaran untuk parameter yang jarang diperbarui.
7. Nadam: Kombinasi dari *Adam* dengan *Nesterov* momentum, yang dapat mempercepat konvergensi.

Tabel 3.3 Parameter *Googlenet*

No	*Input shape*	*Batch size*	*Epoch*	*Optimizer*	*Learning rate*
1	224x224x3	32	20	*Adam*	0.001
2	224x224x3	32	20	*Adamax*	0.001
3	224x224x3	32	20	*Nadam*	0.001
4	224x224x3	32	20	*AdaDelta*	0.001
5	224x224x3	32	20	*SGD*	0.001
6	224x224x3	32	20	*RMSproop*	0.001
7	224x224x3	32	20	*AdaGrad*	0.001

3.7 Confusion Matrix

Proses uji evaluasi klasifikasi data pada penelitian ini dengan *Confusion Matrix* dengan menghitung *akurasi, precision, recal dan F1-score* dalam menghitung nilai dan menentukan hasil dari pengujian TP dan FP pada data yang diolah menggunakan arsitektur *Googlenet*. Berikut tampilan dari ilustrasi *Confusion Matrix.*

Gambar 3.3 Perhitungan Confusion Matrix presisi, Recal, F1-score

3.8 Hasil Klasifikasi

Hasil klasifikasi akan dirangkum dalam satu kesimpulan dengan hasil akurasi terbaik dari arsitektur *Googlenet*, dengan melampirkan nilai dan proses akurasi pada arsitektur yang terlebih dahulu sudah di olah menggunakan metode evaluasi *Confusion Matrix F1-score* menjadi hasil akhir arsitektur *Googlenet*.

3.9 Alat dan Bahan Penelitian

Adapaun alat bahan yang digunakan untuk membantu proses penelitian ini yaitu sebagai berikut .

3.9.1 *Hardware*

Pada tabel 3.5 ditunjukkan tampilan spesifikasi *hardware* yang digunaan dalam proses pengujian arsitektur *GoogleNet*.

Tabel 3.4 *Hardware* yang Digunakan

No	*Hardware*	Keterangan
1.	Laptop	ASUS TUG *Gamming* F15 FX506LH_FX506LH
2.	*Prosessor*	*Intel (R) Core (TM)* i5-10300H CPU @ 2.50GHz (8 CPUs)
3.	SSD	500 GB
4.	RAM	8.00 GB

3.9.2. *Software*

Pada tabel 3.6 ditampilkan *software* yang akan memproses hasil pengujian dari arsitektur *GoogleNet*.

Tabel 3.5 *Software* yang Digunakan

No	*Software*	Keterangan
1.	Sistem Operasi	*Windows 11 Home Single Language*
2.	*Google Colab*	-RAM : 12.7 GB -Disk : 107.7 GB
3.	RAM	8 GB

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang sudah dilakukan, hasil percobaan pengujian berbagai *optimizer* pada arsitektur *GoogleNet* menunjukkan bahwa *RMSprop* memberikan performa terbaik dengan *F1-score* tertinggi sebesar 0.9917, berkat kemampuannya menyesuaikan *learning rate* secara dinamis. *Adamax* dan *Nadam* menyusul dengan *F1-score 0.9834,* menawarkan stabilitas dan efisiensi berkat penggabungan adaptasi *learning rate* dan momentum. *Adadelta* memperoleh *F1-score 0.9751*, sementara *Adam* mencapai 0.9461. *Adagrad dan SGD*, masing-masing dengan *F1-score 0.9336,* menunjukkan kinerja lebih rendah. Kesimpulannya, *RMSprop, Adamax,* dan *Nadam* adalah pilihan *optimizer* terbaik untuk model *GoogleNet*, sementara *Adagrad dan SGD* kurang optimal untuk tugas ini. Penelitian ini bermanfaat bagi penyandang tunanetra dalam mendeteksi abjad (bahasa isyarat tangan), pengembang teknologi dalam penerapan *deep learning*.

5.2. Saran

Untuk peneliti selanjutnya saya merekomendasikan agar memperbesar dan memperluas dataset, selain itu peneliti juga bisa menambahkan aplikasi *mobile android,* agar penyandang tuna netra dapat mendeteksi bahasa isyarat tangan secara *real time* di lapangan. Peneliti juga dapat menambahkan teknologi sensor yang memudahkan penyandang dalam memahami bahasa isyarat tangan, peneliti juga

dapat mengembangkan teknologi berbasis *camera* 3D dalam mendeteksi bahasa gerakan tangan, kemudian pentingnya kualitas gambar pada dataset mempengaruhi hasil akurasi yang akan di uji coba, usahakan gambar terlihat jelas dan memiliki kualitas baik sehingga model dapat mendeteksi jenis bahasa isyarat tangan dengan akurat agar penyandang dapat lebih mudah memahami bahasa gerakan tangan dalam berkomunikasi.

DAFTAR PUSTAKA

Abdulhakim, R., Carudin, & Arif Dermawan, B. (2021). Analisis dan Penerapan Algoritma Convolutional Neural Network untuk Klasifikasi Kendaraan Prioritas. *Jurnal Sains Dan Informatika*, *7*(2), 135–144.

Budiyanta, N. E., Mulyadi, M., & Tanudjaja, H. (2021). Sistem Deteksi Kemurnian Beras berbasis Computer Vision dengan Pendekatan Algoritma YOLO. *Jurnal Informatika: Jurnal Pengembangan IT*, *6*(1), 51–55.

Dako, R. D., & Ridwan, W. (2021). Pengujian karakteristik Functional Suitability dan Performance Efficiency tesadaptif.net. *Jambura Journal of Electrical and Electronics Engineering*, *3*(2), 66–71.

Fauzi, I., & Moenir, A. (2021). Klasifikasi Spesies Tanaman Magnolia Menggunakan Metode Convolutional Neural Networks. *Journal of Artificial Intelligence and Innovative Applications*, *2*(3), 2775–4057.

Fitrianingsih, & Rodiah. (2020). Klasifikasi Jenis Citra Daun Mangga Menggunakan Convolutional Neural Network. *Jurnal Ilmiah Teknologi Dan Rekayasa*, *25*(3), 223–238.

Hafifah, F., Rahman, S., & Asih, S. (2021). Klasifikasi Jenis Kendaraan Pada Jalan Raya Menggunakan Metode Convolutional Neural Networks (CNN). *TIN: Terapan Informatika Nusantara*, *2*(5), 292–301.

Hasby, A. N. F. A. (2022). *Sistem diagnosa Citra Histopatologi kanker Ginjal menggunakan Metode CNN Model GoogleNet*.

Hatta Fudholi, D., Abida Nayoan, R. N., Suyuti, M., & Rahmadi, R. (2021). Deteksi Indikasi Kelelahan Menggunakan Deep Learning. *Jurnal Sains Komputer & Informatika (J-SAKTI*, *5*(1), 1–9.

Kusumawardani, R., & Karningsih, P. D. (2021). Detection and Classification of Canned Packaging Defects Using Convolutional Neural Network. *PROZIMA (Productivity, Optimization and Manufacturing System Engineering)*, *4*(1), 1–11.

Lestari, D., & Aini, L. Q. (2021). Pengujian Konsorsium Bakteri Antagonis Untuk Mengendalikan Penyakit Bercak Daun Cercospora dan Virus Kuning pada Tanaman Cabai Merah Besar (Capsicum annum L.) di Kecamatan Dampit Kabupaten Malang. Jurnal Hama Dan Penyakit Tumbuhan, 9(3), 107– 114.

Mulyana, H. (2020). Klasifikasi Citra Pornografi Dengan Metode Convolutional Neural Network Pada Perangkat Smartphone Berbasis Android. *Universitas Mataram*.

Mustamin, F., Sari, Y., Khatimi, H., Informasi, S. T., Teknik, F., Mangkurat, U. L., & Banjarmasin, K. (2021). Menggunakan Arsitektur Cnn. *Jurnal Ilmu*

Komputer, *8*(1), 49–59.

Mustamin, N. F., Sari, Y., & Khatimi, H. (2021). Klasifikasi Kualitas Kayu Kelapa Menggunakan Arsitektur Cnn. Klik - Kumpulan Jurnal Ilmu Komputer, 8(1), 49.

Nasha Hikmatia A.E., & Zul, M. I. (2021). Aplikasi Penerjemah Bahasa Isyarat Indonesia menjadi Suara berbasis Android menggunakan Tensorflow. *Jurnal Komputer Terapan*, *7*(1), 74–83.

Queentinela, V. A., & Triyani, Y. (2021). Klasifikasi Penyakit Diabetic Retinopathy Pada Citra Fundus Berbasis Deep Learning. *9th Applied Business and Engineering Conference*, *1*, 1–11.

Ririd, N. A. R. T. H., Yunhasnawa, N. Y., & Buata, Y. G. (2018). Sistem Pengenalan Huruf Bahasa Isyarat Menggunakan Adaptive Learning Vector Quantization. *Jurnal Informatika Polinema*, *4*(2), 145. https://doi.org/10.33795/jip.v4i2.163.

Rizal, S., Ibrahim, N., Pratiwi, N. K. C., Saidah, S., & Fu'adaH, R. Y. N. (2020). Deep Learning untuk Klasifikasi Diabetic Retinopathy menggunakan Model EfficientNet. *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, *8*(3), 693.

Rochmawanti, O., Utaminingrum, F., & Bachtiar, F. A. (2021). Analisis Performa Pre-Trained Model Convolutional Neural Network dalam Mendeteksi Penyakit Tuberkulosis. *Jurnal Teknologi Informasi Dan Ilmu Komputer*, *8*(4), 805–814.

Setiawan, W. (2020). Perbandingan Arsitektur Convolutional Neural Network Untuk Klasifikasi Fundus. *Jurnal Simantec*, *7*(2), 48–53.

Sholawati, M., Auliasari, K., & Ariwibisono, F. (2022). Pengembangan Pengenalan Bahasa Isyarat Abjad Sibi Menggunakan Metode Convolutional Neural Network (Cnn). *Jati (Jurnal Mahasiswa Teknik Informatika)*, *6*(1), 134–144.

Soegeng, M. K., Liliana, L., & Noertjahyana, A. (2021). Penerapan Convolutional Neural Network untuk Klasifikasi Kanker Kulit Melanoma pada Dataset Gambar Kulit. *Jurnal Infra*, *9*(1), 47–51.

Solikin, S. (2020). Deteksi Penyakit Pada Tanaman Mangga Dengan Citra Digital: Tinjauan Literatur Sistematis (SLR). *Bina Insani Ict Journal*, *7*(1), 63.

Swapna, M., Sharma, D. Y. K., & Prasad, D. B. (2020). CNN Architectures: Alex Net, Le Net, VGG, Google Net, Res Net. *International Journal of Recent Technology and Engineering (IJRTE)*, *8*(6), 953–959.

Thiodorus, G., Prasetia, A., Ardhani, L. A., & Yudistira, N. (2021). Klasifikasi citra makanan/non makanan menggunakan metode Transfer Learning dengan model Residual Network. *Teknologi*, *11*(2), 74–83.

Wicaksono, M. I. (2020). Pendeteksian Api Berbasis Pengolahan Citra Digital Menggunakan Metode Convolutional Neural Network. *Etd.Repository.Ugm.Ac.Id*.

Yolanda, D., Gunadi, K., & Setyati, E. (2020). Pengenalan Alfabet Bahasa Isyarat Tangan Secara Real-Time dengan Menggunakan Metode Convolutional Neural Network dan Recurrent Neural Network. *Jurnal Infra*, *8*(1), 203–208.

Zakiah, Patmasari, R., & Saidah, S. (2021). Klasifikasi Jenis Kulit Wajah Menggunakan Metode Convolutional Neural Network (Skin Classification Using Convolutional Neural Network). *E-Proceeding of Engineering*, *8*(6), 11610–11616.

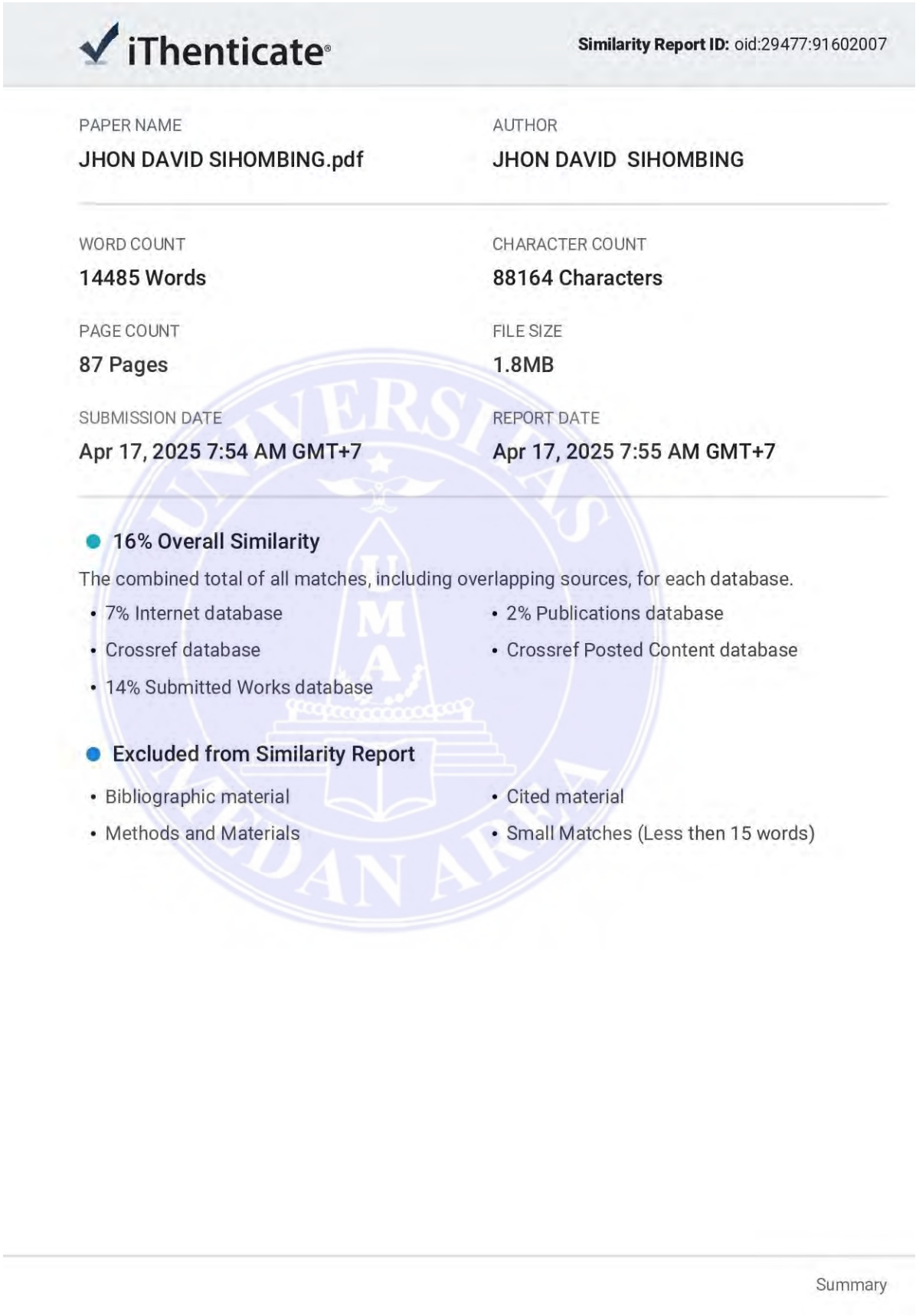
LAMPIRAN

Lampiran 1 Kode Program

```
import tensorflow as tf
from tensorflow.keras.applications.inception_v3 import InceptionV3
import numpy as np
from sklearn.model_selection import train_test_split
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau, TensorBoard
from tensorflow.keras.preprocessing.image import ImageDataGenerator
from sklearn.metrics import confusion_matrix, classification_report, f1_score
import matplotlib.pyplot as plt
import seaborn as sns
from google.colab import drive
import os
from tensorflow.keras.layers import GlobalAveragePooling2D, Dense, Dropout, BatchNormalization
from tensorflow.keras.models import Model
from tensorflow.keras.optimizers import RMSprop
from tensorflow.keras import regularizers
import time

import tensorflow as tf  # Import TensorFlow
from tensorflow.keras.applications.inception_v3 import InceptionV3  # Import GoogleNet (InceptionV3)
from tensorflow.keras.layers import Flatten, Dense  # Import Flatten, Dense
from tensorflow.keras.models import Model  # Import Model
from tensorflow.keras.optimizers import RMSprop  # Import RMSprop
import warnings
warnings.simplefilter("ignore", UserWarning)

# Gunakan GPU jika tersedia
with tf.device('/GPU:0'):
    # Load model InceptionV3 pre-trained tanpa fully connected layers
        base_model = InceptionV3(weights='imagenet', include_top=False, input_shape=(224, 224, 3))

    # Tambahkan layer baru di atas model InceptionV3
    x = base_model.output
    x = Flatten()(x)
    x = Dense(256, activation='relu')(x)
     predictions = Dense(24, activation='softmax')(x)  # Output untuk 24 kelas (categorical)

    # Freeze layer dari InceptionV3
    for layer in base_model.layers:
        layer.trainable = False
```

```
    # Buat model akhir
    model = Model(inputs=base_model.input, outputs=predictions)

    # Compile model
    model.compile(optimizer=RMSprop(learning_rate=0.001),
                  loss='categorical_crossentropy',  # Gunakan categorical crossentropy
untuk multi-class
                  metrics=['accuracy'])

    # Train model
    history = model.fit(
        train_generator,
        steps_per_epoch=train_generator.samples // train_generator.batch_size,
        epochs=20,  # Menjalankan training hingga 20 epoch
        validation_data=validation_generator,
                                  validation_steps=validation_generator.samples        //
validation_generator.batch_size
    )
```

Lampiran 2 Hasil Plagiasi



iThenticate®

Similarity Report ID: oid:29477:91602007

PAPER NAME
JHON DAVID SIHOMBING.pdf

AUTHOR
JHON DAVID SIHOMBING

WORD COUNT
14485 Words

CHARACTER COUNT
88164 Characters

PAGE COUNT
87 Pages

FILE SIZE
1.8MB

SUBMISSION DATE
Apr 17, 2025 7:54 AM GMT+7

REPORT DATE
Apr 17, 2025 7:55 AM GMT+7

16% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

- 7% Internet database
- 2% Publications database
- Crossref database
- Crossref Posted Content database
- 14% Submitted Works database

Excluded from Similarity Report

- Bibliographic material
- Cited material
- Methods and Materials
- Small Matches (Less then 15 words)

Summary

Document Accepted 26/8/25

Lampiran 3 SK Pembimbing

UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK
Kampus I : Jalan Kolam Nomor 1 Medan Estate / Jalan Gedung PBSI, Medan 20223
Kampus II : Jalan Sei Serayu Nomor 70 A / Jalan Setia Budi Nomor 79 B, Medan 20112 Telepon : (061) 8225602, 8201994
Fax : (061) 8226331 HP : 0811 607 259 website: www.uma.ac.id Email : univ_medanarea@uma.ac.id

Nomor : 1232/FT/01.10/V/2024 21 Mei 2024
Lampiran : -
Hal : **Pembimbing Tugas Akhir**

Yth. Pembimbing Tugas Akhir
Muhathir S.T, M.Kom (Sebagai Pembimbing)
di Tempat

Dengan hormat, sehubungan telah dipenuhinya persyaratan untuk memperoleh Tugas Akhir dari mahasiswa atas :

Nama : JHON DAVID SIHOMBING
NIM : 198160019
Jurusan : TEKNIK INFORMATIKA

Maka dengan hormat kami mengharapkan kesediaan saudara :

Muhathir S.T, M.Kom (Sebagai Pembimbing)

Adapun Tugas Akhir Skripsi berjudul :

Pengenalan Bahasa Isyarat Tangan Menggunakan Arsitektur Googlenet

SK Pembimbing ini berlaku selama enam bulan terhitung sejak SK ini diterbitkan. Jika proses pembimbing melebihi batas waktu yang telah ditetapkan, SK ini dapat ditinjau ulang.

Demikian kami sampaikan, atas kesediaan saudara diucapkan terima kasih.

Dekan,

Dr Eng. Supriatno.ST, MT.

Lampiran 4 Surat Pengantar Riset

UNIVERSITAS MEDAN AREA
FAKULTAS TEKNIK
Kampus I : Jalan Kolam Nomor 1 Medan Estate ☎ (061) 7360168, Medan, 20223
Kampus II : Jalan Setiabudi Nomor 79 / Jalan Sei Serayu Nomor 70 A ☎ (061) 42402994, Medan, 20122
Website: www.teknik.uma.ac.id E-mail: univ_medanarea@uma.ac.id

Nomor : 453 /FT.6/01.10/XI/2024 25 November 2024
Lamp : -
H a l : **Penelitian Dan Pengambilan Data Tugas Akhir**

Yth. Kepala Desa Medan Estate
Jln. Kolam No.12
Di
Medan

Dengan hormat,
Kami mohon kesediaan Bapak/Ibu berkenan untuk memberikan izin dan kesempatan kepada mahasiswa kami tersebut dibawah ini :

NO	NAMA	NPM	PRODI
1	Jhon David Sihombing	198160019	Teknik Informatika

Untuk melaksanakan Penelitian dan Pengambilan Data Tugas Akhir pada perusahaan/Instansi yang Bapak/Ibu Pimpin.

Perlu kami jelaskan bahwa Pengambilan Data tersebut adalah semata-mata untuk tujuan ilmiah dan Skripsi yang merupakan salah satu syarat bagi mahasiswa tersebut untuk mengikuti ujian sarjana pada Fakultas Teknik Universitas Medan Area dan tidak untuk dipublikasikan, dengan judul penelitian :

Pengenalan Bahasa Isyarat Tangan Menggunakan Arsitektur *GoogleNet*

Atas perhatian dan kerja sama yang baik diucapkan terima kasih.

Dekan,

Dr. Eng. Supriatno, ST, MT

Tembusan :
1. Ka. BPMPP
2. Mahasiswa
3. File

Lampiran 5 Surat Selesai Riset

PEMERINTAH KABUPATEN DELI SERDANG
KECAMATAN PERCUT SEI TUAN
DESA MEDAN ESTATE

ALAMAT : JALAN KOLAM NO. 12 KODE POS 20371

Medan Estate, 17 Desember 2024

Nomor : 140 / /J3
Sifat : Biasa
Lamp : ---
H a l : **Izin Penelitian**

Kepada Yth :
Dekan Fakultas Teknik
UNIVERSITAS MEDAN AREA
Di--
Tempat

Dengan hormat,

Sehubungan dengan Surat Dekan Fakultas Teknik Universitas Medan Area Nomor : 453/FT.6/01.10/XI/2024 tertanggal 25 November 2024 hal Penelitian dan Pengambilan Data Tugas Akhir terhadap mahasiswa yang namanya tercantum dibawah ini :

Nama	: **JHON DAVID SIHOMBING**
NIM	: 198160019
Jurusan/Prodi	: Teknik Informatika
Judul Penelitian	: **Pengenalan Bahasa Isyarat Tangan Menggunakan Arsitektur *Google Net***

bersama ini kami sampaikan bahwa Mahasiswa tersebut diatas telah selesai melakukan Penelitian dan Pengambilan Data di Desa Medan Estate.-

Demikian hal ini kami sampaikan agar menjadi maklum.

Kepala Desa Medan Estate
Kecamatan Percut Sei Tuan

PEMERINTAH KABUPATEN DELI SERDANG DESA MEDAN ESTATE KEC. PERCUT SEI TUAN

ASDAT LUBIS