

II.5.4.7. *Restricted Boltzmann Machine (RBM)*

Restricted boltzmann machine (RBM) adalah model grafis probabilistik atau jenis jaringan saraf acak. Algoritme ini adalah arsitektur yang kuat untuk pemfilteran kolaboratif dan melakukan analisis faktor biner dengan komunikasi terbatas antar kelas untuk pembelajaran yang efisien. Jaringan memiliki lapisan unit yang terlihat, lapisan unit tersembunyi, dan unit bias yang terhubung ke semua unit yang terlihat dan tersembunyi. Unit tersembunyi independen sebagai sarana untuk menyediakan sampel yang tidak bias. Neuron dalam grafik bipartit memiliki koneksi simetris. Namun, tidak ada koneksi antar node dalam grup. RBM menawarkan keunggulan dalam pembelajaran berbasis energi, seperti fleksibilitas desain yang berguna pada model statistik probabilistik dan non-probabilistik, konektivitas terbatas untuk kemudahan pelatihan dan penggunaan dengan model statistik, klasifikasi, regresi, dan generalisasi. RBM berguna untuk sistem pemberi rekomendasi, pemfilteran fitur, pembelajaran, pengubahan ukuran, pemodelan topik.

II.5.4.8. *Deep Belief Network (BDN)*

Deep Belief Network (DBN) adalah algoritma pembelajaran mendalam probabilistik tanpa pengawasan di mana jaringan memiliki model pembelajaran umum. Algoritme adalah kombinasi dari jaringan grafik terarah dan tidak terarah, dengan lapisan atas RBM tidak terarah dan lapisan bawah menghadap ke bawah. Algoritma ini memungkinkan pra-pelatihan dan jaringan feed-forward untuk tahap

penghalusan. DBN memberikan pembelajaran berbasis energi dan bisa mendapatkan keuntungan dari data yang tidak berlabel.

II.6 Related Works

Pada bagian ini kami mengumpulkan beberapa *paper* yang berkaitan dengan penelitian ini, yang tujuannya adalah sebagai pedoman ataupun pembanding yang intinya dapat menghasilkan informasi yang dapat membantu dalam penelitian ini. kumpulan review paper tersebut akan di sajikan pada Tabel II.2.

Dalam tabel II.2 dapat dilihat bahwa penggunaan metode *deep learning* pada *face identification* dan *face recognition* masih menjadi metode yang lebih unggul dibandingkan metode lainnya. Tingkat akurasi yang di peroleh pada metode *deep learning* tersebut masih mengungguli metode lainnya dalam hal deteksi dan pengenalan wajah. Atas landasan tersebut yang mendorong penulis untuk menggunakan metode *deep learning* tersebut untuk di terapkan pada penelitian ini.

Tabel II.2 Penelitian Terkait dengan Deep Learning Dan Deteksi Wajah

No	Nama Penulis	Publish	Judul	Pembahasan
1	Iwan Setyawan ¹ , Ivanna K. Timotius ² dan Andreas A. Febrianto (2011)	Journal of ICT Research and Applications, Issue: 3, Volume: 5, Page: 157-172 Tahun 2011	Frontal Face Detection using Haar Wavelet Coefficients and Local Histogram Correlation	Dalam penelitian ini penulis melihat bahwa makalah yang menyajikan sistem deteksi wajah yang dirancang untuk mendeteksi wajah frontal menggunakan wavelet Haar koefisien dan korelasi histogram lokal sebagai fitur pembeda. training sistem menggunakan 100 gambar training dan Eksperimen tersebut menunjukkan menunjukkan bahwa sistem yang diusulkan bekerja dengan baik selama Testing, mencapai tingkat deteksi 91,5%.
2	Alfredo Bayu Satriya, Siti Agustini (2017)	INTEGER: Journal of Information Technology Tahun 2017	Face Recognition using Modified Triangle Method	Dalam penelitian ini penulis melihat bahwa deteksi wajah menggunakan Metode Triangle modifikasi 9 mencapai 90% mencapai tingkat keberhasilan untuk deteksi wajah dan 50% untuk pengenalan wajah. "
3	Diaa Salama Abdelminaam , Abdulrhman M Almansori, Mohamed Taha (2020)	Public Library of Science Tahun 2020	A deep facial recognition system using computational intelligent algorithms	Pada penelitian ini penulis melihat bahwa computational intelligent algorithms memiliki akurasi tinggi mencapai 99,4% sistem pengenalan wajah mendalam
4	Romi Fadillah Rahmat, M P Loi, S Faza, D Arisandi, Rahmat Budiarto (2019)	Journal of Physics: Conference Series, Issue: 1, Volume: 1235, Page: 12119 tahun 2019	Facial Recognition for Car Security System Using Fisherface Method	Pada penelitian ini penulis melihat bahwa hasil penelitian menunjukkan bahwa metode fisherface yang diterapkan pada pengenalan wajah untuk keamanan mobil bekerja dengan baik dengan akurasi sebesar 83,04%.

No	Nama Penulis	Publish	Judul	Pembahasan
5	Subiyanto , Dina Priliyana , Moh Eki Riyadani , Nur Iksan , Hari Wibawanto (2020)	Jurnal Teknologi dan Sistem Komputer, Issue: 3, Volume: 8, Page: 210-216 tahun 2020	Face recognition system with PCA-GA algorithm for smart home door security using Rasberry Pi	Dalam penelitian ini penulis melihat bahwa bahwa pengenalan wajah menggunakan PCA-GA memiliki akurasi 90%, sedangkan PCA dan LBPH-GA masing-masing memiliki akurasi 80% dan 90%.
6	Jacky Efendi , Muhammad Ihsan Zul , Wawan Yunanto (2017)	JOIV : International Journal on Informatics Visualization, Issue: 4 tahun 2017	Real Time Face Recognition using Eigenface and Viola-Jones Face Detector	Dalam penelitian ini berdasarkan Testing yang dilakukan didapatkan bahwa aplikasi ini dapat mengenali citra wajah dengan akurasi sebesar 90% dan terendah 6,67%.
7	Anifuddin Azis , Muhamad Haikal (2007)	IJCCS (Indonesian Journal of Computing and Cybernetics Systems), Issue: 2, Volume: 1 tahun 2007	Face Detection Using Backpropagation Neural Networks	Penulis melihat hasil yang diperoleh pada penelitian ini yaitu,persentasi keberhasilan yang dihasilkan adalah sebesar 90%.
8	Mohammed Alghaili ,Zhiyong Li , Hamdi A R Ali (2020)	Scientific Programming, Tahun 2020	FaceFilter: Face Identification with Deep Learning and Filter Algorithm	penulis melihat pada penelitian ini dengan menggunakan deep learning memiliki hasil akurasi 99.7% pada kumpulan data Wajah Berlabel di Alam Liar dan 94,02% di YouTube Faces DB

No	Nama Penulis	Publish	Judul	Pembahasan
9	Sara Shahsavarani , Morteza Analoui , Reza Shoja Ghiass (2020)	arXiv: Computer Vision and Pattern Recognition, Tahun 2020	Deep-ID: A Novel Model for Multi-View Face Identification Using Convolutional Deep Neural Networks	Dalam penelitian ini penulis melihat bahwa Hasil eksperimen yang dilakukan menggunakan Convolutional Deep Neural Networks menunjukkan bahwa akurasi yang dihasilkan mencapai 99,8% sedangkan metode mutakhir mencapai akurasi 97%.
10	Monali Nitin Chaudhari, Mrinal Deshmukh , Gayatri Ramrakhiani, Rakshita Parvatikar (2018)	Fourth International Conference on Computing Communication Control and Automation (ICCUBECA), pada tahun 2018	Face Detection Using Viola Jones Algorithm and Neural Networks	Dalam penelitian ini penulis melihat bahwa algoritma yang di usulkan mencapai akurasi keberhasilan sebesar 90 % dalam hal mengidentifikasi wajah dengan kehadiran mata
11	Nanik Herawati (2014)	Program Studi Magister Teknik Informatika Fakultas Ilmu Komputer Universitas Dian Nuswantoro Semarang 50131, Indonesia tahun 2014	Algoritma Viola Jones dan Adaboost Untuk Deteksi Wajah Manusia Berdasarkan Perbedaan Ruang Warna	Penulis melihat bahwa Hasil yang diperoleh dari penelitian ini adalah ruang warna yang terbaik pada citra dengan satu wajah menggunakan ruang warna RGB dengan rata-rata MSE senilai 135,49 dengan keberhasilan mendeteksi wajah 96%.

No	Nama Penulis	Publish	Judul	Pembahasan
12	Marde Fasma Ul'aza (2020)	Institut Teknologi Sepuluh Nopember tahun 2020	Pengenalan Ekspresi Wajah pada Sistem Monitoring Perkuliahan Menggunakan Metode Deep Learning	Pada penelitian ini penulis melihat Berdasarkan evaluasi terhadap deteksi wajah dengan menggunakan IoU threshold, model yang dibangun menghasilkan precision sebesar 98,83%, recall sebesar 99,88%, F1-score sebesar 99,35%, dan Average Precision sebesar 99,72% yaitu pada nilai IoU threshold sebesar 50%. Berdasarkan evaluasi terhadap waktu proses sistem, didapatkan rata-rata waktu proses deteksi wajah selama ± 300 milidetik, rata-rata waktu proses pengenalan ekspresi wajah selama ± 8 milidetik, dan rata-rata waktu proses pelacakan wajah selama ± 10 milidetik.
13	Wulan Anggraini (2020)	repository.ar-raniry.ac.id tahun 2020	Deep Learning Untuk Deteksi Wajah Yang Berhijab Menggunakan Algoritma Convolutional Neural Network (CNN) Dengan Tensorflow	Pada penelitian ini, penulis menemukan bahwa aplikasi CNN menggunakan Tensorflow dengan bahasa pemrograman Python. Jumlah dataset yang digunakan sebanyak 300 citra wajah hijab. Berdasarkan hasil pembahasan, tingkat akurasi yang dicapai adalah 92% saat pelatihan dan 87% saat pengujian. Oleh karena itu, dari penelitian ini dapat disimpulkan bahwa kinerja model yang diimplementasikan dalam penelitian ini dapat dianggap sebagai kinerja yang optimal dalam mendeteksi citra wajah menggunakan atribut jilbab.
14	Yunita Aulia Hasma , Widy Silfianti (2018)	Jurnal Ilmiah Teknologi dan Rekayasa, Issue: 2, Volume: 23, Page: 89-102 tahun 2018	IMPLEMENTASI DEEP LEARNING MENGGUNAKAN FRAMEWORK TENSORFLOW DENGAN METODE FASTER REGIONAL CONVOLUTIONAL NEURAL NETWORK UNTUK	Penulis melihat dalam penelitian ini bahwa pada tahap Testing gambar yang pertama dengan menggunakan 1 gambar, hasil dari proses pendeteksian terhadap model memiliki presentasi yang cukup tinggi, yaitu 98% untuk bekas, 99% untuk jerawat dan 99% untuk pus. Testing kedua menggunakan gambar/model yang memiliki beberapa objek di dalamnya dan didapatkan akurasi keberhasilan pendeteksian objek sebesar 72,4%.

No	Nama Penulis	Publish	Judul	Pembahasan
			PENDETEKSIAN JERAWAT	
15	Dwi Agung Al Ayubi , Dwi Arman Prasetya , Irfan Mujahidin (2020)	CYCLOTRON, Issue: 1, Volume: 3, tahun 2020	Pendeteksi Wajah Secara Real Time pada 2 Degree of Freedom (DOF) Kepala Robot Menggunakan Deep Integral Image Cascadeac	Penulis melihat pada penelitian ini Untuk akurasi realtime face detector pada penelitian ini dengan Testing presisi terbesar yaitu 95,25% dengan waktu respon deteksi tercepat 7 detik dengan waktu terlama 8,55 detik rata-rata data citra semuanya tidak terdeteksi dengan baik.
16	Daru Prasetyawan, Shofwatul Uyun (2020)	JISKA (Jurnal Informatika Sunan Kalijaga), Issue: 1, Volume: 5, Page: 23-35 tahun 2020	PENENTUAN EMOSI PADA VIDEO DENGAN CONVOLUTIONAL NEURAL NETWORK	Penulis melihat dalam penelitian ini akurasi yang diperoleh 74%, rata-rata presisi 75,05%, dan rata-rata recall 74%. Metode ini masih dirasa kurang efektif diterapkan pada deteksi emosi pada video yang dibuktikan dengan nilai akurasinya yang kurang baik.

No	Nama Penulis	Publish	Judul	Pembahasan
17	Freddy Nur Afandi , Ramses Parulian Sinaga, Yuthsi Aprilinda, Fenty Ariani (2019)	Jurnal Sistem informasi dan telematika, Issue: 2, Volume: 10, Page: 331211 tahun 2019	IMPLEMENTASI FACE DETECTION PADA SMART CONFERENCE MENGUNAKAN VIOLA JONES	Penulis melihat Berdasarkan hasil penelitian yang dilakukan terhadap Implementasi face detection pada Smart Conference menggunakan Viola-Jones, maka dapat disimpulkan bahwa RabbitMQ sudah dapat diterapkan dengan membangun aplikasi service untuk parallel computing sebagai penghubung antar client dan server dan Sistem dapat mendeteksi wajah manusia dengan tingkat akurasi 72%.
18	Feni Budi Antono , Faqih Rofii, Istiadi Istiadi (2020)	Techno.Com, Issue: 1, Volume: 19, Page: 12-23 tahun 2020	Deteksi Jumlah dan Pengenalan Wajah Manusia Menggunakan Metode Histogram Of Oriented Gradient dan Viola Jones	Pada penelitian ini penulis melihat bahwa hasil Testing menunjukkan tingkat akurasi mencapai 92,72 % dan nilai eror sebesar 7,28 % dalam mengenali jumlah manusia. Pada Testing pengenalan wajah sistem mampu mendeteksi pada jarak 1 – 3 meter Hasil Testing menunjukkan tingkat akurasi mencapai 26,67% dan memiliki nilai error sebesar 73,33 % dalam mengenali wajah manusia. Pada pendeteksian adanya wajah didapatkan akurasi sebesar 100 % dan error 0 %.
19	Mohamad ihsan , Ratih Kumalasari Niswatin, Daniel Swanjaya (2021)	Universitas PGRI Kediri Tahun 2021	DETEKSI EKSPRESI WAJAH MENGUNAKAN TENSORFLOW	Penelitian ini menggunakan machine learning open source library Tensorflow dengan metode Convolutional Neural Network (CNN) yang dirancang khusus untuk pengenalan dan menentukan klasifikasi terhadap 7 ekspresi dasar wajah manusia ditambah ekspresi netral, metode ini memiliki hasil paling signifikan dalam hal pengenalan citra. Pemerataan distribusi data akan dilakukan untuk meningkatkan kinerja model. Hasil dari Testing analisis di dapatkan hasil parameter optimal batch 32, epoch 100 dan dropout 0.6 dengan tingkat akurasi training 62.24%, akurasi validasi 62,44%, training loss 4,54% dan validation loss 4,02%.

No	Nama Penulis	Publish	Judul	Pembahasan
20	Aries Suharso (2017)	Jurnal Ilmu Komputer dan Teknologi Informasi, Issue: 2, Volume: 1, tahun 2017	Pengenalan Wajah Menggunakan Metode Viola-Jones dan Eigenface Dengan Variasi Posisi Wajah Berbasis Webcam	Penerapan Metode Viola-Jones untuk mendeteksi wajah dan metode eigenface digunakan untuk mengenali wajah, aplikasi yang dibangun menggunakan Microsoft visual studio dan fungsi capture pada webcam menggunakan OpenCV. Testing dilakukan pada intensitas cahaya normal dengan jarak dan posisi wajah yang telah ditentukan, pengolahan data uji dilakukan menggunakan Matlab. Hasil dari penelitian ini menunjukkan tingkat akurasi deteksi citra wajah sebesar 88.89% dengan sudut rotasi variasi posisi citra wajah menghadap ke depan (frontal) - 150, kesamping kiri 300 dan kanan 300 dan terbukti pada nilai threshold 75 memiliki akurasi tertinggi dan jumlah dikenali tertinggi yaitu sebesar 90.90%
21	W. Hutamaputra, R. Y. Krisnabayu, M. Mawarni, N. Yudistira, and F. A. Bachtiar (2022)	Jurnal Teknologi dan Sistem Komputer, vol. 0, Jan. 2022. https://doi.org/10.14710/jtsiskom.2021.14045	Perbandingan Convolutional Neural Network VGG16 dan ResNet34 pada Sistem Klasifikasi Sampah Botol	Hampir semua botol minuman kemasan yang beredar di masyarakat terbuat dari bahan plastik dikarenakan plastik adalah bahan yang murah dan mudah dibentuk. Plastik adalah bahan non-organik yang sulit diuraikan sehingga botol plastik dapat menyebabkan pencemaran lingkungan. Sehingga diperlukan suatu solusi yang efektif untuk mengatasi kerusakan lingkungan yang disebabkan oleh sampah botol plastik. Salah satu solusi yang dapat dilakukan yaitu melakukan klasifikasi dan daur ulang sampah botol plastik. Pengklasifikasian sampah botol plastik dan sampah botol bukan plastik ke dalam kategori yang ditentukan sesuai dengan persyaratan kemudian didaur ulang agar dapat diolah kembali agar tidak merusak lingkungan. Artikel ini mengusulkan model VGG16 dan ResNet34 berbasis deep learning menggunakan CNN (Convolutional Neural Network) untuk mengidentifikasi dan mengklasifikasikan sampah botol. Berdasarkan hasil Testing menggunakan Convolutional Neural Network, arsitektur VGG16 memiliki akurasi sebesar 90% dan ResNet34 memiliki akurasi sebesar 50% pada klasifikasi botol plastik dan bukan botol plastik. Masing-masing arsitektur menggunakan 10 epoch, 32 batch, 1655 gambar.
22	Rizqy Joventus Gunawa, Budhi Irawan, Casi Setianingsih	e-Proceeding of Engineering : Vol.8, No.5 Oktober 2021	Pengenalan Ekspresi Wajah Berbasis Convolutional Neural Network Dengan Model Arsitektur Vgg16	Facial expression recognition system technology can continue to develop over time with new researches to create more sophisticated and accurate systems, one of the developments in this case is using the Deep Learning method.. In this research, to design a facial expression recognition system using the CNN algorithm, the model used is a CNN-based architecture, namely Visual Geometry Group-16 Weight Layer (VGG16). The dataset used in this system is FER2013, where this dataset has a large number of facial images, amounting to 35,887 images with 7 categories of emotions. The best performance results were obtained in the proposed model, namely the Modified VGG16 model with the parameters using augmentation data, epoch 100, and learning rate 0.001 which reached a test accuracy of 70.63%, this accuracy is much better than previous studies using the VGG16 model base and the FER2013 dataset.

II.7 Deteksi Wajah dengan Opencv dan Python

Mengingat *opencv* adalah *library* yang dikembangkan program tingkat baris perintah kecil untuk memanfaatkan fungsionalitas *opencv* yang ingin kami optimalkan (Bruce dan Aiken 2016). Jadi *opencv* adalah library yang menggunakan perintah baris sederhana. Dalam hal ini *opencv* nantinya akan dikolaborasikan dengan Library Tensorflow yang digunakan untuk meningkatkan akurasi pengenalan gambar.

Dalam hal ini digunakan *python* sebagai bahasa pemrograman. Bahasa Python adalah improvement dari bahasa C dan C++. Bahasa *python* dibuat sebagai bentuk dari keinginan para developer untuk membuat bahasa serupa C++ (Suadi, Christia 2013). Berikut adalah alasan dipilihnya bahasa python sebagai bahasa pemrograman. Python adalah bahasa pemrograman model skrip yang berorientasi obyek. Python dapat diimplementasikan untuk berbagai keperluan pengembangan perangkat lunak dan dapat dioperasikan pada berbagai platform sistem operasi. Para programmer menerapkan bahasa python untuk membuat sebuah aplikasi baik aplikasi desktop, web, game atau aplikasi yang lainnya. Karena saat ini python adalah salah satu bahasa yang populer digunakan (Malikhah, Lidya, Sukendar 2016). Python dipilih karena memiliki struktur lebih sederhana dibanding pemrograman lain (Prasetya, Nurviyanto 2012).

Dari uraian diatas penulis memutuskan untuk menggunakan library *opencv* dipadukan dengan bahasa pemrograman python karena penggunaannya yang sederhana dan diharapkan dapat diterapkan pada sistem deteksi wajah yang menjadi topik pada penelitian ini secara efektif.

II.8 Ringkasan Bab II

Identifikasi wajah adalah bidang penelitian dalam Computer Vision dan otentikasi biometrik yang dinilai paling aktif. deteksi wajah dapat dilakukan melalui foto, video, serta web cam, dan deteksi wajah adalah syarat utama pada pekerjaan yang berhubungan dengan kecerdasan buatan yaitu verifikasi wajah, penandaan dan pengambilan wajah, serta pelacakan wajah. Pengenalan wajah adalah bagian dari pengenalan biometrik karena wajah adalah bagian unik dari tubuh manusia. Deteksi wajah sedang menjadi topik pembahasan yang paling populer dibahas pada beberapa tahun terakhir ini. Pada penelitian ini penulis memilih metode deep learning sebagai metode penelitian karena dapat dilihat dari penelitian-penelitian sebelumnya yang tertulis pada bab ini melalui tabel II.1 yang berisi review perbandingan penelitian-penelitian sebelumnya bahwa metode deep learning lebih unggul dalam hal face identification dan face recognition dibandingkan metode lainnya. Dalam penelitian ini akan dikhususkan Algoritma CNN yang didukung dengan model arsitektur Visual Geometry Group 16 (VGG 16) Itulah alasan kuat penulis memilih metode deep learning yang menggunakan algoritma CNN dengan model arsitektur VGG 16 sebagai metode dan model dalam penelitian ini.

Berbagai kajian yang terdapat pada bab ini akan digunakan untuk proses penelitian yang akan dilakukan. Pada Bab III akan lebih dijelaskan lagi tentang metodologi dan alur dari penelitian ini. Yaitu dari proses awal pada penelitian ini hingga proses akhir penelitian.



BAB III
METODOLOGI PENELITIAN

BAB III

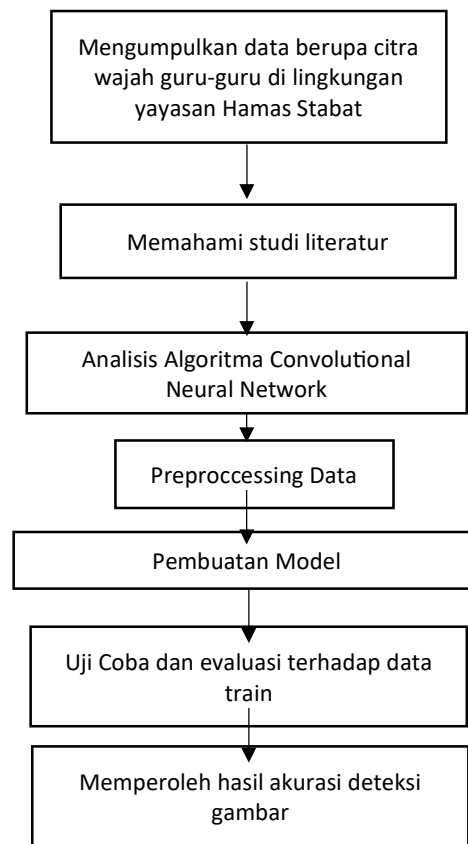
METODOLOGI PENELITIAN

III.1. Pendahuluan

Pada bab ini menjelaskan tentang metodologi penelitian apa yang akan digunakan untuk penyelesaian masalah pada penelitian ini. Pada metodologi penelitian ini akan sangat membantu penulis dalam menyelesaikan masalah karena berguna dalam menemukan suatu kebenaran, terkait dengan permasalahan yang sedang diteliti. Pada proses metodologi penelitian ini dilakukan dengan menggunakan tiga tahap penelitian yaitu pada Tahap pertama dilakukannya studi literatur yang diambil dari penelitian – penelitian terdahulu yang telah ada dan dapat menghasilkan sebuah proposal penelitian. Tahap kedua yaitu melakukan proses pemodelan dan perancangan terhadap sistem dengan cara pengumpulan data dan bagaimana cara dalam menganalisis data yang akan diterapkan terhadap algoritma yang digunakan. Tahap ketiga yaitu melakukan proses pemilihan *software* yang akan digunakan dalam mengimplementasi proses analisis data dan proses Testing hasil penelitian.

III.2. Kerangka Kerja Penelitian

Pada proses kerangka kerja ini adalah langkah-langkah yang akan dilakukan dalam upaya penyelesaian masalah yang akan dibahas. Adapun kerangka kerja dari penelitian ini dapat dilihat pada gambar III.1



Gambar III.1 Kerangka Kerja Penelitian

Adapun penjelasan dari kerangka kerja diatas yaitu :

III.2.1 Tahap Pertama

a. Mengumpulkan Data

Pada tahap ini dilakukanlah proses pengumpulan data yang akan digunakan dalam deteksi wajah. Proses pengumpulan datanya yaitu berupa pengambilan foto wajah guru yayasan Hamas stabat sebanyak 30 orang dengan jumlah gambar yang diambil pada setiap orangnya yaitu 10 gambar dan dilakukan proses augmentasi

sebanyak lima kali yang setiap gambarnya memiliki posisi yang sedikit berbeda. Secara keseluruhan gambar diambil pada tempat yang sama yaitu didalam ruangan yang tidak teralu gelap dan tidak terlalu terang dengan cahaya stabil. Ke stabilan cahaya dapat diatur dengan menggunakan alat bantu berupa lampu khusus yang biasanya digunakan untuk mengambil gambar secara stabil. Kemudian dalam ruangan tersebut disediakan sebuah kain yang fungsinya untuk baground, agar baground pada setiap gambar sama. Pengambilan gambar dilakukan menggunakan satu jenis kamera yang sama untuk keseluruhan gambar yaitu menggunakan kamera smartphone buatan Cina dengan merk OPPO A7 dengan kapasitas 13 mega pixel.

b. Memahami Studi Literatur

Pada tahapan ini dilakukannya proses pembelajaran terhadap *metode deep learning* terutama terutama difokuskan pada algoritma *convolutional neural network* (CNN) yang studi kasusnya difokuskan pada *face identification*, *Convolutional Neural Network* adalah algoritma yang efisien dan sangat mudah beradaptasi yang mudah dilatih karena parameter pelatihan lebih sedikit dan dapat digunakan untuk pemrosesan citra, pengenalan, dan klasifikasi, memudahkan peneliti untuk mengimplementasikan pencarian sekarang lebih cepat dan lebih akurat.

c. Analisis algoritma *convolutional neural network* (CNN)

Pada Tahapan Ini Dilakukannya Proses Analisis data terhadap data yang telah dikumpulkan dengan menerapkan algoritma *convolutional neural network* (CNN) menentukan metode identifikasi wajah baru pada pintu masuk Yayasan Hamas Stabat.

d. Preprocessing Data

Pada bagian ini, langkah selanjutnya adalah menginstal pustaka untuk digunakan, mengimpor gambar, menyempurnakan, membuat folder ekspor gambar, mengenali gambar, membagi gambar menjadi data pelatihan dan pengujian, mengubah ukuran gambar agar mudah dikenali oleh komputer, menggabungkan gambar ke dalam data pelatihan, dan beri label pada data pelatihan dan pengujian.

III.2.2 Tahap Kedua

a. Instalasi library

Pustaka yang akan diputuskan pada tahap ini adalah opencv (anaconda 3.8.3) dengan bahasa pemrograman Python versi 3.8. Kemudian instal perpustakaan di komputer atau laptop Anda. Untuk penelitian ini, penulis menggunakan HP merk HP dengan spesifikasi processor Intel(R) Core(TM) i3-6006U, CPU @ 2.00 GHz 2.00 GHz, RAM 4.00 GB (tersedia 3.89 GB), tipe sistem operasi 64 bit, x64 Saya menggunakan laptop saya. prosesor dasar.

b. Input Citra Wajah

Pada tahapan ini penulis menginput Input citra wajah guru yang datanya telah dikumpulkan sebelumnya menggunakan syntax dengan bahasa pemrograman *python*.

c. Membuat Folder Output Gambar

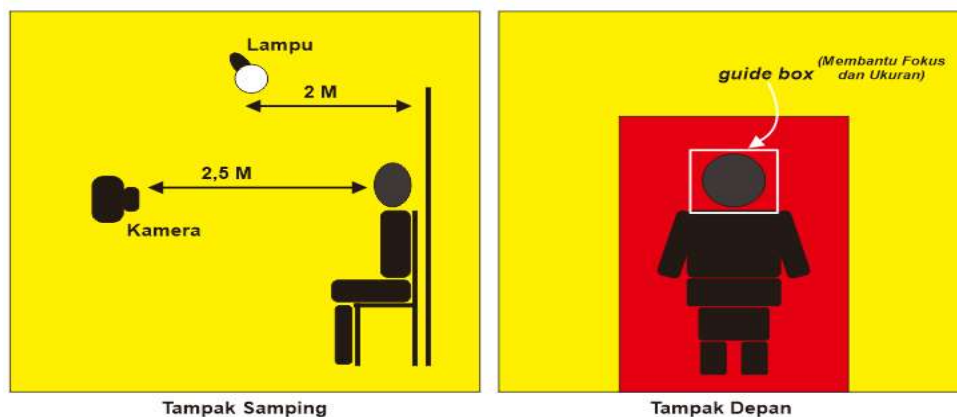
Membuat folder yang berguna untuk menyimpan hasil dari citra yang di inputkan.

d. Identifikasi Gambar

Pada bagian ini kita akan menamai gambar yang telah kita simpan pada folder yang telah kita buat sebelumnya, kemudian kita akan mengatur lebar dan tinggi yang diinginkan dan hasil dari perubahan tersebut akan disimpan.

e. Membagi Gambar menjadi Data Train dan Data Test

Pada tahap ini citra yang ada dibagi menjadi dua bagian yaitu data latih dan data uji. Data dibagi menjadi data latih hingga 80% dan data uji hingga 20%. Karena total citra yang diperoleh sebanyak 1500 citra, maka jumlah data untuk data kereta api sebanyak 1200 citra dan untuk data uji tidak kurang dari 300 citra. Pada Gambar III.2 berikut dapat dilihat gambaran umum komponen perlengkapan dan posisi menembak dari sisi kiri target taker dan di depan target taker. Perlengkapan yang digunakan pada bagian ini antara lain kamera dengan jarak lebih dari + 2,5m di depan subjek wajah dan penggunaan cahaya untuk stabilisasi cahaya. Gambar III.2 hanya memperlihatkan tampak samping dan tampak depan saja, karena background yang digunakan diletakkan pada dinding.



Gambar III.2 Proses Pengambilan Gambar Wajah

III.2.3 Tahap Ketiga

Setelah langkah pertama selesai, itu memasuki langkah ini yaitu. membuat modelnya. Model convolutional neural network (CNN) dipilih sebagai model.

III.2.4 Tahap Ke Empat

a. Testing dan Evaluasi Terhadap Data Train

Pada tahapan Testing gambar yang digunakan sebanyak 240 gambar sedangkan pada bagian evaluasi data gambar yang digunakan sebanyak 60 gambar yang akan dilakukan perhitungan nilai akurasi pada gambar tersebut.

III.2.5 Tahap Kelima

a. Hasil

Apa bila data train yang diperoleh telah sesuai maka lanjut pada tahap selanjutnya, namun apabila data train belum sesuai maka kembali lagi ke proses input.

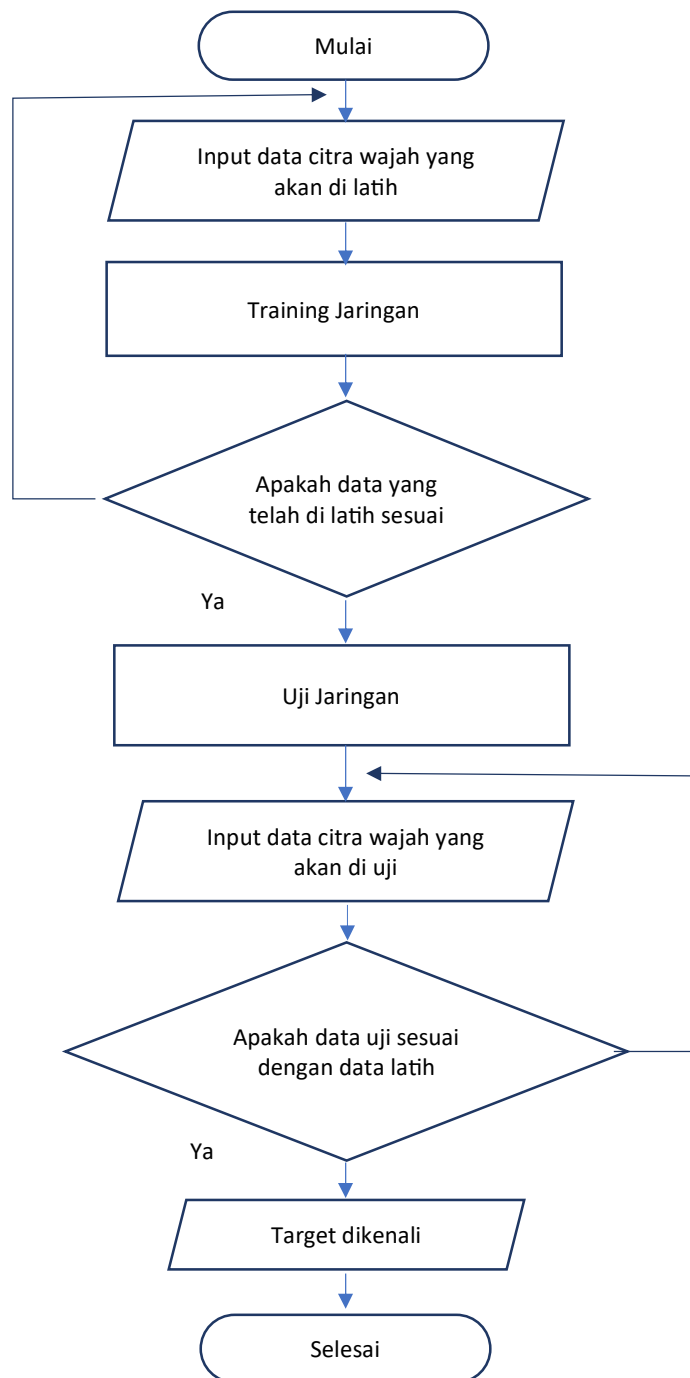
III.2.6 Tahap Ke Enam

Melakukan Testing Dengan Menggunakan library Opencv (Anaconda 3.8)
Pada tahap ini dilakukan Testing dengan menggunakan *opencv* (Anaconda 3.8) dengan data citra wajah guru Hamas stabat.

III.2.7 Tahap Ketujuh

Melihat hasil ujinya bagaimana? Apabila data yang diuji telah sesuai maka selesai namun apabila data uji belum selesai maka kembali ke tahapan input citra wajah yang akan di uji.

Berikut Akan disajikan *flowchart* dalam penelitian ini :



Gambar III.3 *Flowchart* Metode *Deeplearning* dalam training dan Testing deteksi wajah

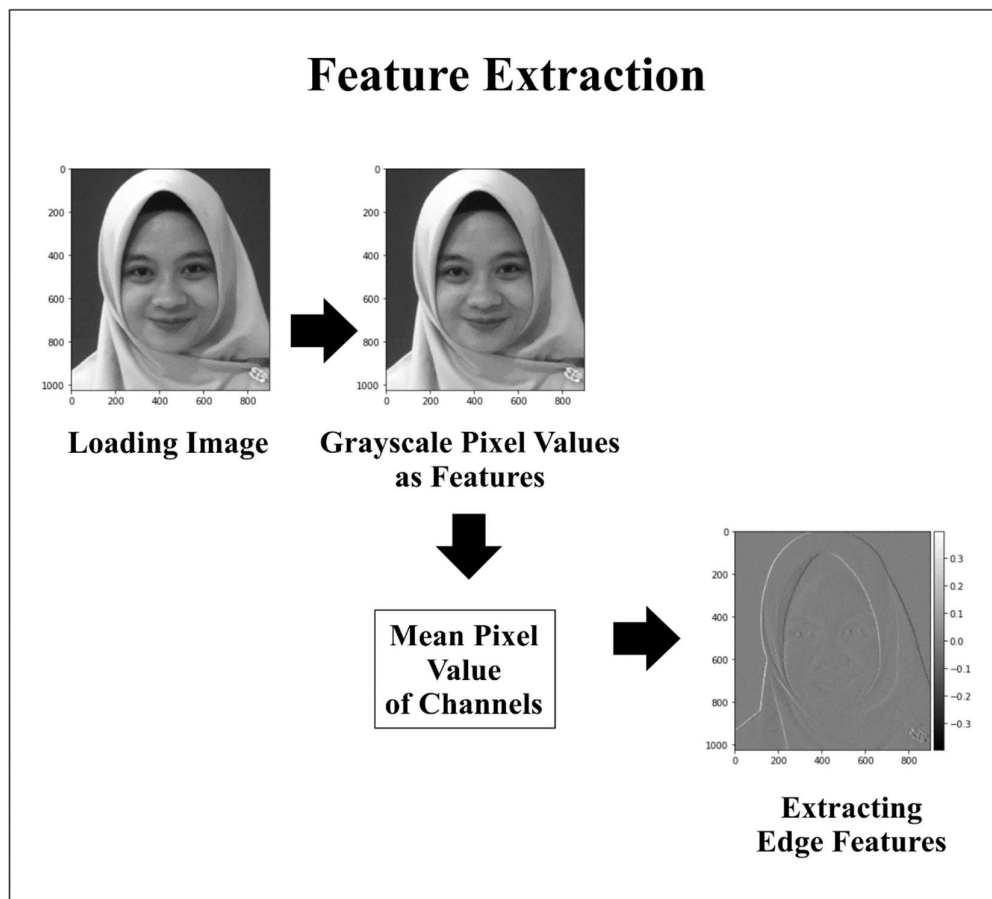
Gambar (III.5) merupakan flowchart yang menunjukkan alur kerja yang akan dilakukan pada sistem yang akan dibuat untuk pengenalan visa. Flowchart ini menjelaskan bahwa langkah pertama yang harus dilakukan adalah memasukkan data dari gambar seorang guru Hamas Foundation yang merupakan contoh wajah orang-orang di sekitar Hamas Foundation. Kemudian selesaikan pelatihan atau pelatihan online. Jika data yang dilatih memenuhi kriteria yang diharapkan, proses dilanjutkan untuk menyelidiki atau menguji jaringan, tetapi jika data tidak memenuhi kriteria, proses kembali ke tahap awal yaitu H. untuk memasukkan data dari citra wajah. Kemudian proses selanjutnya adalah pengujian jaringan atau network testing yaitu menginjeksikan data baru yang belum dilatih ke dalam jaringan, jika data tersebut cocok dengan data pelatihan maka dapat diketahui targetnya. Namun jika tidak, objek tersebut tidak dapat dikenali.

III.3 Implementasi

III.3.1 Implementasi Image Acquisition

Implementasi praproses diawali dengan mengumpulkan data image yang diambil dari data image wajah guru-guru di Yayasan hamas yang cara pengambilan gambarnya dilakukan didalam ruangan tertutup menggunakan bantuan cahaya lampu untuk membantu cahaya yang di nilai kurang. Pengambilan gambar dilakukan menggunakan kamera *Smart Phone* OPPO seri A7 yang memiliki spesifikasi kamera 13 MP, pengambilan gambar dilakukan dengan jarak 2,5 M dengan posisi cahaya lampu berada diatas objek. Kemudian file image di cropping agar data image lebih focus ke wajah dan selanjutnya image diberi nama sesuai

dengan orang yang di ambil gambarnya dan dimasukkan kedalam sebuah folder dataset. Kemudian dilakukan proses feature extraction pada salah satu gambar yang terdapat dalam data set dengan tujuan untuk dapat mengenali karakter serta ciri-ciri khusus dari sebuah objek. Pada gambar III. 4 dapat dilihat proses ekstraksi fitur yang dilakukan terhadap salah satu gambar yang terdapat pada data set. Pada gambar tersebut dapat dilihat tahapan dari ekstraksi fitur yaitu : Memuat gambar (Loading Gambar) , Mencari Nilai Piksel Grayscale sebagai Fitur setelah itu dilanjutkan dengan mencari nilai rata-rata pixel pada gambar, dan tahap terakhir adalah mengekstrak fitur tepi



III.3.2 Implementasi Praproses

Implementasi preprocessor menggunakan library OpenCV, Dlib dan Openface. Langkah selanjutnya adalah mengambil sampel posisi wajah dari situs web di atas dan memuatnya menggunakan pustaka openface. Kemudian library Dlib digunakan untuk melakukan pengenalan wajah menggunakan metode HOG. Setelah wajah didapatkan, pencarian fitur wajah dilakukan dengan menggunakan openface library dan dlib . Kemudian gunakan templat yang dimuat sebelumnya untuk menyempurnakan atau memodifikasi berbagai fitur wajah yang disimpan. Transformasi yang dilakukan meliputi transformasi translasi, scaling, rotasi, dan affine. Semua transformasi ini dilakukan sesuai kebutuhan untuk menghasilkan citra wajah rata-rata. Saat penyelarasan wajah dilakukan, ini menghasilkan kumpulan data gambar wajah yang tidak terhalang oleh latar belakang dan media. Kemudian ubah ukuran gambar menjadi 150x150 menggunakan pustaka OpenCV. Dan terakhir ubah gambar menjadi hitam putih menggunakan library OpenCV. Code Snippet III.1 Berikut adalah code snippet untuk mengimplementasikan data preprocessing menggunakan bahasa pemrograman Python.

```

predictor_model =
"model/shape_predictor_68_face_landmarks.dat"

# create a HOG face detector using the built-in dlib
class face_detector = dlib.get_frontal_face_detector()
face_aligner = openface.AlignDlib(predictor_model)

for file in imlisting:    im =
cv2.imread(path1+'\\'+file)

    # run the HOG face detector on the image data
detected_faces = face_detector(im, 1)

    print("Found {} faces in the image file
{}").format(len(detected_faces), file))

    # loop each faces in the photo
face_rect = list(detected_faces)
if len(face_rect)<1:

    # if no face is found, resize the image
print("No face is found, resizing image")
alignedFace = cv2.resize(im, (150, 150))
elif len(face_rect)>1:

    # if more than 1 face, find the biggest
face    area = [this.area() for this in
face_rect]    max_area = max(area)

    max_index = area.index(max_area)

```

Potongan Kode III.1 Implementasi praproses

III.3.3 Implementasi Augmentasi Dataset

Penyempurnaan yang dilakukan pada citra yang terkumpul terdiri dari dua penyempurnaan, yaitu penyempurnaan dengan prosedur filter dan penyempurnaan dengan prosedur transformasi geometri. Penambahan proses filtering menggunakan library OpenCV pada implementasinya, sedangkan penambahan transformasi geometri menggunakan library Keras Image Data Generator pada implementasinya. Di bawah ini adalah paragraf III.2 yang berisi pseudocode untuk menambahkan data dengan memfilter. Sedangkan Code Snippet III.3 berisi pseudocode yang berasal dari pengolahan data menggunakan transformasi geometri.

```
path = "dataset" imagelist = listdir(path)

For im in imagelist      image = loadImage(im)
    image_blur1 = image.addblur(kernel_size=5)
    image_blur2 = image.addblur(kernel_size=9)
    image_noise = image.addnoise(std=25)    image_sharpen =
image.sharpen(intensity=9)
    write(image_blur1, path)  write(image_blur2, path)
    write(image_noise, path)
    write(image_sharpen, path)
```

Potongan Kode III.2 Implementasi augmentasi data dengan filtering

```

path = "dataset"

imagelist = listdir(path)

Transformation = Datagen(zoom = 0.2, rotation =
30, translation = 0.2, shear = 0.2, flip = true)

For im in imagelist
    Image = loadImage(im)
    For numberofcopy < 5
        Transformation.apply(random)
        Transformation.save(path)

```

**Potongan Kode III.3 Implementasi Augmentasi Data Dengan
Transformasi Geometri**

III.3.4 Implementasi Arsitektur CNN (VGG16)

Menggunakan perpustakaan Keras, langkah-langkah berikut dijelaskan untuk mengimplementasikan arsitektur CNN (VGG16) dengan lapisan yang diaktifkan Leaky ReLU. Leaky ReLU digunakan untuk kinerja yang lebih baik seperti yang ditunjukkan oleh hasil tes pada Bab IV.

Pertama tentukan template dalam mode berurutan. Mode sekuensial keras memungkinkan pelapisan yang lebih fleksibel dan intuitif.

```
model = Sequential()
```

Pada model VGG16, layer pertama yang ditambahkan adalah convolutional layer dengan kedalaman neuron 64, ukuran kernel 3x3, padding 1 (sama) dan step 1. Kemudian tambahkan layer dengan kode berikut:

```
model.add(Conv2D(64, (3, 3), padding='same', input_shape=(m,n,1)))
```

Parameter Step dihilangkan (dibiarkan kosong) karena Keras men-default parameter Step menjadi 1. Sedangkan parameter Padding adalah sama. Format input adalah resolusi gambar input dan jumlah saluran. Setelah lapisan konvolusi, masukkan fungsi aktivasi nonlinear LeakyReLU online.

```
model.add(LeakyReLU())
```

Setelah fungsi aktivasi diimpor, tambahkan lapisan konvolusi berikut, dengan kedalaman neuron 64, core 3x3, padding 1 (sama) dan langkah 1 (kosong), bersama dengan fungsi aktivasi LeakyReLU.

```
model.add(Conv2D(64,(3,3), padding='same')) model.add(LeakyReLU())
```

Setelah melakukan 2 konvolusi, sampel menggunakan lapisan MaxPooling. Ukuran kolam 2x2.

```
model.add(MaxPooling2D(pool_size=(2, 2)))
```

Setelah lapisan komposit ditambahkan, blok konvolusi pertama selesai. Selanjutnya, tambahkan layer konvolusi dan pengelompokan untuk blok 2, blok 3, blok 4, dan blok 5, seperti yang ditunjukkan pada contoh berikut.

```

#block1

model.add(Conv2D(64,      (3,  3),  padding='same',
input_shape=(m,n,1))) model.add(LeakyReLU())

model.add(Conv2D(64, (3, 3), padding='same')) model.add(LeakyReLU())

model.add(MaxPooling2D(pool_size=(2, 2)))

#block2

model.add(Conv2D(128, (3, 3), padding='same')) model.add(LeakyReLU())

model.add(Conv2D(128, (3, 3), padding='same')) model.add(LeakyReLU())

model.add(MaxPooling2D(pool_size=(2, 2))) #block3 dan
seterusnya #...

```

Selanjutnya kita akan menambahkan 2 blok yang terhubung sepenuhnya (Dense) dengan pengaturan kedalaman saraf 4096. Namun sebelum itu, gambar perlu dibentuk ulang dari 2-D menjadi 1-D menggunakan kelas Flatten. Ini sebuah contoh.

```

model.add(Flatten())

#block6 model.add(Dense(4096))

model.add(LeakyReLU())

#block7 model.add(Dense(4096))

model.add(LeakyReLU())

```

Dan pada lapisan terakhir, ditambahkan lapisan yang terhubung sepenuhnya dengan kedalaman neuron berdasarkan jumlah lapisan dalam kumpulan data, dengan fungsi aktivasi softmax.

```

model.add(Dense(num_classes, activation='softmax'))

```

Kode berikut dijalankan ketika jaringan saraf selesai:

```
opt = SGD(lr=1e-3)
model.compile(loss='categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])
```

Parameter loss digunakan `categorical_crossentropy` karena dataset yang digunakan terdiri dari lebih dari 2 class. Pengoptimal yang digunakan adalah SGD dengan learning rate 0,001. Metrik adalah variabel yang digunakan untuk mengevaluasi kinerja jaringan.

Potongan Kode III.4 memuat potongan kode utuh dari arsitektur VGG16 yang dibangun.

```

model = Sequential()
model.add(Conv2D(64, (3, 3), padding='same',
input_shape=(m,n,1))) model.add(LeakyReLU())
model.add(Conv2D(64, (3, 3), padding='same')) model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(2, 2)) model.add(Conv2D(128, (3,
3), padding='same')) model.add(LeakyReLU())
model.add(Conv2D(128, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(256, (3,
3), padding='same')) model.add(LeakyReLU())
model.add(Conv2D(256, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(Conv2D(256, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(512, (3,
3), padding='same')) model.add(LeakyReLU())

```

```

model.add(Conv2D(512, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(Conv2D(512, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(2, 2))) model.add(Conv2D(512, (3,
3), padding='same')) model.add(LeakyReLU())
model.add(Conv2D(512, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(Conv2D(512, (3, 3), padding='same'))
model.add(LeakyReLU())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten()) model.add(Dense(4096))
model.add(LeakyReLU()) model.add(Dense(4096))
model.add(LeakyReLU())
model.add(Dense(num_classes, activation='softmax')) opt =
SGD(lr=1e-3)
model.compile(loss='categorical_crossentropy', optimizer=opt,
metrics=['accuracy'])

```

Potongan Kode III.4 Implementasi arsitektur VGG16

III.3.5 Arsitektur ResNet

ResNet berasal dari kata Residual Networks adalah jaringan saraf klasik. Model itu memenangkan Tantangan ImageNet pada tahun 2015. Premis dasar ResNet adalah memungkinkan untuk melatih jaringan saraf yang sangat dalam dengan lebih dari 150 lapisan. Sebelum pelatihan ResNet, jaringan saraf dalam sangat sulit karena masalah kehilangan gradien. Arsitektur Resnet menunjukkan

bahwa jaringan saraf lebih mudah dioptimalkan dan dapat mencapai akurasi kedalaman yang lebih tinggi (He et al., 2016). Gambar 10 menunjukkan lapisan yang tersisa. ResNet adalah solusi jaringan saraf yang dalam, semakin dalam pelatihannya, semakin kompleks. Kedalaman sangat penting selama pelatihan agar parameter atau neuron dapat mengingat atau menyimpan nilai pelatihan yang optimal. Agar pelatihan tidak membosankan dan terhindar dari error yang cukup besar, terdapat fungsi resnet yang meningkatkan identitas layer.

III.4 Ringkasan Bab III

Bab ini membahas kerangka kerja penelitian ini, yang meliputi antara lain mengumpulkan sampel citra wajah untuk penelitian dan kemudian memahami tinjauan literatur. Kemudian memfokuskan penelitian ini pada algoritma deep learning yaitu Convolutional Neural Network (CNN), kemudian menganalisis algoritma tersebut. Setelah menganalisis algoritme, algoritme digabungkan dengan model VGG 16 diuji keandalan pemrosesan citra, setelah itu dilanjutkan ke fase pra-pemrosesan, yaitu fase pemasangan alat yang diperlukan, yang dalam hal ini termasuk Perpustakaan. digunakan dalam penelitian ini OpenCV dan bahasa pemrograman Python 3. Kemudian pada langkah selanjutnya membuat model berupa coding yang berfungsi sebagai program pengenalan wajah pada penelitian ini. Berlatih dan coba di langkah selanjutnya. Langkah ini dilakukan dengan pengetikan dan pelatihan, jika wajah terdeteksi lanjutkan ke langkah berikutnya, tetapi bagaimana jika tidak maka mulailah pelatihan mengetik lagi. Setelah data pelatihan teridentifikasi, masuk ke data uji dengan memasukkan data uji, yaitu. H.

gambar wajah orang yang sama tetapi dengan posisi yang sedikit berbeda, kemudian data tersebut diuji. Jika wajah terdeteksi, langkah ini selesai, tetapi jika tidak, ulangi pengujian input data. Setelah wajah terdeteksi pada data test, bandingkan data training dengan data test yang sama, jika sama berarti wajah berhasil dideteksi, tetapi jika tidak maka wajah tidak dapat dideteksi. Langkah-langkah ini dilakukan untuk semua sampel citra wajah yang ada. Dan terakhir, kami membandingkan berapa banyak data pelatihan dan data uji yang sesuai dan berapa banyak yang tidak. Selain itu, tingkat keberhasilan pembelajaran mendalam dalam pengenalan wajah dapat ditentukan dalam penelitian ini. Proses persiapan diawali dengan fotografi dengan kamera smartphone seri OPPO A7 yang beresolusi 13 megapiksel dan menggunakan cahaya sebagai alat bantu pencahayaan yang inferior. Kemudian langkah selanjutnya adalah memberi nama file gambar yang terkumpul dan memotong gambar satu per satu agar gambar lebih fokus pada wajah. Langkah selanjutnya adalah melanjutkan ke langkah berikutnya, dimana model orientasi wajah diambil dari halaman web di atas dan dimuat dengan library pembuka. Selanjutnya implementasi arsitektur CNN (VGG16) menggunakan library Keras. Parameter categorical_crossentropy loss digunakan karena kumpulan data yang digunakan terdiri dari lebih dari dua kelas. Pengoptimal yang digunakan adalah SGD dengan learning curve 0.00.



BAB IV
HASIL

BAB IV

HASIL

IV.1 Persiapan Testing

Di bagian Persiapan Pengujian, kasus pengujian yang akan dijalankan serta lingkungan perangkat yang digunakan selama pengujian dijelaskan.

IV.1.1 Skenario Testing

Pada tugas akhir ini, pengujian akan dilakukan berdasarkan 3 skenario pengujian utama berikut ini:

1. Skenario Testing 1

Uji perbandingan menggunakan fungsi aktivasi ReLU dan LeakyReLU di VGG16 menggunakan data yang dihasilkan sendiri. Dibandingkan dengan sampel lain, VGG16 yang diuji memiliki akurasi yang lebih tinggi daripada Resnet, khususnya Resnet VGG 90 memiliki akurasi 50%, yang tercermin dalam klasifikasi botol plastik dan botol non-plastik.

2. Skenario Testing 2

Pada titik ini, uji arsitektur VGG16 terbaik dari hasil #1 untuk kumpulan data yang jumlah bingkainya bervariasi dengan akurasi system. ini juga dibuktikan dengan akurasi VGG 98% dan 80% sedangkan Resnet-50 memiliki akurasi 70% dan 50%

3. Skenario Testing 3

Pada fase ini, kami mempertimbangkan dampak penambahan data (dengan filter dan geometri transformasi) dan penyelarasan wajah pada akurasi sistem,

pada kumpulan data. Dalam hal ini, kami hanya akan menggunakan fungsi pemisahan lapisan di VGG 16.

IV.1.2 Lingkungan Testing

Selama fase ini, pengujian dilakukan dalam bahasa pemrograman Python menggunakan library Keras di Tensorflow. Untuk memastikan bahwa hasil pengujian yang diperoleh konsisten dan dapat direproduksi, nilai yang sama diberikan ke setiap nilai acak yang dihasilkan oleh sistem. Data teknis perangkat baik perangkat lunak (sistem operasi dan perangkat lunak) maupun perangkat keras disajikan pada Tabel IV.1.

Tabel IV.1 Spesifikasi perangkat

Perangkat Lunak	
OS	Windows 11 - 64 Bit
IDE	Spyder
Perangkat Keras	
CPU	Intel Core i3 – 1005G1
RAM	4 GB
GPU	Nvidia GTX 850M
GPU Memory	4 GB

IV.2 Hasil Testing

Bagian ini akan memuat hasil pengujian dari 3 kasus uji yang diidentifikasi pada subbab IV.1.1.

IV.2.1 Skenario Testing 1 : Testing Perbandingan Performa antara Fungsi

Aktivasi ReLU dan LeakyReLU.

Dataset yang digunakan dalam kasus uji kinerja sistem adalah dataset yang dihasilkan sendiri. Gambar IV.1 menunjukkan diagram alur pelatihan CNN saat

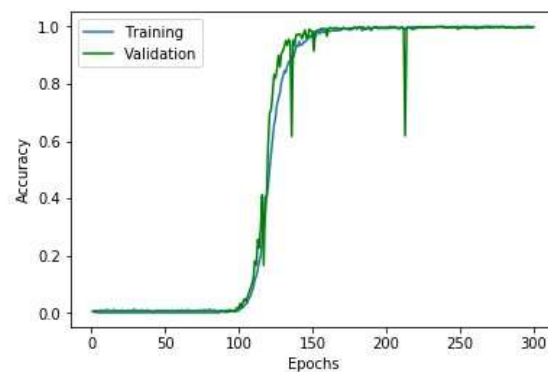
fungsi aktivasi yang digunakan adalah ReLU, sedangkan Gambar IV.2 menunjukkan proses pelatihan menggunakan fungsi aktivasi LeakyReLU.

Kedua fungsi aktivasi ini memiliki beberapa perbedaan. Fungsi aktivasi ReLU mengabaikan semua nilai negatif dari neuron sebelumnya, sehingga dapat menyebabkan neuron mati jika neuron terus menerima nilai negatif. Sedangkan fungsi aktivasi LeakyReLU selalu mengambil nilai negatif, namun nilai negatif tersebut akan diminimalkan.

Berikut adalah hasil pengujian penggunaan kedua fungsi tersebut:

1. Dengan menggunakan fungsi aktivasi ReLU, diagram proses pembentukan dapat dilihat pada Gambar IV.1 di bawah ini.

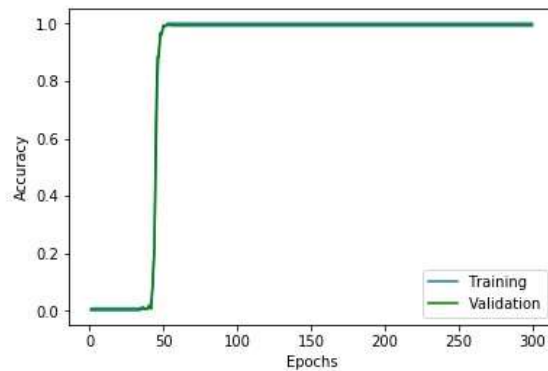
Nilai akurasi *testing* pada dataset yang diuji: 99,8%



Gambar IV.1 VGG16 menggunakan ReLU

2. Dengan menggunakan fungsi aktivasi LeakyReLU, grafik proses pembentukan dapat dilihat pada Gambar IV.2.

Nilai akurasi *testing* pada dataset yang diuji: 99.8%



Gambar IV.2 VGG16 menggunakan Leaky ReLU

IV.2.2 Skenario Testing 2: Testing Perbandingan Variasi Jumlah Citra

Dataset yang digunakan dalam kasus uji ini adalah dataset. Tes ini digunakan untuk menentukan apakah variasi jumlah gambar per lapisan mempengaruhi akurasi CNN. Variasi jumlah gambar yang digunakan adalah sebagai berikut:

Dataset Varian A (berisi 10 citra wajah dari setiap jenis), Dataset Varian B (berisi 40 citra wajah per kategori) dan Dataset Varian C (terdiri dari lebih dari 40 citra wajah per kategori).

1. Dataset Variasi A dan Testing.

Pemilihan data dilakukan dengan memilih dan menyimpan katalog dengan minimal 10 gambar. Kemudian dari semua kategori yang dipilih, jika jumlah gambar lebih dari 10, ambil 10 pertama dan hapus sisanya. Proses seleksi ini menghasilkan dataset yang konsisten dengan 10 gambar per kategori dan total 1.580 gambar (158 kategori). Nilai akurasi pengenalan wajah pada data uji adalah 13,5%. Diagram proses pembentukan ditunjukkan pada Gambar IV.3