

**PENGEMBANGAN SISTEM KUNCI PINTU OTOMATIS BERBASIS ESP8266
MENGUNAKAN CCTV RUMAH DAN ALGORITMA *MULTI-TASK*
CASCADE CONVOLUTIONAL NEURAL NETWORK UNTUK
IDENTIFIKASI PEMILIK RUMAH**

SKRIPSI

DEO PRANATA SILITONGA

171402065



**PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024**

PENGEMBANGAN SISTEM KUNCI PINTU OTOMATIS BERBASIS ESP8266
MENGUNAKAN CCTV RUMAH DAN ALGORITMA *MULTI-TASK*
CASCADE CONVOLUTIONAL NEURAL NETWORK UNTUK
IDENTIFIKASI PEMILIK RUMAH

SKRIPSI

Diajukan untuk melengkapi tugas dan memenuhi syarat memperoleh ijazah Sarjana
Teknologi Informasi

DEO PRANATA SILITONGA

171402065



PROGRAM STUDI TEKNOLOGI INFORMASI
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI
UNIVERSITAS SUMATERA UTARA
MEDAN
2024

PERSETUJUAN

Judul : Pengembangan Sistem Kunci Pintu Otomatis Berbasis
ESP8266 Menggunakan CCTV Rumah dan Algoritma
Multi-Task Cascade Convolutional Neural Network
Untuk Identifikasi Pemilik Rumah

Kategori : Skripsi

Nama Mahasiswa : Deo Pranata Silitonga

Nomor Induk Mahasiswa : 171402065

Program Studi : Sarjana (S-1) Teknologi Informasi

Fakultas : Ilmu Komputer dan Teknologi Informasi
Universitas Sumatera Utara

Medan, 11 Juli 2024

Komisi Pembimbing:

Pembimbing 2,



Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc.
NIP. 198603032010121004

Pembimbing 1,



Seniman S.Kom., M.Kom.
NIP. 198705252014041001

Diketahui/disetujui oleh

Program Studi S-1 Teknologi Informasi

Ketua,



Dedy Arisandi S.T., M.Kom.
NIP. 197908312009121002

PERNYATAAN

PENGEMBANGAN SISTEM KUNCI PINTU OTOMATIS BERBASIS ESP8266
MENGUNAKAN CCTV RUMAH DAN ALGORITMA *MULTI-TASK*
CASCADE CONVOLUTIONAL NEURAL NETWORK UNTUK
IDENTIFIKASI PEMILIK RUMAH

SKRIPSI

Saya mengakui bahwa skripsi ini adalah hasil karya saya sendiri, kecuali beberapa kutipan dan ringkasan yang masing-masing telah disebutkan sumbernya.

Medan, 11 Juli 2024

Deo Pranata Silitonga

171402065

UCAPAN TERIMA KASIH

Puji dan syukur penulis ucapkan kepada Tuhan Yang Maha Esa yang senantiasa memberikan berkat dan karunia-Nya sehingga penulis dapat menyelesaikan skripsi ini sebagai salah satu syarat untuk mendapat gelar Sarjana Komputer, pada Program Studi S1 Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara

Dalam menyelesaikan penulisan skripsi ini, penulis telah banyak mendapatkan bimbingan, dukungan, bantuan serta doa dari berbagai pihak. Adapun pada kesempatan ini penulis ingin mengucapkan terimakasih kepada:

1. Keluarga penulis, Alm A.D.H. Silitonga dan Ibu Julia Derwati Sianturi yang telah membesarkan serta memberikan kasih sayang, doa, masukan, bimbingan dan semangat mulai dari pendidikan hingga selesai tugas akhir ini, begitu juga kepada adik-adik penulis yaitu Ovan Silitonga, Egant Silitonga dan Sacha Silitonga yang telah memberikan doa dan dukungan.
2. Keluarga besar Alm A.D.H. Silitonga dan Ibu Julia Derwati Sianturi yang telah banyak membantu saya untuk dapat menyelesaikan tugas akhir ini.
3. Bapak Seniman S.Kom., M.Kom. selaku Dosen Pembimbing 1 dan Bapak Dr. Romi Fadillah Rahmat B.Comp.Sc., M.Sc. selaku Dosen Pembimbing 2 yang telah membimbing, memberikan saran dan kritik yang membangun, memotivasi dan memberikan semangat kepada penulis dalam penelitian serta penulisan skripsi ini.
4. Bapak Ainul Hizriadi S.Kom., M.Sc. dan Bapak Fahrurrozi Lubis B.IT., M.Sc.IT. selaku Dosen Penguji yang telah banyak membantu memberikan saran dan dukungan kepada penulis.
5. Ibu Dr. Maya Silvi Lydia, B.Sc., M.Sc. selaku Dekan Fasilkom-TI Universitas Sumatera Utara.
6. Bapak Dedy Arisandi S.T., M.Kom. selaku Ketua Program Studi S1 Teknologi Informasi Universitas Sumatera Utara.
7. Seluruh Dosen Program Studi S1 Teknologi Informasi yang telah memberikan ilmu yang bermanfaat bagi penulis selama masa perkuliahan.

8. Staff dan pegawai Fasilkom-TI Universitas Sumatera Utara yang membantu segala urusan administrasi dalam menyelesaikan skripsi.
9. Teman-teman dekat penulis yaitu Mersi Doknauli Sianturi, M Rizki Fatihah, Gilbert Sihura, M Farras Siraj, Nabila Sagita, M Taufik Baskoro, Ade Rizky, Ulwan yang telah membantu, mendukung, dan memberikan semangat dalam masa perkuliahan.
10. Teman-teman di PCS yaitu Fajar, Ibnu, Jackie, Rafif, Rizky, Dinul, Fakhri, Farras, Gilbert, Muharis, Preston, Rafid, Rezky, Prima dan Haqi yang menemani selama di kontrakan, masa perkuliahan hingga masa penulisan Tugas Akhir perkuliahan.
11. Teman-teman angkatan 2017 Teknologi Informasi yang telah membantu dan berjuang bersama penulis dalam menghadapi perkuliahan.

Penulis menyadari bahwa dalam penulisan skripsi ini masih terdapat kekurangan, oleh karena itu penulis mengharapkan kritik dan saran yang membangun untuk penyempurnaan skripsi ini.

Medan, 11 Juli 2024

Penulis

PENGEMBANGAN SISTEM KUNCI PINTU OTOMATIS BERBASIS ESP8266
MENGUNAKAN CCTV RUMAH DAN ALGORITMA *MULTI-TASK*
CASCADE CONVOLUTIONAL NEURAL NETWORK UNTUK
IDENTIFIKASI PEMILIK RUMAH

ABSTRAK

Salah satu tantangan utama dalam pengelolaan keamanan rumah adalah memastikan bahwa akses hanya diberikan kepada penghuni yang sah. Untuk mengatasi masalah ini, sistem dikembangkan untuk secara otomatis mengenali wajah penghuni yang terdaftar dalam database menggunakan algoritma *Multi-Task Cascade Convolutional Neural Network* (MTCNN). Integrasi dengan CCTV rumah memungkinkan sistem untuk memantau dan memproses gambar pengguna yang mendekati pintu, sehingga meningkatkan keakuratan dalam identifikasi. Selain itu, solusi pintu otomatis yang cerdas juga diharapkan dapat memberikan perlindungan tambahan bagi rumah dan penghuninya, meminimalkan risiko akses tidak sah, dan memberikan rasa percaya diri yang lebih besar kepada penghuni rumah. Tidak hanya meningkatkan keamanan, tetapi sistem ini juga diharapkan dapat meningkatkan kenyamanan dalam mengakses rumah, memberikan fleksibilitas dan kemudahan yang lebih besar bagi penghuni rumah dalam mengelola keamanan rumah mereka. Dengan menggunakan teknologi yang terjangkau dan mudah diakses seperti ESP8266 dan CCTV rumah, pengembangan sistem pintu otomatis berbasis ESP8266 menggunakan CCTV rumah dan algoritma *Multi-Task Cascade Convolutional Neural Network* untuk identifikasi pemilik rumah diharapkan dapat memberikan solusi yang efisien, terjangkau, dan andal untuk meningkatkan keamanan dan kenyamanan rumah secara keseluruhan.

Kata Kunci: Sistem Pintu Otomatis, ESP8266, CCTV Rumah, *Multi-Task Cascade Convolutional Neural Network* (MTCNN), Identifikasi Wajah, Keamanan Rumah.

*DEVELOPMENT OF AN ESP8266-BASED AUTOMATIC LOCK DOOR SYSTEM
USING HOME CCTV AND MULTI-TASK CASCADE CONVOLUTIONAL
NEURAL NETWORK ALGORITHM FOR HOME OWNER
IDENTIFICATION*

ABSTRACT

One of the primary challenges in home security management is ensuring that access is granted only to legitimate residents. To address this issue, a system is developed to automatically recognize the faces of registered occupants using Multi-Task Cascade Convolutional Neural Network (MTCNN) algorithms. Integration with home CCTV enables the system to monitor and process images of users approaching the door, thereby enhancing accuracy in identification. Additionally, the intelligent automatic door solution is expected to provide additional protection for the home and its occupants, minimizing the risk of unauthorized access and instilling greater confidence in residents. Beyond improving security, the system aims to enhance the convenience of accessing the home, offering greater flexibility and ease in managing home security for residents. By leveraging affordable and accessible technologies such as ESP8266 and home CCTV, the development of an ESP8266-based automatic door system using home CCTV and MTCNN algorithms for owner identification is anticipated to offer an efficient, cost-effective, and reliable solution to enhance overall home security and convenience.

Keywords: *Automatic Door System, ESP8266, Home CCTV, Multi-Task Cascade Convolutional Neural Network (MTCNN), Facial Identification, Home Security*

DAFTAR ISI

PERSETUJUAN	ii
PERNYATAAN	iii
UCAPAN TERIMA KASIH	iv
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
BAB 1 PENDAHULUAN	1
1.1. Latar Belakang	1
1.2. Rumusan Masalah	2
1.3. Batasan Masalah	3
1.4. Tujuan Penelitian	4
1.5. Manfaat Penelitian	4
1.6. Metodologi Penelitian	4
1.7. Sistematika Penulisan	5
BAB 2 LANDASAN TEORI	7
2.1. CCTV	7
2.2. Citra	8
2.3. Pengolahan Citra Digital	8
2.4. Multi-Task Convolution Neural Network	9
2.5. NodeMCU ESP8266	11

2.6. Relay	12
2.7. Solenoid Lock	13
2.8. Adaptor Dan DC Power Jack	13
2.9. Python	14
2.10. RTSP (<i>Real Time Streaming Protocol</i>)	14
2.11. Confusion Matrix	15
2.12. Parameter Pengujian Sistem	17
2.13. Penelitian Terdahulu	18
BAB 3 ANALISIS DAN PERANCANGAN SISTEM	21
3.1. Dataset Wajah	21
3.2. Arsitektur Umum	25
3.2.1. Akusisi gambar dan pembagian data	27
3.2.2. Preprocessing Data	29
3.2.3. Identifikasi gambar dengan model MTCNN	30
3.2.4. Pembuatan dan deployment model	31
3.2.5. Pemrosesan real-time oleh CCTV	32
3.2.6. Keputusan dan kontrol akses	34
BAB 4 IMPLEMENTASI DAN PENGUJIAN	36
4.1. Implementasi Sistem	36
4.1.1. Spesifikasi perangkat keras	36
4.1.2. Spesifikasi perangkat lunak	36
4.2. Penggunaan Data	36
4.3. Hasil Pengujian	41
BAB 5 KESIMPULAN DAN SARAN	44
5.1 Kesimpulan	44
5.2 Saran	44
DAFTAR PUSTAKA	46

DAFTAR TABEL

Tabel 2.1 <i>confusion matrix</i>	16
Tabel 2.2 Ringkasan penelitian terdahulu	19
Tabel 3.1 Data citra	21
Tabel 3.2 <i>Pseudocode</i> akuisi gambar dan pembagian <i>dataset</i>	28
Tabel 3.3 <i>Pseudocode preprocessing</i> data	29
Tabel 3.4 <i>Pseudocode</i> identifikasi gambar dengan model MTCNN	30
Tabel 3.5 <i>Pseudocode</i> pembuatan dan <i>deployment</i> model	31
Tabel 3.6 Pemrosesan <i>real-time</i> oleh CCTV	33
Tabel 3.7 <i>Pseudocode</i> keputusan dan kontrol akses	34
Tabel 4.1 Sampel pengujian	41

DAFTAR GAMBAR

Gambar 2.1 CCTV (<i>Closed Circuit Television</i>)	8
Gambar 2.2 P-Net	9
Gambar 2.3 R-Net	10
Gambar 2.4 O-Net	10
Gambar 2.5 NodeMCU ESP8266	11
Gambar 2.6 <i>Mapping pin</i> NodeMCU V3 Lolin	12
Gambar 2.7 <i>Relay</i>	12
Gambar 2.8 <i>Solenoid door lock</i>	13
Gambar 2.9 Bentuk fisik adaptor	14
Gambar 3.1 Arsitektur umum	27
Gambar 4.1 Data citra pemilik rumah	37
Gambar 4.2 Proses menjalankan program	38
Gambar 4.3 Tampilan video <i>live streaming</i> CCTV	38
Gambar 4.4 Peralatan pengujian	39
Gambar 4.5 Lokasi serta posisi CCTV	39
Gambar 4.6 Pengujian ketika mengidentifikasi pemilik rumah	40
Gambar 4.7 Pengujian ketika tidak mengidentifikasi pemilik rumah	41

BAB 1

PENDAHULUAN

1.1. Latar Belakang

Pengenalan wajah merupakan salah satu sistem biometrik yang menggunakan wajah suatu individu sebagai parameternya. Penelitian tentang pengenalan wajah sendiri sudah banyak dilakukan. Salah satu metode yang cukup efisien untuk mengenali wajah adalah *Multi-Task Cascade Convolutional Neural Network* (MTCNN). MTCNN atau *Multi-Task Cascaded Convolutional Neural Network* adalah perombakan dari gabungan model CNN dengan 3 lapisan jaringan yaitu P-Net, R-Net, O-Net yang menggunakan pengelompokan kandidat dan klasifikasi untuk mendapatkan hasil deteksi wajah yang efisien dan cepat (Du, 2020).

Penelitian yang dilakukan oleh (Pratama, 2019) mengemukakan bahwa dalam hasil pengujian secara otomatis menggunakan berbagai pola variasi dalam kumpulan data gambar Aberdeen, pengujian hingga 60 gambar dari 30 orang berbeda yang digunakan dalam Sistem Pencarian Pengenalan Wajah menggunakan *Multitasking Cascade Convolutional Neural Network* (MTCNN) dengan beberapa parameter citra yang mempengaruhi seperti variasi kecerahan (gelap, terang, sedang), variasi posisi objek (antara lain dapat diambil dekat dengan wajah, dari jarak dengan kemiringan yang berbeda), kemudian posisi grip (sedikit miring, ekspresi wajah dalam gambar objek yang sangat berbeda (marah, sedih, kesal, tersenyum, datar, cemberut), sistem pencarian pengenalan wajah menggunakan MTCNN (*Multi-Task Cascaded Convolutional Neural Network*) berhasil melakukan pengenalan wajah 100%.

Penelitian yang dilakukan oleh (Mugalu, et al., 2021) tentang pengenalan wajah sebagai metode otentikasi sebuah sistem berbasis *web* mengemukakan bahwa hasil yang diperoleh dari algoritma MTCNN memiliki tingkat akurasi dan presisi yang lebih

tinggi dari pada algoritma *Haar Cascade* dimana MTCNN mendapatkan akurasi 98% dan presisi 98%, sedangkan *Haar-Cascade* mendapatkan akurasi 74% dan presisi 74% dengan menggunakan dataset yang sama.

Penelitian yang dilakukan oleh (Rosa Andrie, 2022) dimana menguji kinerja dari sistem yang meliputi membandingkan teknik deteksi wajah antara CNN dengan *Haar-Cascade*, membandingkan model pengenalan wajah *facenet* dan *arcface* dan performa dari model pengenalan *facenet* dan *arcface*. Dari hasil pengujian metode deteksi wajah antara *Haar Cascade* dengan CNN dengan berbagai variasi dan kondisi dari data pengujian, untuk *Haar Cascade* diperoleh akurasi rata-rata sebesar 81,12%, sedangkan jika menggunakan CNN diperoleh akurasi sebesar 86,53%. Pada pengujian *encoding* pengenalan wajah dengan CNN, tingkat akurasi yang lebih tingkat tinggi diperoleh jenis model pengenalan *arcface* yakni 67,69% sedangkan *facenet* hanya memperoleh akurasi 66,46%.

Penelitian yang dilakukan oleh (Rafael *et al.*, 2020) tentang pemanfaatan *cloud computing* untuk pengenalan ekspresi wajah menggunakan *Amazon Web Services* (AWS). *Machine learning* yang dipakai pada penelitian ini adalah AWS DeepLens yang ada pada fitur AWS. Penelitian menghasilkan akurasi mencapai 76,16%.

Penelitian yang dilakukan oleh (Edwin jose, 2019) tentang sistem pengawasan berbasis pengenalan wajah menggunakan *facenet* dan MTCNN. data latih yang digunakan berupa foto, sementara untuk data uji menggunakan data foto dan video. Akurasi yang didapat mencapai 97%.

Oleh karena itu, berdasarkan latar belakang yang telah dituliskan sebelumnya. Penulis membuat sistem dengan benda yang sudah ada sebelumnya yaitu “PENGEMBANGAN SISTEM KUNCI PINTU OTOMATIS BERBASIS ESP8266 MENGGUNAKAN CCTV RUMAH DAN ALGORITMA *MULTI-TASK CASCADE CONVOLUTIONAL NETWORK* UNTUK IDENTIFIKASI PEMILIK RUMAH”.

1.2. Rumusan Masalah

Dengan berkembangnya teknologi IoT (*Internet of Things*), peningkatan keamanan rumah telah menjadi perhatian utama dalam masyarakat modern. Namun, pengembangan sistem keamanan yang efektif dan terjangkau masih menjadi tantangan. Oleh karena itu, penelitian ini bertujuan untuk memanfaatkan visual dari CCTV rumah sebagai sumber data untuk meningkatkan keamanan dengan mengembangkan sistem

penguncian pintu otomatis. Permasalahan utama yang ingin diselesaikan adalah bagaimana merancang dan mengimplementasikan sistem pintu otomatis yang dapat mengidentifikasi pemilik rumah secara akurat melalui analisis visual dari CCTV, sehingga memberikan lapisan keamanan tambahan bagi penghuni rumah tanpa mengorbankan kenyamanan penggunaan sehari-hari.

1.3. Batasan Masalah

Untuk memastikan fokus penelitian dan menghindari penyimpangan atau perluasan topik yang tidak diinginkan, berikut adalah batasan masalah yang ditetapkan untuk skripsi ini:

1. Sistem Pintu Otomatis

Penelitian ini akan terbatas pada pengembangan sistem pintu otomatis berbasis ESP8266 yang terhubung dengan CCTV rumah dan menggunakan algoritma *Multi-Task Cascade Convolutional Neural Network* (CNN) untuk identifikasi pemilik wajah. Tidak akan dibahas integrasi dengan jenis sistem keamanan lainnya.

2. Pengenalan Pemilik Wajah

Identifikasi pemilik wajah akan dilakukan berdasarkan data visual yang diperoleh dari IPCCTV Bardi, tanpa mempertimbangkan metode identifikasi lainnya seperti sidik jari atau pengenalan suara.

3. Pengujian pada Lingkungan Rumah

Pengujian sistem akan dilakukan dalam lingkungan rumah yang terkontrol, dengan pencahayaan yang cukup dan dengan jarak berkisar 1-2 meter.

4. Batasan Jumlah Pengguna

Jumlah pengguna yang terdaftar dalam sistem untuk proses identifikasi wajah akan dibatasi agar dapat memperoleh hasil pengujian yang lebih terfokus dan relevan.

Dengan mempertimbangkan batasan-batasan ini, diharapkan penelitian ini dapat mencapai tujuan dengan lebih mudah dan terarah, serta menghasilkan kontribusi yang signifikan dalam pengembangan sistem keamanan rumah yang cerdas.

1.4. Tujuan Penelitian

Tujuan utama penelitian ini adalah mengembangkan sistem pintu otomatis yang menggunakan visual dari CCTV rumah dan algoritma *Multi-Task Cascade Convolutional Neural Network* (MTCNN) untuk mengidentifikasi pemilik wajah dengan akurat, sehingga meningkatkan keamanan penguncian rumah dengan memanfaatkan teknologi modern.

1.5. Manfaat Penelitian

Manfaat penelitian ini adalah meningkatkan tingkat keamanan rumah tangga melalui pengembangan sistem pintu otomatis berbasis CCTV dengan identifikasi pemilik wajah. Dengan adanya sistem ini, penghuni rumah dapat merasa lebih aman karena akses masuk hanya akan diberikan kepada individu yang teridentifikasi sebagai pemilik wajah yang terdaftar, mengurangi risiko akses yang tidak sah dan potensialnya meningkatkan keselamatan rumah dan penghuninya.

1.6. Metodologi Penelitian

Metode penelitian yang akan dilakukan dalam penelitian ini meliputi tahapan-tahapan sebagai berikut:

1. Studi Literatur

Tahap pertama melibatkan pengumpulan data dan tinjauan pustaka terkait dengan pengenalan wajah, pengolahan citra, dan penggunaan ESP8266 dalam sistem IoT. Referensi akan diperoleh dari jurnal ilmiah, buku, artikel, dan sumber-sumber lain yang relevan untuk mendapatkan pemahaman yang mendalam tentang teknologi yang akan digunakan dalam penelitian ini.

2. Analisis Permasalahan

Tahap ini melibatkan analisis terhadap literatur yang telah dikumpulkan untuk memahami metode pengenalan wajah menggunakan algoritma *Multi-Task Cascade Convolutional Network* (MTCNN) dan implementasi ESP8266 dalam sistem pintu otomatis. Analisis ini akan membantu dalam menetapkan pendekatan yang paling sesuai untuk mencapai tujuan penelitian.

3. Perancangan Sistem

Berdasarkan hasil analisis, tahap ini akan memfokuskan pada perancangan arsitektur sistem pintu otomatis yang terkoneksi dengan CCTV dan algoritma MTCNN untuk identifikasi pemilik wajah. Sistem akan direncanakan mulai dari struktur perangkat keras ESP8266 hingga integrasi dengan algoritma pengenalan wajah.

4. Implementasi

Tahap ini melibatkan implementasi dari perancangan sistem yang telah disusun sebelumnya. Peneliti akan mengembangkan perangkat keras dan perangkat lunak yang diperlukan untuk membangun prototipe sistem pintu otomatis berbasis ESP8266 dan algoritma MTCNN.

5. Pengujian

Setelah implementasi selesai, sistem akan diuji untuk mengevaluasi kinerja pengenalan wajah dan fungsionalitas pintu otomatis. Pengujian akan dilakukan dengan berbagai skenario dan kondisi lingkungan untuk memastikan keandalan dan akurasi sistem.

6. Penyusunan Laporan

Tahap terakhir melibatkan penyusunan laporan berisi hasil penelitian, analisis, dan kesimpulan. Laporan ini akan mencakup semua tahapan yang telah dilakukan serta temuan-temuan yang diperoleh selama penelitian.

1.7. Sistematika Penulisan

Sistematika penulisan skripsi ini terdiri dari lima bab yang disusun sebagai berikut:

BAB 1: PENDAHULUAN

Bab ini memuat latar belakang penelitian yang menjelaskan konteks dan urgensi penelitian, rumusan masalah yang menjadi fokus utama penelitian, tujuan penelitian yang ingin dicapai, batasan masalah yang membatasi lingkup penelitian, manfaat penelitian bagi perkembangan ilmu pengetahuan dan masyarakat, metodologi penelitian yang digunakan, serta sistematika penulisan yang menguraikan struktur keseluruhan skripsi.

BAB 2: LANDASAN TEORI

Bab ini berisi rangkuman teori-teori yang relevan dengan penelitian, seperti konsep pengenalan wajah menggunakan algoritma *Multi-Task Cascade Convolutional Neural Network* (MTCNN), prinsip kerja ESP8266, dan konsep dasar CCTV. Selain itu, bab ini juga mencakup teori-teori terkait lainnya yang mendukung pemahaman terhadap penelitian.

BAB 3: ANALISIS DAN PERANCANGAN SISTEM

Bab ini menjelaskan secara rinci tentang analisis dan perancangan sistem pintu otomatis berbasis ESP8266 dan pengenalan wajah menggunakan algoritma MTCNN. Termasuk di dalamnya adalah arsitektur sistem, tahapan pre-processing data, proses training model, proses testing, dan perancangan antarmuka sistem.

BAB 4: IMPLEMENTASI DAN PENGUJIAN SISTEM

Pada bab ini, dilakukan implementasi dari analisis dan perancangan yang telah dibuat sebelumnya. Tahapan implementasi dari sistem pintu otomatis dan pengenalan wajah akan dijelaskan secara detail, serta hasil pengujian terhadap sistem yang telah dibangun akan disajikan.

BAB 5: KESIMPULAN DAN SARAN

Bab ini berisi kesimpulan dari hasil penelitian yang telah dilakukan, termasuk jawaban dari rumusan masalah, temuan-temuan penting, dan evaluasi terhadap kinerja sistem yang telah dibangun. Selain itu, bab ini juga berisi saran-saran untuk pengembangan penelitian selanjutnya guna meningkatkan kualitas sistem yang telah dibuat.

BAB 2

LANDASAN TEORI

2.1. CCTV

CCTV (*Closed Circuit Television*) adalah kamera video digital yang digunakan untuk mengirim sinyal ke layar monitor di suatu ruang atau tempat tertentu. Hal tersebut memiliki tujuan untuk dapat memantau situasi dan kondisi tempat tertentu. Pada umumnya CCTV seringkali digunakan untuk mengawasi area publik. Awalnya gambar dari kamera CCTV hanya dikirim melalui kabel ke sebuah ruang monitor tertentu dan dibutuhkan pengawasan secara langsung oleh operator/petugas keamanan dengan resolusi gambar yang masih rendah. Namun seiring dengan perkembangan teknologi yang sangat pesat seperti saat ini, banyak kamera CCTV yang telah menggunakan sistem teknologi yang modern. Sistem kamera CCTV digital saat ini dapat dioperasikan maupun dikontrol melalui *personal computer* atau telepon genggam, serta dapat dimonitor dari mana saja dan kapan saja selama ada komunikasi dengan internet maupun akses GPRS (Mardiana, 2018).

Sistem CCTV biasanya terdiri dari komunikasi *fixed (dedicated)* antara kamera dan monitor. Teknologi CCTV modern terdiri dari sistem terkoneksi dengan kamera yang bisa digerakkan (diputar, ditekuk, dan di-*zoom*) serta dapat dioperasikan dari jarak jauh lewat ruang kontrol, dan dapat dihubungkan dengan suatu jaringan baik LAN, *Wireless-LAN* maupun internet.



Gambar 2.1 CCTV (*Closed Circuit Television*) (<https://www.sourcesecurity.com>)

2.2. Citra

Citra atau gambaran adalah representasi dua dimensi dari objek yang sebenarnya dalam ruang koordinat *cartesian* x - y , di mana setiap koordinat mewakili satu elemen terkecil dari objek tersebut (Kulkarni, 2001). Fungsi citra adalah model matematika yang umum digunakan untuk analisis di mana citra dianggap sebagai fungsi dengan dua variabel, yang kemudian digunakan untuk mempertimbangkan berbagai analisis.

Citra merupakan hasil rekaman visual yang dihasilkan oleh kamera atau sensor. Definisi citra mencakup dua hal: pertama, gambaran dari objek yang dihasilkan oleh pantulan atau pembiasan sinar yang difokuskan melalui lensa atau cermin; kedua, gambaran rekaman objek (biasanya dalam bentuk foto) yang diperoleh melalui metode optik, elektro-optik, optik mekanik, atau elektronik. Biasanya, gambar direkam ketika radiasi elektromagnetik yang dipancarkan atau dipantulkan dari objek tidak langsung direkam pada film. Penginderaan jauh, di sisi lain, adalah bidang ilmu dan seni yang berkaitan dengan memperoleh informasi tentang objek, daerah, atau fenomena dengan menganalisis data yang diperoleh melalui alat tanpa kontak langsung dengan objek, daerah, atau fenomena yang sedang diteliti.

2.3. Pengolahan Citra Digital

Pengolahan citra atau *image processing*, adalah suatu sistem di mana proses dilakukan dengan memasukkan citra dan menghasilkan citra sebagai *output* (Basuki, 2005). Awalnya, pengolahan citra dilakukan untuk meningkatkan kualitas citra. Namun, dengan kemajuan komputasi yang ditandai oleh peningkatan kapasitas dan kecepatan komputer, serta kemampuannya untuk mengambil informasi dari citra, pengolahan citra menjadi terkait erat dengan bidang visi komputer.

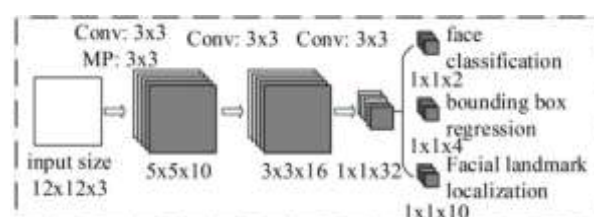
Seiring dengan perkembangan visi komputer itu sendiri, pengolahan citra memiliki dua tujuan utama. Pertama, adalah memperbaiki kualitas citra sehingga citra yang dihasilkan dapat menampilkan informasi secara jelas, yang dapat diinterpretasikan oleh manusia. Dalam hal ini, interpretasi terhadap informasi tetap dilakukan oleh manusia. Kedua, adalah mengekstraksi informasi yang menonjol dalam suatu citra, di mana hasilnya adalah informasi citra yang diperoleh manusia secara numerik. Dengan kata lain, komputer melakukan interpretasi terhadap informasi yang ada dalam citra melalui data yang dapat dibedakan dengan jelas.

2.4. Multi-Task Convolution Neural Network

MTCNN atau Multi-Task Cascaded Neural Networks adalah jaringan saraf yang digunakan untuk mendeteksi wajah dan *landmark* wajah pada gambar (Wang & Ma, 2018). MTCNN diterbitkan pada tahun 2016 oleh Zhang et al. MTCNN adalah suatu algoritma yang terdiri dari 3 jaringan saraf konvolusional, yaitu P-Net, R-Net, dan O-Net, yang mendeteksi *bounding box* pada wajah dan 5 *point face landmarks* pada gambar. Secara bertahap, setiap jaringan saraf meningkatkan hasil deteksi dengan melewati *input*-nya melalui CNN dan diikuti oleh *Non-Maximum Suppression* (NMS) yaitu metode yang mengurangi jumlah *bounding box* (Mühler, 2018).

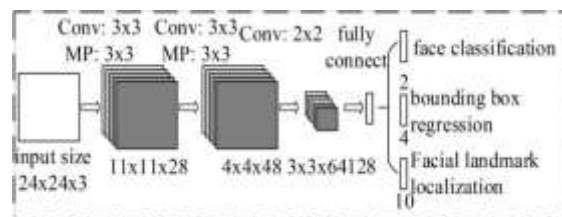
Berikut adalah cara kerja masing-masing tahap pada MTCNN:

1. *Proposal Network* (P-Net): Tahap pertama dari algoritma MTCNN adalah P-Net, yang menghasilkan sekumpulan kotak pembatas kandidat yang mungkin berisi wajah. P-Net mengambil citra masukan dan menerapkan serangkaian filter konvolusional untuk menghasilkan sekumpulan peta fitur. Peta fitur ini kemudian diproses oleh sekumpulan lapisan yang terhubung penuh untuk memprediksi probabilitas wajah yang ada di setiap wilayah citra. P-Net juga meregresi koordinat kotak pembatas di sekitar wajah yang terdeteksi.



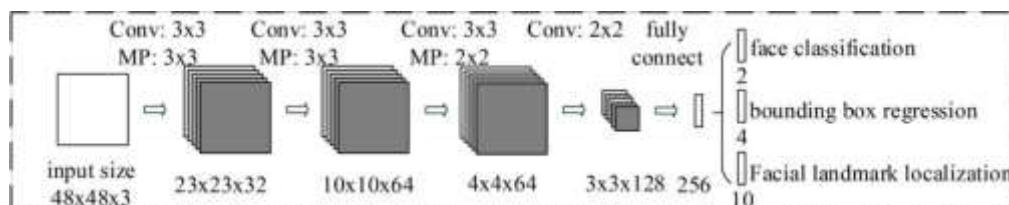
Gambar 2.2 P-Net (Zhang, et al., 2016)

2. *Refinement Network (R-Net)*: Tahap kedua dari algoritma MTCNN adalah R-Net, yang menyempurnakan kotak pembatas kandidat yang dihasilkan oleh P-Net. R-Net mengambil kotak pembatas kandidat dan memotong daerah yang sesuai dari gambar masukan. Daerah yang dipotong ini kemudian diubah ukurannya ke ukuran tetap dan melewati serangkaian lapisan konvolusional dan terhubung penuh untuk mengklasifikasikan setiap kotak pembatas sebagai wajah atau bukan wajah. R-Net juga meregresi koordinat kotak pembatas untuk menyempurnakan lokasi wajah yang terdeteksi.



Gambar 2.3 R-Net (Zhang, et al., 2016)

3. *Output Network (O-Net)*: Tahap akhir dari algoritma MTCNN adalah O-Net, yang selanjutnya menyempurnakan kotak pembatas dan mengekstraksi titik acuan wajah. O-Net mengambil kotak pembatas yang telah disempurnakan dari R-Net dan memotong daerah yang sesuai dari gambar masukan. Daerah yang dipotong ini kemudian diubah ukurannya ke ukuran tetap dan melewati serangkaian lapisan konvolusional dan terhubung penuh untuk mengklasifikasikan setiap kotak pembatas sebagai wajah atau bukan wajah. O-Net juga meregresi koordinat kotak pembatas untuk selanjutnya menyempurnakan lokasi wajah yang terdeteksi dan mengekstraksi koordinat lima titik acuan wajah (yaitu, dua mata, hidung, dan mulut).



Gambar 2.4 O-Net (Zhang, et al., 2016)

2.5. NodeMCU ESP8266

NodeMCU adalah sebuah platform *Internet of Things* (IoT) yang bersifat *open-source*, terdiri dari perangkat keras yang menggunakan *System On Chip* ESP8266 dari *Esperessif System*. NodeMCU dapat dianggap sebagai *board* Arduino yang terhubung dengan ESP8266. *Platform* ini telah mengemas ESP8266 ke dalam sebuah board yang telah terintegrasi dengan berbagai fitur layaknya mikrokontroler, serta memiliki kemampuan akses terhadap *WiFi* dan chip komunikasi berupa *USB to serial*. Dalam pemrograman, hanya diperlukan kabel data USB.



Gambar 2.5 NodeMCU ESP8266 (<https://sariteknologi.com>)

Karena sumber utama dari NodeMCU adalah ESP8266, khususnya seri ESP-12 seperti ESP-12E, fitur-fitur yang dimilikinya akan mirip dengan ESP-12. Beberapa fitur yang tersedia antara lain:

1. 10 *Port* GPIO dari D0 hingga D10
2. Fungsionalitas PWM
3. Antarmuka I2C dan SPI
4. Antarmuka 1 *Wire*
5. ADC

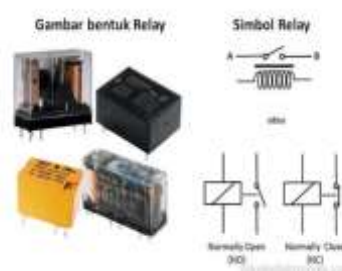


Gambar 2.6 Mapping pin NodeMCU V3 Lolin (createc3d.com)

2.6. Relay

Relay adalah Saklar (*Switch*) yang dioperasikan secara listrik dan merupakan komponen *Electromechanical* (Elektromekanikal) yang terdiri dari 2 bagian utama yakni elektromagnet (*Coil*) dan mekanikal (seperangkat kontak saklar/*Switch*). *Relay* menggunakan prinsip elektromagnetik untuk menggerakkan kontak saklar sehingga dengan arus listrik yang kecil (*low power*) dapat menghantarkan listrik yang bertegangan lebih tinggi. Sebagai contoh, dengan *Relay* yang menggunakan Elektromagnet 5V dan 50 mA mampu menggerakkan *Armature Relay* (yang berfungsi sebagai saklarnya) untuk menghantarkan listrik 220V 2A.

Relay dibutuhkan dalam rangkaian elektronika sebagai eksekutor sekaligus *interface* antara beban dan sistem kendali elektronik yang berbeda sistem *power supply*nya. Secara fisik antara saklar atau kontaktor dengan elektromagnet relay terpisah sehingga antara beban dan sistem kontrol terpisah. Bagian utama relay elektro mekanik adalah sebagai berikut. Kumparan elektromagnet Saklar atau kontaktor *Swing Armatur Spring* (Pegas) (Turang, 2015).



Gambar 2.7 Relay (<https://teknikelektronika.com>)

2.7. Solenoid Lock

Solenoid door lock adalah salah satu solenoid yang difungsikan khusus sebagai solenoid untuk pengunci pintu secara elektronik (Saghoa, Sompie, & Tulung, 2018). Solenoid ini mempunyai dua sistem kerja, yaitu *Normaly Close* (NC) dan *Normaly Open* (NO). Perbedaannya adalah jika cara kerja solenoid NC apabila diberi tegangan, maka solenoid akan memanjang (tertutup). Dan untuk cara kerja dari *Solenoid* NO adalah kebalikannya dari *solenoid* NC. Biasanya kebanyakan *solenoid door lock* membutuhkan *input* atau tegangan kerja 12V DC, tetapi ada juga *solenoid door lock* yang hanya membutuhkan input tegangan 5V DC sehingga dapat langsung bekerja dengan tegangan *output* dari pin IC digital. Namun jika menggunakan *solenoid door lock* yang 12V DC. Pada kondisi normal *solenoid* dalam posisi tuas memanjang / terkunci. Jika diberi tegangan tuas akan memendek/terbuka. Solenoid ini bisa digabungkan dengan sistem pengunci elektrik berbasis RFID dan *password*. Cocok dipakai untuk pengunci pintu ataupun locker/lemari. Membutuhkan *power supply* 12V dan sebuah relay untuk mengaktifkannya.



Gambar 2.8 *Solenoid door lock* (<https://www.electroschematics.com>)

2.8. Adaptor Dan DC Power Jack

Adaptor adalah perangkat elektronik yang digunakan dalam rangkaian yang dapat merubah tegangan besar menjadi tegangan kecil, atau AC ke DC. Adaptor juga merupakan bagian utama pada peralatan elektronik. Adaptor ini digunakan untuk mengecilkan tegangan AC 22V menjadi tegangan AC kecil 3-12V tergantung kebutuhan perangkat. Berdasarkan sistem kerjanya, ada dua jenis adaptor: adaptor sistem *switching* dan adapter sistem *step-down* (Riskiono, Pamungkas, & Arya, 2020).

Dalam prinsip kerjanya kedua sistem adaptor tersebut berbeda, adaptor *step-down* menggunakan teknik induksi medan magnet, komponen utamanya adalah kawat email yang di lilit pada teras besi, terdapat 2 lilitan yaitu lilitan primer dan lilitan skunder, ketika listrik masuk kelilitan primer maka akan terjadi induksi pada kawat

email sehingga akan terjadi gaya medan magnet pada teras besi kemudian akan menginduksi lilitan sekunder. Sedangkan sistem *switching* menggunakan teknik transistor maupun IC *switching*, adaptor ini lebih baik dari pada adaptor teknik induksi, tegangan yang di keluarkan lebih stabil dan komponennya suhunya tidak terlalu panas sehingga mengurangi tingkat resiko kerusakan karena suhu berlebih, biasanya regulator ini digunakan pada peralatan elektronik digital.



Gambar 2.9 Bentuk fisik adaptor

2.9. Python

Menurut pengertian dari Python Software Foundation (2016), Python adalah bahasa pemrograman interpretatif, berorientasi objek dan semantik yang dinamis. Python memiliki *high-level* struktur data, *dynamic typing* dan *dynamic binding*. Python memiliki sintaks sederhana dan mudah dipelajari untuk penekanan pada kemudahan membaca dan mengurangi biaya perbaikan program. Python mendukung modul dan paket untuk mendorong kemodularan program dan *code reuse*. Interpreter Python dan *standard library*-nya tersedia secara gratis untuk semua *platform* dan dapat secara bebas disebar. Bahasa pemrograman ini dibuat oleh Guido van Rossum dari Belanda pada tahun 1992.

2.10. RTSP (*Real Time Streaming Protocol*)

Real Time Streaming Protocol (RTSP) adalah sebuah protokol level aplikasi untuk kontrol atas pengiriman data dengan sifat real-time. RTSP menyediakan kerangka *extensible* untuk mengaktifkan kendali pada pengiriman data *real-time*, seperti *audio* dan *video*, dengan menggunakan *Transmission Control Protocol* (TCP) atau *User Data Protocol* (UDP) (Nuryaman, 2019). Proses kerja *Real Time Streaming Protocol* adalah protokol level aplikasi yang bertujuan untuk menyediakan sebuah protokol yang kuat

untuk multimedia *streaming* satu ke banyak aplikasi secara *unicast* dan *multicast* serta untuk mendukung interoperabilitas antara *client* dan *server* dari vendor yang berbeda. RTSP dianggap lebih dari kerangka daripada protokol. RTSP didesain untuk bekerja di atas RTP untuk mengontrol dan menyampaikan konten secara *real-time*.

RTSP merupakan protokol yang digunakan pada layanan *streaming*, teknologi *streaming* itu sendiri merupakan sebuah teknologi yang mampu mengompresi atau menyusutkan ukuran *file* audio dan video agar mudah ditransfer melalui jaringan data. Protokol ini digunakan untuk menetapkan dan mengendalikan sesi media antara dua titik ujungnya. Klien dari *server* media mengeluarkan perintah seperti VCR, seperti *play* dan *pause*, untuk mendukung kendali waktu nyata dari berkas media yang dijalankan *server*.

2.11. Confusion Matrix

Confusion matrix menurut Han dan Kamber dapat diartikan sebagai suatu alat yang memiliki fungsi untuk melakukan analisis apakah *classifier* tersebut baik dalam mengenali tuple dari kelas yang berbeda. Nilai dari *True Positive* dan *True Negative* memberikan informasi ketika *classifier* dalam melakukan klasifikasi data bernilai benar, sedangkan *False Positive* dan *False Negative* memberikan informasi ketika *classifier* salah dalam melakukan klasifikasi data (Fibrianda & Bhawiyuga, 2018).

Berdasarkan jumlah keluaran kelasnya, sistem dapat dibagi menjadi 4 (empat) jenis yaitu klasifikasi *binary*, *multi-class*, *multi-label* dan *hierarchical*. Pada klasifikasi *binary*, data masukan dikelompokkan ke dalam salah satu dari dua kelas. Jenis klasifikasi ini merupakan bentuk klasifikasi yang paling sederhana dan banyak digunakan. Contoh penggunaannya antara lain dalam sistem yang melakukan deteksi orang atau bukan, sistem deteksi kendaraan atau bukan, dan sistem deteksi pergerakan atau bukan.

Sementara itu, pada bentuk klasifikasi *multi-class*, data masukan diklasifikasikan menjadi beberapa kelas. Sebagai contoh sistem yang dapat mengklasifikasikan jenis kendaraan seperti sepeda, sepeda motor, mobil, bus, truk, dan sebagainya. Bentuk klasifikasi *multi-label* pada dasarnya sama dengan *multi-class* dimana data dikelompokkan menjadi beberapa kelas, namun pada klasifikasi *multi-label*, data dapat dimasukkan dalam beberapa kelas sekaligus. Bentuk klasifikasi yang terakhir adalah *hierarchical*. Data masukan dikelompokkan menjadi beberapa kelas,

namun kelas tersebut dapat dikelompokkan kembali menjadi kelas-kelas yang lebih sederhana secara hirarkis.

Contohnya dalam penelitian ini, arah pergerakan dikelompokkan menjadi 12 arah pergerakan yang tentunya dapat disederhanakan menjadi 4 arah. Pada pengukuran kinerja menggunakan *confusion matrix*, terdapat 4 (empat) istilah sebagai representasi hasil proses klasifikasi. Keempat istilah tersebut adalah *True Positive* (TP), *True Negative* (TN), *False Positive* (FP) dan *False Negative* (FN). Nilai *True Negative* (TN) merupakan jumlah data negatif yang terdeteksi dengan benar, sedangkan *False Positive* (FP) merupakan data negatif namun terdeteksi sebagai data positif. Sementara itu, *True Positive* (TP) merupakan data positif yang terdeteksi benar. *False Negative* (FN) merupakan kebalikan dari *True Positive*, sehingga data positif, namun terdeteksi sebagai data negatif. Pada jenis klasifikasi *binary* yang hanya memiliki 2 keluaran kelas, *confusion matrix* dapat disajikan seperti pada Tabel 2.1.

Tabel 2.1 *confusion matrix*

Kelas	Terklasifikasi Positif	Terklasifikasi Negatif
Positif	TP (True Positive)	FN (False Negative)
Negatif	FP (False Positive)	TN (True Negative)

Berdasarkan nilai *True Negative* (TN), *False Positive* (FP), *False Negative* (FN), dan *True Positive* (TP) dapat diperoleh nilai akurasi, presisi dan *recall*. Nilai akurasi menggambarkan seberapa akurat sistem dapat mengklasifikasikan data secara benar. Dengan kata lain, nilai akurasi merupakan perbandingan antara data yang terklasifikasi benar dengan keseluruhan data. Nilai akurasi dapat diperoleh dengan Persamaan 2.3. Nilai presisi menggambarkan jumlah data kategori positif yang diklasifikasikan secara benar dibagi dengan total data yang diklasifikasi positif. Presisi dapat diperoleh dengan Persamaan 2.4 Sementara itu, *recall* menunjukkan berapa persen data kategori positif yang terklasifikasikan dengan benar oleh sistem. Nilai *recall* diperoleh dengan persamaan berikut.

$$Akurasi = (TP + TN) / (TP + TN + FP + FN)) * 100\% \dots\dots\dots (2.3)$$

$$Precision = (TP / (TP + FP)) * 100\%. \dots\dots\dots (2.4)$$

$$Recall = (TP / (TP + FN)) * 100\%. \dots\dots\dots (2.5)$$

Dimana:

- TP adalah *True Positive*, yaitu jumlah data positif yang terklasifikasi dengan benar oleh sistem.
- TN adalah *True Negative*, yaitu jumlah data negatif yang terklasifikasi dengan benar oleh sistem.
- FN adalah *False Negative*, yaitu jumlah data negatif namun terklasifikasi salah oleh sistem.
- FP adalah *False Positive*, yaitu jumlah data positif namun terklasifikasi salah oleh sistem.

2.12. Parameter Pengujian Sistem

Pengujian sistem pintu otomatis berbasis ESP8266 menggunakan CCTV rumah dan Algoritma *Multi-Task Cascade Convolutional Neural Network* (MTCNN) untuk identifikasi pemilik rumah memerlukan penetapan parameter yang cermat untuk mengevaluasi kinerja sistem secara efektif. Pengujian dilakukan membagi jumlah pengujian berhasil/jumlah sampel pengujian selanjutnya dikalikan dengan 100 persen. Berikut ini rumus yang digunakan untuk mengetahui hasil akurasi pengenalan wajah:

$$Akurasi = \frac{\text{Jumlah Data Uji Yang Sukses}}{\text{Total Data Uji}} \times 100\%$$

Selanjutnya, kesalahan sistem dalam mendeteksi pemilik rumah dalam pengenalan identitas dari dataset yang telah dilakukan pelatihan (*training*) sebelumnya maupun pengujian pengenalan identitas wajah yang tidak terdapat dalam dataset dapat dilakukan pengukuran parameter pengujian dengan menggunakan rumus *False Acceptance Rate (FAR)*, sebagai berikut:

$$FAR = \frac{\text{Jumlah Data Uji Yang Gagal}}{\text{Total Data Uji}} \times 100\%$$

2.13. Penelitian Terdahulu

Beberapa penelitian terdahulu telah dilakukan untuk memperbaiki deteksi dan pengenalan wajah menggunakan berbagai metode Pratama (2019). Dari hasil pengujian yang dilakukan secara otomatis menggunakan beberapa model variasi pengujian citra dataset Aberdeen sebanyak 60 citra dari 30 orang berbeda yang digunakan pada sistem riset pengenalan wajah menggunakan MTCNN dengan parameter citra yang mempengaruhi seperti variasi pencahayaan, variasi posisi objek, kemudian posisi yang diambil serta ekspresi wajah pada citra objek, sistem riset berhasil melakukan pengenalan wajah sebesar 100%.

Edwin Jose (2019) tentang sistem pengawasan berbasis pengenalan wajah menggunakan *facenet* dan MTCNN. data latih yang digunakan berupa foto, sementara untuk data uji menggunakan data foto dan video. Akurasi pengenalan yang didapat mencapai 97%.

Rafael (2020) memanfaatkan cloud computing untuk pengenalan ekspresi wajah menggunakan *Amazon Web Services* (AWS) .Machine learning yang dipakai pada penelitian ini adalah *AWS DeepLens* yang ada pada fitur AWS dan mendapatkan akurasi 76,16%.

Mugalu (2021) mendapati algoritma MTCNN memiliki tingkat akurasi dan presisi yang lebih tinggi dari pada algoritma *Haar Cascade* dimana MTCNN mendapatkan akurasi 98% dan presisi 98%, sedangkan *Haar Cascade* mendapatkan akurasi 74% dan presisi 74% dengan menggunakan dataset yang sama.

Di sisi lain Rosa Andrie (2022) dari hasil pengujian metode deteksi wajah antara *Haar Cascade* dengan CNN dengan berbagai variasi dan kondisi dari data pengujian, untuk *Haar Cascade* diperoleh akurasi rata-rata sebesar 81,12%, sedangkan jika menggunakan CNN diperoleh akurasi sebesar 86,53%. Pada pengujian *encoding* pengenalan wajah dengan CNN, tingkat akurasi yang lebih tinggi diperoleh jenis model pengenalan *arcface* yakni 67,69% sedangkan *facenet* hanya memperoleh akurasi 66,46%.

Tabel 2.2 Ringkasan penelitian terdahulu

No.	Peneliti	Tahun	Metode	Hasil
1	Yovi Pratama, Marrylinteri Istoningtyas, Errisya Rasywir	2019	MTCNN	Sistem riset pengenalan wajah menggunakan MTCNN dengan parameter citra yang mempengaruhi seperti variasi pencahayaan, variasi posisi objek, kemudian posisi yang diambil serta ekspresi wajah pada citra objek, sistem riset berhasil melakukan pengenalan wajah sebesar 100%.
2	Edwin Jose <i>et.al</i>	2019	MTCNN, facenet	Data latih yang digunakan berupa foto, sementara untuk data uji menggunakan data foto dan video. Deteksi akurasi yang didapat mencapai 97%.
3	Gregorius Rafael, Hendra Kusuma, Tasripan	2020	AWS Deeplens	Memanfaatkan cloud computing untuk pengenalan ekspresi wajah menggunakan Amazon Web Services (AWS). Machine learning yang dipakai pada penelitian ini adalah AWS DeepLens yang ada pada fitur AWS dan mendapatkan akurasi 76,16%.
4	Ben Wycliff Mugalu, Rodrick Calvin Wamala, Jonathan Serugunda, Andrew Katumba	2021	MTCNN dan Haar Cascade	MTCNN mendapatkan akurasi 98% dan presisi 98%, sedangkan Haar-Cascade mendapatkan akurasi 74% dan presisi 74%.
5	Rosa Andrie Asmara, Muhammad Ridwan, Gunawan	2022	CNN dan Haar Cascade	Haar Cascade diperoleh akurasi rata-rata sebesar 81,12%, CNN diperoleh akurasi sebesar 86,53%. Pada pengujian encoding pengenalan wajah dengan CNN, tingkat akurasi yang lebih tinggi tingkat tinggi diperoleh jenis model

No.	Peneliti	Tahun	Metode	Hasil
	Budi P, Anik Nur Handayani			pengenalan arcface yakni 67,69% sedangkan facenet hanya memperoleh akurasi 66,46%.

Perbedaan penelitian yang dilakukan penulis dengan penelitian terdahulu adalah penulis mengimplementasi sistem pengenalan wajah dengan algoritma MTCNN melalui CCTV yang tersedia.

BAB 3





ANALISIS DAN PERANCANGAN SISTEM

3.1. Dataset Wajah

Dalam penelitian ini, hal yang utama adalah pengumpulan data yang disebut juga sebagai dataset berupa citra pemilik rumah yang diambil. Data citra pemilik rumah yang digunakan dalam penelitian ini diambil langsung dari CCTV. Dataset dapat dilihat pada Tabel 3.1.




Tabel 3.1 Data citra

No	Gambar	Label
1		Deo

No	Gambar	Label
2		Deo
3		Deo
4		Deo
5		Deo

No	Gambar	Label
6		Frans
7		Frans
8		Frans

No	Gambar	Label
9		Frans
10		Frans
11		Rizki
12		Rizki

No	Gambar	Label
13		Rizki
14		Rizki
15		Rizki

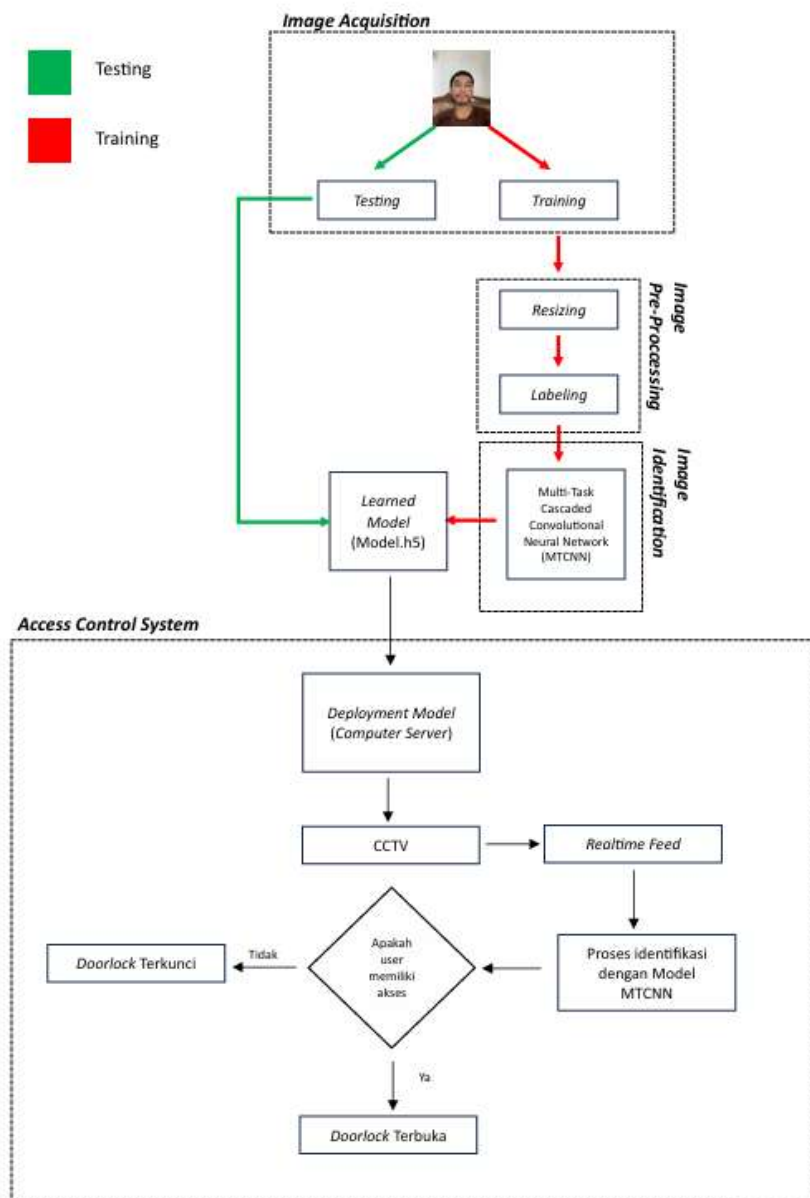
3.2. Arsitektur Umum

Sistem yang sedang dikembangkan dimulai dengan tahap akuisisi gambar, di mana citra dari lingkungan sekitar dibagi menjadi dua bagian utama: data *training* dan data *testing*. Data *training* mengalami proses *preprocessing* untuk mengubah ukuran dan memberi label sebelum diarahkan ke tahap identifikasi gambar menggunakan model *Multi-Task Cascade Convolutional Neural Network* (MTCNN). Proses *training* dilakukan untuk membangun model yang mampu mengenali pola dan fitur dari wajah dalam gambar.

Setelah proses *training* selesai, model yang terlatih disimpan dalam format .h5 dan *deploy* ke server komputer yang terkoneksi dengan CCTV melalui protokol RTSP. CCTV, sebagai sensor visual utama, mengambil gambar secara *real-time* dan melakukan pemrosesan dengan model MTCNN untuk mengidentifikasi keberadaan dan identitas pemilik rumah berdasarkan wajah yang terdeteksi.

Ketika wajah yang terdeteksi cocok dengan data pemilik rumah, CCTV mengirimkan sinyal ke ESP atau perangkat terkait untuk membuka kunci pintu. Sebaliknya, jika wajah tidak dikenali sebagai pemilik rumah, CCTV mengirimkan instruksi untuk mengunci pintu. Proses ini mengintegrasikan teknologi *machine learning* dalam pengambilan keputusan berbasis visual untuk otomatisasi kontrol akses pintu dengan akurasi dan keandalan yang tinggi.

Dengan menggabungkan deteksi visual *real-time* dan pengolahan data menggunakan model *machine learning* yang canggih, sistem ini tidak hanya meningkatkan keamanan rumah tetapi juga menawarkan solusi otomatis yang efisien dan responsif terhadap identitas pengguna yang berada di depan pintu. Serangkaian proses ini dapat dilihat pada Gambar 3.1.



Gambar 3.1 Arsitektur umum

3.2.1. Akusisi gambar dan pembagian data

Pertama-tama, sistem mengambil gambar dari lingkungan sekitar menggunakan perangkat CCTV atau webcam. Gambar yang diambil dibagi menjadi dua kelompok utama: data *training* dan data *testing*. Data *training* digunakan untuk melatih model *machine learning*, sedangkan data *testing* digunakan untuk menguji kinerja model setelah dilatih. Berikut merupakan *pseudocode* untuk mendeteksi wajah dan mengambilnya menjadi *dataset*.

Tabel 3.2 *Pseudocode* akuisi gambar dan pembagian *dataset*

```

Mulai
  Definisikan path folder untuk citra asli
  Definisikan path folder untuk data training
  Definisikan path folder untuk data testing
  Definisikan proporsi pembagian data (misalnya 0.8 untuk
training, 0.2 untuk testing)

  Fungsi      split_data(input_folder,      training_folder,
testing_folder, split_ratio)
  Mulai
    Dapatkan daftar file citra dalam input_folder
    Acak urutan file citra
    Tentukan batas pembagian berdasarkan split_ratio
    Pisahkan file citra menjadi data_training dan data_testing
berdasarkan batas
    Pindahkan data_training ke training_folder
    Pindahkan data_testing ke testing_folder
  Selesai
  Akhir Fungsi

  Panggil fungsi split_data dengan parameter (path folder citra
asli, path folder training, path folder testing, proporsi
pembagian data)
Selesai

```

Pseudocode di atas menjelaskan proses pembagian data citra menjadi dua kelompok: data training dan data testing. Pertama, path folder untuk citra asli, data *training*, dan data *testing* didefinisikan, serta proporsi pembagian data ditentukan, misalnya 0.8 untuk training dan 0.2 untuk testing. Fungsi `'split_data'` kemudian digunakan untuk melakukan pembagian ini. Fungsi tersebut mendapatkan daftar *file* citra dalam folder *input*, mengacak urutan *file* untuk memastikan distribusi acak, dan menentukan batas pembagian berdasarkan proporsi yang ditentukan. Setelah itu, *file* citra dipisahkan menjadi dua kelompok: data *training* dan data *testing*. *File* dalam kelompok data training dipindahkan ke folder *training*, sedangkan *file* dalam kelompok

data *testing* dipindahkan ke folder *testing*. Fungsi ini kemudian dipanggil dengan parameter yang sesuai untuk memulai proses pembagian data.

3.2.2. *Preprocessing Data*

Data *training* mengalami tahap *preprocessing* untuk mempersiapkannya sebelum dilatih dengan model. Proses *preprocessing* mencakup dua tahap utama: *resizing* dan *labeling*. *Resize* dilakukan untuk memastikan bahwa semua gambar memiliki ukuran yang konsisten, yang penting untuk konsistensi dalam proses *training*. *Labeling* mengidentifikasi kategori atau identitas yang sesuai dengan setiap gambar, yang penting untuk pelatihan model terawasi.

Tabel 3.3 *Pseudocode preprocessing data*

```

Mulai
  Definisikan path folder untuk data training
  Definisikan path folder untuk menyimpan data yang telah di-
  preprocess
  Definisikan ukuran pixel baru (target_size)

  Fungsi preprocess_image(image_path, output_path, target_size)
  Mulai
    Baca citra dari image_path
    Ubah ukuran citra menjadi target_size
    Simpan citra yang telah di-resize ke output_path
  Selesai
  Akhir Fungsi

  Untuk setiap file citra dalam folder data training
  Mulai
    Panggil fungsi preprocess_image dengan parameter (path file
    citra asli, path folder output, ukuran pixel baru)
  Selesai
Selesai

```

Pseudocode di atas menggambarkan proses *preprocessing* data citra untuk persiapan *training* model. Pertama, *path* folder untuk data *training* dan *folder* untuk menyimpan data yang telah di-*preprocess* didefinisikan, serta ukuran *pixel* baru (*target_size*) ditentukan. Fungsi `preprocess_image` digunakan untuk membaca setiap citra dari folder data *training*, mengubah ukurannya sesuai dengan *target_size*, dan menyimpan citra yang telah di-*resize* ke folder *output*. Fungsi ini kemudian dipanggil untuk setiap *file* citra dalam folder data *training*. Proses ini memastikan bahwa semua citra dalam data *training* memiliki ukuran yang konsisten, yang penting untuk konsistensi dalam proses *training* model *machine learning*.

3.2.3. Identifikasi gambar dengan model MTCNN

Setelah *preprocessing* selesai, data *training* diteruskan ke tahap identifikasi gambar menggunakan model *Multi-Task Cascade Convolutional Neural Network* (MTCNN), khususnya MTCNN (*Multi-Task Cascaded Convolutional Neural Network*). MTCNN digunakan untuk deteksi wajah dalam gambar dan ekstraksi fitur-fitur penting dari wajah tersebut, seperti titik *landmark*. Proses *training* ini bertujuan untuk mengajarkan model mengenali pola dan fitur yang spesifik dari wajah.

Tabel 3.4 *Pseudocode* identifikasi gambar dengan model MTCNN

```

Mulai
  Definisikan path folder untuk data training yang telah di-
  preprocess
  Definisikan path folder untuk menyimpan model terlatih
  Definisikan parameter model MTCNN

  Fungsi          train_mtcnn_model(training_data_folder,
  model_output_path, parameters)
  Mulai
    Muat dan persiapkan data dari training_data_folder
    Inisialisasi model MTCNN dengan parameters
    Latih model MTCNN menggunakan data training
    Simpan model terlatih ke model_output_path
  Selesai

```

Akhir Fungsi

```
Panggil fungsi train_mtcnn_model dengan parameter (path
folder data training, path folder output model, parameter model
MTCNN)
Selesai
```

Pseudocode di atas menjelaskan proses identifikasi gambar menggunakan model MTCNN. Pertama, *path* folder untuk data *training* yang telah di-*preprocess* dan folder untuk menyimpan model terlatih didefinisikan, serta parameter untuk model MTCNN ditentukan. Fungsi `train_mtcnn_model` digunakan untuk memuat dan mempersiapkan data dari folder data *training*, menginisialisasi model MTCNN dengan parameter yang ditentukan, dan melatih model tersebut menggunakan data training. Setelah model dilatih, model yang terlatih disimpan ke *path* folder *output* yang telah ditentukan. Fungsi ini kemudian dipanggil dengan parameter yang sesuai untuk memulai proses pelatihan model MTCNN. Proses ini memungkinkan model MTCNN untuk mengenali pola dan fitur dari wajah dalam data *training*, sehingga dapat digunakan untuk identifikasi gambar dalam aplikasi real-time.

3.2.4. Pembuatan dan deployment model

Setelah proses *training* selesai, model yang terlatih disimpan dalam format `.h5`, yang merupakan format standar untuk menyimpan model *machine learning* dalam TensorFlow atau Keras. Model ini kemudian *deploy* ke *server* komputer yang terhubung dengan CCTV melalui protokol RTSP. Pada tahap ini, model siap digunakan untuk menganalisis gambar yang diambil secara real-time oleh CCTV.

Tabel 3.5 *Pseudocode* pembuatan dan *deployment* model

Mulai

```
Definisikan path folder untuk data training yang telah di-
preprocess
```

```
Definisikan path folder untuk menyimpan model terlatih
```

```
Definisikan path untuk menyimpan model dalam format .h5
```

```
Definisikan parameter model MTCNN
```

```

Fungsi          train_and_save_model(training_data_folder,
model_output_path, h5_output_path, parameters)
Mulai
    Muat dan persiapkan data dari training_data_folder
    Inisialisasi model MTCNN dengan parameters
    Latih model MTCNN menggunakan data training
    Simpan model terlatih ke model_output_path
    Simpan model terlatih dalam format .h5 ke h5_output_path
Selesai
Akhir Fungsi

Panggil fungsi train_and_save_model dengan parameter (path
folder data training, path folder output model, path file .h5,
parameter model MTCNN)
Selesai

```

Pseudocode di atas menjelaskan proses pembuatan dan *deployment* model untuk identifikasi gambar menggunakan MTCNN. Langkah pertama adalah mendefinisikan *path* folder untuk data *training* yang telah di-*preprocess*, *path folder* untuk menyimpan model terlatih, *path* untuk menyimpan model dalam format .h5, serta parameter untuk model MTCNN. Fungsi `train_and_save_model` digunakan untuk memuat dan mempersiapkan data dari folder data training, menginisialisasi model MTCNN dengan parameter yang telah ditentukan, dan melatih model tersebut menggunakan data training. Setelah pelatihan selesai, model yang terlatih disimpan ke path folder output yang telah ditentukan, dan juga disimpan dalam format .h5 ke path file yang telah ditentukan. Fungsi ini kemudian dipanggil dengan parameter yang sesuai untuk memulai proses pelatihan dan penyimpanan model MTCNN. Proses ini memastikan bahwa model MTCNN siap digunakan untuk identifikasi gambar dalam aplikasi real-time dan dapat dideploy ke *server* untuk pemrosesan lebih lanjut.

3.2.5. Pemrosesan real-time oleh CCTV

CCTV berfungsi sebagai sensor visual utama dalam sistem. Mengambil gambar secara real-time, CCTV memproses gambar menggunakan model MTCNN yang sudah

terlatih. Tujuan utama pemrosesan ini adalah untuk mengidentifikasi keberadaan dan identitas pengguna berdasarkan wajah yang terdeteksi dalam gambar.

Tabel 3.6 Pemrosesan *real-time* oleh CCTV

```

Mulai
  Definisikan alamat RTSP untuk CCTV
  Definisikan path untuk model yang telah di-deploy dalam format
  .h5
  Inisialisasi koneksi ke CCTV menggunakan RTSP
  Muat model MTCNN dari path file .h5

  Fungsi process_frame(frame, model)
  Mulai
    Gunakan model untuk mendeteksi wajah pada frame
    Jika wajah terdeteksi
      Identifikasi apakah wajah adalah pemilik rumah
      Jika wajah adalah pemilik rumah
        Kirim sinyal ke ESP untuk membuka kunci pintu
      Jika tidak
        Kirim sinyal ke ESP untuk mengunci pintu
    Selesai
  Akhir Fungsi

  Loop untuk setiap frame yang diterima dari CCTV
  Mulai
    Baca frame dari CCTV
    Panggil fungsi process_frame dengan parameter (frame, model)
  Selesai
Selesai

```

Pseudocode di atas menjelaskan proses pemrosesan *real-time* oleh CCTV untuk identifikasi pemilik rumah. Pertama, alamat RTSP untuk CCTV dan path untuk model MTCNN yang telah di-*deploy* dalam format .h5 didefinisikan. Koneksi ke CCTV diinisialisasi menggunakan RTSP, dan model MTCNN dimuat dari *path file* .h5. Fungsi `process_frame` digunakan untuk memproses setiap *frame* yang diterima dari CCTV.

Fungsi ini mendeteksi wajah pada *frame* menggunakan model MTCNN. Jika wajah terdeteksi, sistem mengidentifikasi apakah wajah tersebut adalah pemilik rumah. Jika wajah adalah pemilik rumah, sinyal dikirim ke ESP untuk membuka kunci pintu. Jika wajah tidak dikenali sebagai pemilik rumah, sinyal dikirim ke ESP untuk mengunci pintu. *Loop* terus berjalan untuk setiap *frame* yang diterima dari CCTV, memanggil fungsi `process_frame` untuk memproses *frame* tersebut. Proses ini memungkinkan sistem untuk melakukan identifikasi wajah secara *real-time* dan mengambil tindakan yang sesuai berdasarkan hasil identifikasi.

3.2.6. Keputusan dan kontrol akses

Berdasarkan hasil identifikasi, sistem membuat keputusan apakah wajah yang terdeteksi merupakan pemilik rumah atau bukan. Jika pemilik rumah teridentifikasi, CCTV mengirimkan sinyal ke perangkat ESP atau alat terkait untuk membuka kunci pintu. Sebaliknya, jika wajah tidak dikenali sebagai pemilik rumah, CCTV mengirimkan instruksi untuk mengunci pintu. Proses ini memberikan keamanan tambahan dengan memanfaatkan teknologi deteksi visual dan pengambilan keputusan berbasis *machine learning*.

Tabel 3.7 Pseudocode keputusan dan kontrol akses

```

Mulai
  Definisikan alamat RTSP untuk CCTV
  Definisikan path untuk model yang telah di-deploy dalam format
  .h5
  Inisialisasi koneksi ke CCTV menggunakan RTSP
  Muat model MTCNN dari path file .h5
  Inisialisasi koneksi ke ESP untuk kontrol akses

  Fungsi process_frame(frame, model, esp_connection)
  Mulai
    Gunakan model untuk mendeteksi wajah pada frame
    Jika wajah terdeteksi
      Identifikasi apakah wajah adalah pemilik rumah
      Jika wajah adalah pemilik rumah

```

```

    Kirim sinyal ke ESP untuk membuka kunci pintu
    Jika tidak
        Kirim sinyal ke ESP untuk mengunci pintu
    Selesai
    Akhir Fungsi

Loop untuk setiap frame yang diterima dari CCTV
Mulai
    Baca frame dari CCTV
    Panggil fungsi process_frame dengan parameter (frame, model,
esp_connection)
    Selesai
Selesai

```

Pseudocode di atas menggambarkan proses pengambilan keputusan dan kontrol akses dalam sistem keamanan pintu otomatis berbasis deteksi wajah. Langkah pertama adalah mendefinisikan alamat RTSP untuk CCTV dan path untuk model MTCNN yang telah di-*deploy* dalam format .h5. Koneksi ke CCTV diinisialisasi menggunakan RTSP, dan model MTCNN dimuat dari path file .h5. Selain itu, koneksi ke perangkat ESP untuk kontrol akses juga diinisialisasi.

Fungsi `process_frame` digunakan untuk memproses setiap *frame* yang diterima dari CCTV. Fungsi ini mendeteksi wajah pada *frame* menggunakan model MTCNN. Jika wajah terdeteksi, sistem mengidentifikasi apakah wajah tersebut adalah pemilik rumah. Jika wajah dikenali sebagai pemilik rumah, sinyal dikirim ke ESP untuk membuka kunci pintu. Jika wajah tidak dikenali sebagai pemilik rumah, sinyal dikirim ke ESP untuk mengunci pintu.

Loop utama berjalan terus menerus untuk setiap *frame* yang diterima dari CCTV, memanggil fungsi `process_frame` dengan parameter *frame*, model, dan koneksi ESP. Proses ini memastikan bahwa sistem secara *real-time* dapat mengidentifikasi wajah dan mengambil tindakan yang tepat untuk mengontrol akses pintu, meningkatkan keamanan dan efisiensi sistem.

BAB 4

IMPLEMENTASI DAN PENGUJIAN

4.1. Implementasi Sistem

Pada tahap ini, implementasi *Multi-Task Cascade Convolutional Neural Network* (MTCNN) untuk mengidentifikasi pemilik rumah menggunakan ESP8266 serta CCTV. Oleh karena itu sistem ini memerlukan perangkat keras dan perangkat lunak pendukung, yaitu:

4.1.1. Spesifikasi perangkat keras

Spesifikasi perangkat keras yang digunakan untuk membangun sistem ini adalah:

1. Perangkat laptop yang digunakan adalah: Asus ROG Strix G531GD
2. *Processor* 2,4 GHz Octa-Core Intel Core i5-933
3. GPU Nvidia GeForce GTX 1050
4. Kapasitas RAM sebesar 8 GB
5. Kapasitas SSD sebesar 512 GB

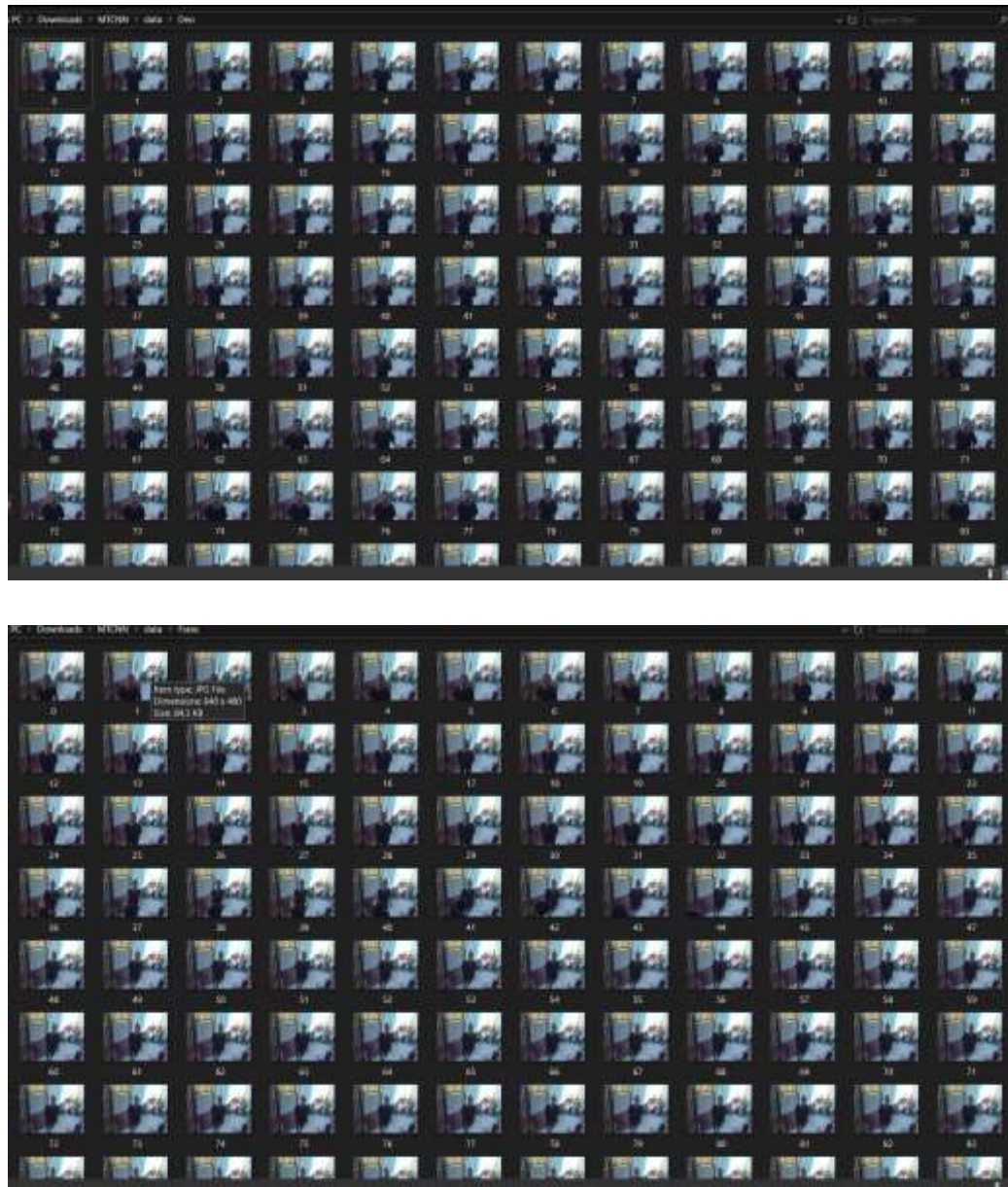
4.1.2. Spesifikasi perangkat lunak

Spesifikasi perangkat lunak yang digunakan untuk membangun sistem ini adalah:

1. Sistem Operasi Windows 10 Home
2. Text Editor Visual Studio

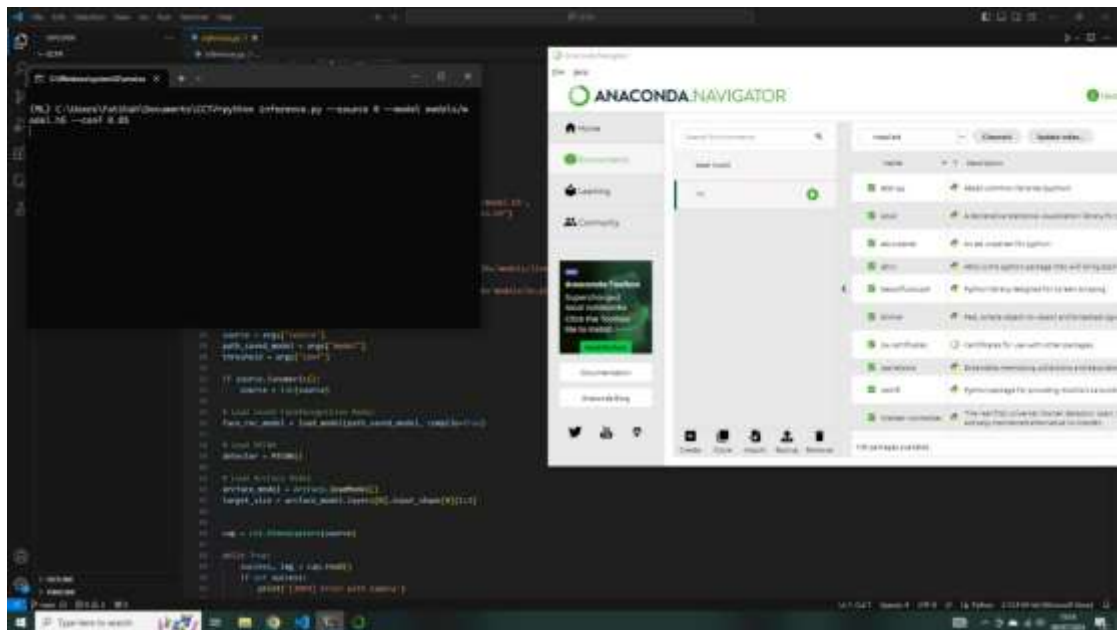
4.2. Penggunaan Data

Data yang diambil untuk digunakan dalam penelitian ini yaitu data yang berasal dari wajah pemilik. Pengambilan data citra wajah diambil langsung menggunakan kamera CCTV. Jumlah data yang dikumpulkan yaitu sebanyak 100 citra. Adapun beberapa contoh data dapat dilihat pada Gambar 4.1.



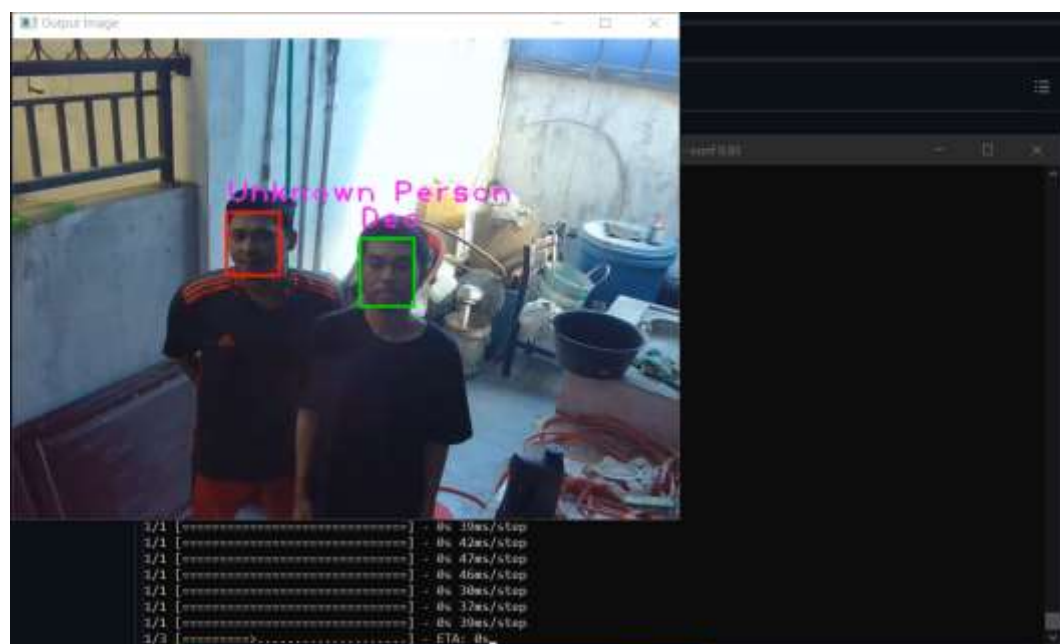
Gambar 4.1 Data citra pemilik rumah

Pada pengujian ini peneliti menggunakan ESP8266 yang telah dirakit serta telah terhubung dengan CCTV rumah yang digunakan untuk mengidentifikasi pemilik rumah agar dapat membuka dan mengunci pintu. Adapun program yang terdapat MTCNN tersebut harus dijalankan terlebih dahulu. Proses menjalankannya dapat dilihat pada Gambar 4.2.



Gambar 4.2 Proses menjalankan program

Setelah program dijalankan, kita dapat melihat tampilan CCTV yang sudah terkoneksi dan jika CCTV mendeteksi pemilik rumah maka akan muncul *border* sebagai tanda bahwasannya CCTV mendeteksi pemilik rumah tersebut. Untuk tampilannya dapat dilihat pada Gambar 4.3.



Gambar 4.3 Tampilan video *live streaming* CCTV

Peralatan untuk pengujian dapat dilihat pada Gambar 4.4 dan lokasi serta posisi CCTV dapat dilihat pada Gambar 4.5.

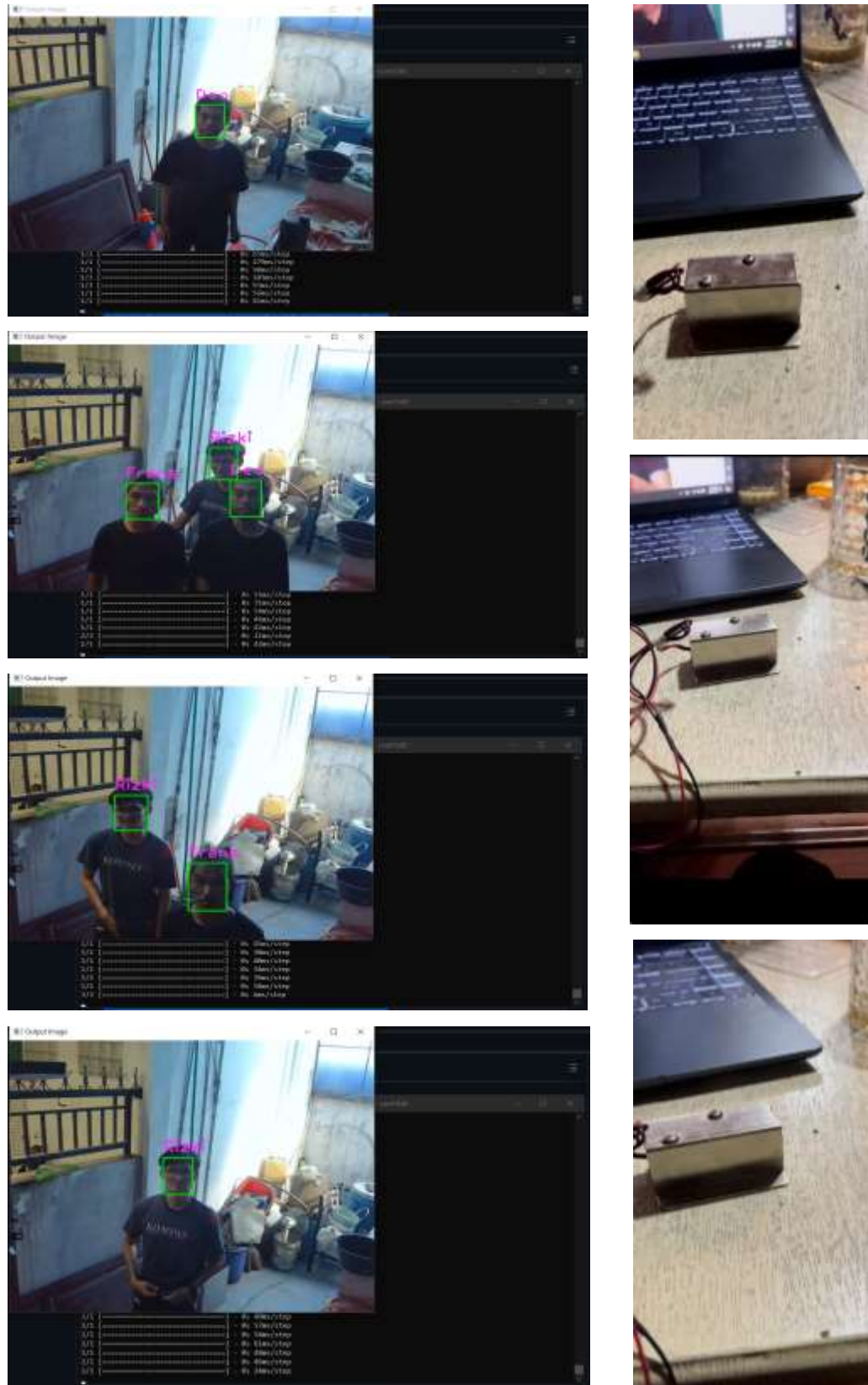


Gambar 4.4 Peralatan pengujian

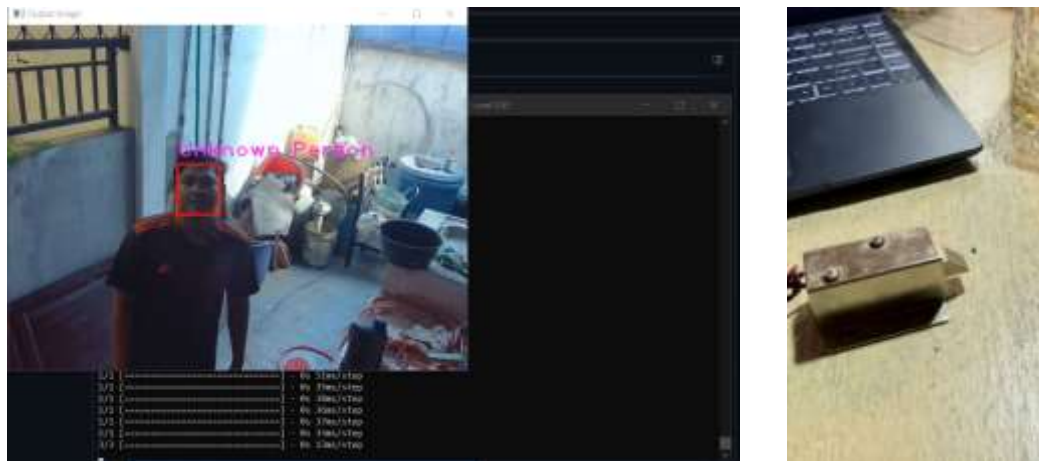


Gambar 4.5 Lokasi serta posisi CCTV

Proses pengujian dalam mengidentifikasi pemilik rumah dapat dilihat pada Gambar 4.6 dan Gambar 4.7.



Gambar 4.6 Pengujian ketika mengidentifikasi pemilik rumah



Gambar 4.7 Pengujian ketika tidak mengidentifikasi pemilik rumah

4.3. Hasil Pengujian

Beberapa sampel dari pengujian dapat dilihat pada Tabel 4.1.

Tabel 4.1 Sampel pengujian

No	Actual	Predicted	TP	TN	FP	FN
1	User	User	1	0	0	0
2	User	User	1	0	0	0
3	Unknow Person	User	0	1	0	0
4	User	User	1	0	0	0
5	User	User	1	0	0	0
6	User	User	1	0	0	0
7	Unknow Person	Unknow Person	0	1	0	0
8	Unknow Person	Unknow Person	0	1	0	0
9	User	User	1	0	0	0
10	Unknow Person	Unknow Person	0	1	0	0
11	User	User	1	0	0	0
12	User	User	1	0	0	0
13	User	User	1	0	0	0
14	Unknow Person	Unknow Person	0	1	0	0
15	Unknow Person	Unknow Person	0	1	0	0
16	User	User	1	0	0	0
17	User	User	1	0	0	0
18	User	User	1	0	0	0
19	Unknow Person	Unknow Person	0	1	0	0
20	Unknow Person	User	0	0	1	0

- Accuracy

$$(TP + TN) / (TP + TN + FP + FN)$$

$$= (12 + 7) / (12 + 7 + 1 + 0) = 19 / 20$$

$$= 0.95 \text{ atau } 95\%$$

- Precision

$$TP / (TP + FP)$$

$$= 12 / (12 + 1)$$

$$= 12 / 13 \approx 0.92 \text{ atau } 92.2\%$$

- Recall

$$TP / (TP + FN)$$

$$= 12 / (12 + 0)$$

$$= 12 / 12 \approx 1 \text{ atau } 100\%$$

- F1 Score

$$2 \times (\text{Presisi} \times \text{Recall}) / (\text{Presisi} + \text{Recall})$$

$$= 2 \times (0.92 \times 1) / (0.9 + 1) \approx 0.96$$

Tabel yang disajikan memperlihatkan hasil dari pengujian model deteksi wajah terhadap 20 gambar. Setiap gambar memiliki status "*Actual*" yang menandakan apakah gambar tersebut merupakan data pengguna yang sah ("*User*") atau data yang bukan merupakan pengguna yang sah ("*Unknow Person*"). Selain itu, tabel juga menunjukkan hasil prediksi model deteksi wajah dengan status "*Predicted*", yang menunjukkan apakah model berhasil mengidentifikasi gambar sebagai pengguna yang sah atau bukan.

Dari tabel, terdapat hasil prediksi yang dapat dikelompokkan sebagai *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). *True Positive* (TP) menunjukkan jumlah gambar yang berhasil diprediksi sebagai pengguna yang sah dengan benar. *True Negative* (TN) menunjukkan jumlah gambar yang berhasil diprediksi sebagai bukan pengguna yang sah dengan benar. *False Positive* (FP) menunjukkan jumlah gambar yang diprediksi sebagai pengguna yang sah padahal sebenarnya bukan. Sementara *False Negative* (FN) menunjukkan jumlah gambar yang diprediksi sebagai bukan pengguna yang sah padahal sebenarnya adalah pengguna yang sah.

Dari hasil pengujian tersebut, dapat dihitung nilai *Accuracy*, *Precision*, *Recall*, dan *F1 Score*. Akurasi (*Accuracy*) mengukur seberapa akurat model dalam memprediksi keseluruhan data. Presisi (*Precision*) mengukur seberapa banyak dari prediksi positif yang benar-benar benar. *Recall* mengukur seberapa banyak dari data yang sebenarnya

positif yang diprediksi dengan benar oleh model. *F1 Score* adalah metrik yang mengkombinasikan presisi dan recall untuk memberikan gambaran keseluruhan tentang kinerja model. Dari tabel sampel pengujian menghasilkan *Accuracy* mencapai 95%, *Precision* mencapai 92,2%, *Recall* mencapai 100% serta *F1 Score* mencapai 96%.

BAB 5

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Didapatkan kesimpulan dari keseluruhan penelitian menggunakan *Multi-Task Cascade Convolutional Neural Network* (MTCNN) untuk mengidentifikasi pemilik rumah menggunakan ESP8266 serta CCTV hingga pengujian adalah sebagai berikut:

1. Implementasi *Multi-Task Cascade Convolutional Neural Network* (MTCNN) untuk mengidentifikasi pemilik rumah menggunakan ESP8266 serta CCTV menghasilkan *Accuracy* mencapai 95%, *Precision* mencapai 92,2%, *Recall* mencapai 100% serta *F1 Score* mencapai 96%.
2. Kesalahan sistem dalam mengidentifikasi pemilik rumah dapat dipengaruhi oleh intensitas cahaya dan jarak CCTV.
3. Penerapan sistem ini memiliki potensi besar dalam meningkatkan efisiensi dalam pengamanan rumah dengan mengidentifikasi pemilik rumah dengan CCTV yang terhubung dengan penguncian pintu rumah.

5.2 Saran

Adapun saran untuk pengembangan penelitian berikutnya adalah sebagai berikut:

1. Membuat agar CCTV dapat mendeksi gerakan mencurigakan yang terhubung dengan penguncian pintu sehingga dapat memberikan notifikasi kepada pemilik rumah.
2. Membuat aplikasi berbasis *mobile* yang terhubung dengan CCTV serta Pengamanan rumah agar lebih efisien sehingga dapat dikontrol darimana saja.
3. Menerapkan metode identifikasi yang lain untuk dibandingkan dengan hasil yang telah diperoleh dari metode *Multi-Task Cascade Convolutional Neural Network* (MTCNN).

4. Menambahkan sebuah sensor pada CCTV untuk memindai kedalaman kontur wajah sehingga mampu membedakan antara wajah dalam foto (kertas) dan wajah sungguhan.

DAFTAR PUSTAKA

- A.Saran Kumar, & R. Rekha. (2023). Improving Smart Home Safety with Face Recognition using Machine Learning. *2023 International Conference on Intelligent Systems for Communication, IoT and Security (ICISCOIS)*. Coimbatore, India : IEEE.
- Anggara, K. D., Kartikasari, D. P., & Bakhtiar, F. A. (2023). Implementasi Algoritma MTCNN dalam Mekanisme Autentikasi berbasis Pengenalan Wajah. *Jurnal Pengembangan Teknologi Informasi Dan Ilmu Komputer*, 7(8), 3613–3621.
- Aydin, I., & Othman, N. A. (2017). A new IoT combined face detection of people by using computer vision for security application. *2017 International Artificial Intelligence and Data Processing Symposium (IDAP)*. Malatya, Turkey: IEEE.
- Basuki, A., Palandi, J. F., & Fatchurrochman. (2005). *Pengolahan citra digital menggunakan Visual Basic* (Vol. 1). Yogyakarta, Indonesia: Graha Ilmu.
- Ben Wycliff Mugalu, Rodrick Calvin Wamala, Jonathan Serugunda, & Andrew Katumba. (2021). *Face Recognition as a Method of Authentication in a Web-Based System*. Diambil kembali dari <https://arxiv.org/abs/2103.15144>
- Chatisa, I., Syahbana, Y. A., & Wibowo, A. U. (2023). A building security monitoring system based on the internet of things (IoT) with illumination-invariant face recognition for object detection. *KINETIK*, 8(1).
- Daniel Widjaja, Derrick, Muhammad Fathariq Dimas Octaviandra, Said Achmad, & Rhio Sutoyo. (2022). Important Security Factors for Implementing Internet of Things in Smart Home Systems. *2022 International Conference on Informatics Electrical and Electronics (ICIEE)*. Yogyakarta, Indonesia: IEEE.
- Daniel Widjaja, Derrick, Muhammad Fathariq Dimas Octaviandra, Said Achmad, & Rhio Sutoyo. (2022). Important Security Factors for Implementing Internet of Things in Smart Home Systems. *2022 International Conference on Informatics Electrical and Electronics (ICIEE)*. Yogyakarta, Indonesia: IEEE.
- Faisal, F., & Hossain, S. A. (2019). Smart Security System Using Face Recognition on Raspberry Pi. *2019 13th International Conference on Software, Knowledge,*

- Information Management and Applications (SKIMA)*. Island of Ulkulhas, Maldives: IEEE.
- Jose, E., Greeshma M., Haridas, M. T., & M.H. Supriya. (2019). Face Recognition based Surveillance System Using FaceNet and MTCNN on Jetson TX2. *2019 5th International Conference on Advanced Computing & Communication Systems (ICACCS)*. Coimbatore, India : IEEE.
- Khattar, S., Sachdeva, A., Kumar, R., & Gupta, R. (2019). Smart Home With Virtual Assistant Using Raspberry Pi. *2019 9th International Conference on Cloud Computing, Data Science & Engineering (Confluence)*. Noida, India : IEEE..
- Kulkarni, A. (2001). Computer Vision and Fuzzy Neural System. New Jersey: Prentice Hall PTR.
- Mantoro, T., Ayu, M. A., & Suhendi. (2018). Multi-Faces Recognition Process Using Haar Cascades and Eigenface Methods. *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*. Rabat, Morocco: IEEE.
- Munir, A., Ehsan, S., Raza, S., & Mudassir, M. (2019). Face and Speech Recognition Based Smart Home. *2019 International Conference on Engineering and Emerging Technologies (ICEET)*. Lahore, Pakistan: IEEE.
- Nuryaman, R. (2019). Rancang Bangun Aplikasi Video Streaming Menggunakan Real Time Streaming Protocol (RTSP) Berbasis Android.
- Prayogi, A., Dian , R., Siregar , R. M., & Sugianto , R. A. (2024). Low-Cost CCTV for Home Security With Face Detection Base on IoT. *Journal of Information System and Technology Research*, 3(1).
- Rafael, G., Kusuma, H., & Tasripan. (2020). The Utilization of Cloud Computing for Facial Expression Recognition using Amazon Web Services. *2020 International Conference on Computer Engineering, Network, and Intelligent Multimedia (CENIM)*. Surabaya, Indonesia: IEEE.
- Riskiono, S. D., Pamungkas, R. H., & Arya, Y. (2020). RANCANG BANGUN SISTEM PENYIRAMAN TANAMAN SAYUR BERBASIS ARDUINO DENGAN SENSOR KELEMBABAN TANAH. *Jurnal Ilmiah Mahasiswa Kendali dan Listrik*.
- Rosa Andrie Asmara , Muhammad Ridwan , Gunawan Budi P , & Anik Nur Handayani . (2022). Pengembangan Sistem Face Recognition menggunakan Cloud Service,

- Raspberry Pi dan Convolutional Neural Network (CNN). *Jurnal Informatika Polinema*, 9(1), 95-102.
- Saghoa, Y. C., Sompie, S. R., & Tulung, N. M. (2018). Kotak Penyimpanan Uang Berbasis Mikrokontroler Arduino Uno. *Jurnal Teknik Elektro dan Komputer*.
- Turang, D. A. (2015). PENGEMBANGAN SISTEM RELAYPENGENDALIAN DAN PENGHEMATAN PEMAKAIAAN LAMPU BERBASIS MOBILE. *Seminar Nasional Informatika 2015*.
- Wang, J., & Ma, M. (2018). Multi-view face detection and landmark localization based on MTCNN. *In 2018 Chinese Automation Congress (CAC)* (hal. 4200-4205). IEEE.
- Zhang, K., Zhang, Z., Li, Z., & Qiao, Y. (2016). Joint Face Detection and Alignment using Multi-task Cascaded Convolutional Networks . *IEEE Signal Processing Letters*.



KEMENTERIAN PENDIDIKAN, KEBUDAYAAN,
RISET, DAN TEKNOLOGI
UNIVERSITAS SUMATERA UTARA
FAKULTAS ILMU KOMPUTER DAN TEKNOLOGI INFORMASI

Jalan Universitas No. 9A Gedung A, Kampus USU Medan 20155, Telepon: (061) 821007
Laman: <http://Fasilkomti.usu.ac.id>

KEPUTUSAN
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI
NOMOR : 2715/UN5.2.14.D/SK/SPB/2024
DEKAN FAKULTAS ILMU KOMPUTER
DAN TEKNOLOGI INFORMASI UNIVERSITAS SUMATERA UTARA

- Membaca : Surat Permohonan Mahasiswa Fasilkom-TI USU tanggal 10 Juli 2024 perihal permohonan ujian skripsi:
Nama : DEO PRANATA SILITONGA
NIM : 171402065
Program Studi : Sarjana (S-1) Teknologi Informasi
Judul Skripsi : Pengembangan Sistem Kunci Pintu Otomatis Berbasis Esp8266 Menggunakan CCTV Rumah dan Algoritma Convolutional Neural Network Untuk Identifikasi Pemilik Rumah
- Memperhatikan : Bahwa Mahasiswa tersebut telah memenuhi kewajiban untuk ikut dalam pelaksanaan Meja Hijau Skripsi Mahasiswa pada Program Studi Sarjana (S-1) Teknologi Informasi Fakultas Ilmu Komputer dan Teknologi Informasi Universitas Sumatera Utara TA 2023/2024.
- Menimbang : Bahwa permohonan tersebut diatas dapat disetujui dan perlu ditetapkan dengan surat keputusan
- Mengingat : 1. Undang-undang Nomor 20 Tahun 2003 tentang Sistem Pendidikan Nasional.
2. Peraturan Pemerintah Nomor 17 tahun 2010 tentang pengelolaan dan penyelenggara pendidikan.
3. Keputusan Rektor USU Nomor 03/UN5.1.R/SK/SPB/2021 tentang Peraturan Akademik Program Sarjana Universitas Sumatera Utara.
4. Surat Keputusan Rektor USU Nomor 1876/UN5.1.R/SK/SDM/2021 tentang pengangkatan Dekan Fasilkom-TI USU Periode 2021-2026
- MEMUTUSKAN
- Menetapkan :
Pertama : Membentuk dan mengangkat Tim Penguji Skripsi mahasiswa sebagai berikut:
Ketua : Ainul Hizriadi S.Kom, M.Sc
NIP: 198510272017061001
Sekretaris : Fahrurrozi Lubis B.IT., M.Sc.IT
NIP: 198610122018052001
Anggota Penguji : Seniman S.Kom., M.Kom.
NIP: 198705252014041001
Anggota Penguji : Dr. Romi Fadillah Rahmat, B.Comp.Sc., M.Sc.
NIP: 198603032010121004
Moderator : -
Panitera : -
- Kedua : Segala biaya yang diperlukan untuk pelaksanaan kegiatan ini dibebankan pada Dana Penerimaan Bukan Pajak (PNPB) Fasilkom-TI USU Tahun 2024.
- Ketiga : Keputusan ini berlaku sejak tanggal ditetapkan dengan ketentuan bahwa segala sesuatunya akan diperbaiki sebagaimana mestinya apabila dikemudian hari terdapat kekeliruan dalam surat keputusan ini.

- Tembusan :
- 1. Ketua Program Studi Sarjana (S-1) Teknologi Informasi
 - 2. Yang bersangkutan
 - 3. Arsip

Medan
Ditandatangani secara elektronik oleh:
Dekan



Maya Silvi Lydia
NIP 197401272002122001