

LEMBAR PENGESAHAN

**PENGONTROLAN *AIR CONDITIONER* BERDASARKAN
JUMLAH ORANG DALAM RUANGAN BERBASIS
PENGOLAHAN CITRA**

Oleh:

Eri Nur Mushthafa/1052140

Herda Fitri Permatasari Riva'i/1052143

Laporan akhir ini telah disetujui dan disahkan sebagai salah satu syarat kelulusan
Program Sarjana Terapan Politeknik Manufaktur Negeri Bangka Belitung

Menyetujui,

Pembimbing 1

Pembimbing 2



Aan Febriansyah, M.T



Irwan, MSc., Ph.D

Penguji 1

Penguji 2



Eko Sulistyono, M. T



Dr. Parulian Silalahi, M.Pd

PERNYATAAN BUKAN PLAGIAT

Yang bertanda tangan di bawah ini :

Nama Mahasiswa 1 : Eri Nur Mushthafa NIM : 1052140

Nama Mahasiswa 2 : Herda Fitri Permatasari Riva'i NIM : 1052143

Dengan Judul : Pengontrolan *Air Conditioner* Berdasarkan Jumlah
Orang Dalam Ruangan Berbasis Pengolahan Citra

Menyatakan bahwa laporan akhir ini adalah hasil kerja kami sendiri dan bukan merupakan plagiat. Pernyataan ini kami buat dengan sebenarnya dan bila ternyata dikemudian hari ternyata melanggar pernyataan ini, kami bersedia menerima sanksi yang berlaku.

Sungailiat, 24 Juli 2024

Nama Mahasiswa

Tanda Tangan

1. Eri Nur Muhthafa

2. Herda Fitri Permatasari Riva'i

ii

ABSTRAK

Pengontrolan fan AC dan perhitungan jumlah orang dalam ruangan merupakan salah satu aspek yang sangat berpengaruh untuk menciptakan lingkungan kerja yang efisien dan nyaman. Dengan perkembangan teknologi yang berkembang semakin pesat, image processing hadir dengan daya tarik tersendiri sebagai metode yang dapat digunakan untuk memproses gambar dengan inputan berupa gambar 2 dimensi. Tujuan dari proyek akhir ini yaitu menciptakan sebuah sistem yang menggunakan image processing agar dapat mengontrol fan AC dan menghitung jumlah orang yang masuk ke dalam ruangan. Metodologi yang digunakan dalam menyelesaikan proyek akhir ini yaitu pengumpulan data dan study literatur. Pada tahapan awal yang dilakukan dalam pembuatan hardware dengan pemilihan komponen yang digunakan sesuai dengan fungsi dari alat proyek akhir ini. Sedangkan pada proses pemograman digunakan metode deep learning untuk mendeteksi objek (manusia) dalam gambar dan video secara real time. Dalam pendeteksian objek menggunakan Raspberry Pi yang hasil data perhitungan jumlah objek dan suhu ruangan yang terdeteksi akan ditampilkan di layar monitor. Dengan demikian, proyek akhir ini menghasilkan sistem kontrol fan AC secara otomatis dan perhitungan jumlah objek yang terdeteksi dalam ruangan. Sistem ini dapat digunakan di berbagai lingkungan, seperti kantor atau ruang perkuliahan agar dapat menciptakan ruangan yang nyaman bagi pengguna ruangan.

Kata Kunci : Image Processing, Raspbbery Pi, Fan AC, Sistem Kontrol

ABSTRACT

Controlling the AC fan and calculating the number of people in the room is one of the most influential aspects of creating an efficient and comfortable working environment. With the rapid development of technology, image processing comes with its own charm as a method that can be used to process images with input in the form of 2-dimensional images. The purpose of this final project is to create a system that uses image processing to control the AC fan and count the number of people entering the room. The methodology used in completing this final project is data collection and literature study. In the initial stages carried out in making hardware with the selection of components used in accordance with the function of this final project tool. While in the programming process, the deep learning method is used to detect objects (humans) in images and videos in real time. In detecting objects using Raspberry Pi, the results of object data and room temperature detected will be displayed on the monitor screen. Thus, this final project produces an automatic AC fan control system and calculates the number of objects detected in the room. This system can be used in various environments, such as offices or lecture halls in order to create a comfortable room for room users.

Keywords: Image Processing, Raspbberry Pi, AC Fan, Control System

KATA PENGANTAR

Assalamu'alaikum Warahmatullahi Wabarakatuh. Segala puji bagi Allah SWT yang telah melimpahkan taufiq dan hidayah-Nya kepada penulis, sehingga penulis dapat menyusun laporan proyek akhir ini dengan judul "**Pengontrolan Air Conditioner Berdasarkan Jumlah Orang Dalam Ruangan Berbasis Pengolahan Citra**" dan dapat menyelesaikan Program Studi Teknik Elektro dan Informatika di Politeknik Manufaktur Negeri Bangka Belitung. Sholawat serta salam semoga tercurahkan kepada Nabi Muhammad SAW beserta keluarga dan sahabat-sahabatnya, serta semoga semua umatnya senantiasa dapat menjalankan syari'at-syari'atnya, Aamiin.

Penulis menyadari sepenuhnya, bahwa dalam penyusunan laporan proyek akhir ini banyak kekurangan mengingat terbatasnya kemampuan penulis, namun berkat rahmat Allah SWT serta pengarahan dari berbagai pihak, akhirnya laporan proyek akhir ini dapat diselesaikan. Harapan penulis semoga laporan proyek akhir ini dapat bermanfaat untuk kepentingan bersama.

Pada kesempatan ini, ucapan terimakasih juga disampaikan kepada pihak yang telah membantu serta memberikan motivasi, saran dan kritik yang tentunya sangat diharapkan dalam menyelesaikan proyek akhir ini. Sehubungan dengan itu, penulis menyampaikan ucapan terima kasih kepada:

1. Ibunda dan Ayahanda tercinta serta seluruh keluarga yang dengan penuh keikhlasan dan kesungguhan hati memberikan bantuan moral dan spiritual yang tak ternilai harganya.
2. Bapak I Made Andik Setiawan, M.Eng., Ph.D. selaku Direktur Politeknik Manufaktur Negeri Bangka Belitung yang telah memberikan kemudahan dalam menyelesaikan pendidikan.
3. Bapak Aan Febriansyah, M.T dan Bapak Irwan, M.Sc., Ph.D. selaku dosen pembimbing yang telah membimbing, mengarahkan dan memberi saran-saran dalam pembuatan dan penyusunan laporan proyek akhir ini.

4. Bapak Indra Dwisaputra, M.T. selaku Kepala Prodi D-IV Teknik Elektronika Politeknik Manufaktur Negeri Bangka Belitung.
5. Bapak Zanu Saputra, M.Tr.T selaku Kepala Jurusan Teknik Elektro dan Informatika Politeknik Manufaktur Negeri Bangka Belitung.
6. Dosen dan Staf Pengajar di Politeknik Manufaktur Negeri Bangka Belitung yang telah mendidik, membina dan mengantarkan penulis untuk menempuh kematangan dalam berfikir dan berperilaku.
7. Teman-teman seperjuangan dan semua pihak yang telah memberikan support dan bantuannya selama mengerjakan proyek akhir ini.
8. Juicy Luicy dan Mahalini yang selalu menemani penulis mengerjakan proyek akhir ini dengan *playlist* nya yang berjudul “Mawar Jingga”, “Lampu Kuning”, “Bukan Orangnya” dan “Bawa Dia Kembali”.
9. Kepada seseorang yang pernah bersama penulis dan tidak bisa penulis sebutkan namanya. Terimakasih untuk patah hati yang diberikan saat proses pengerjaan proyek akhir ini. Ternyata perginya anda dari kehidupan penulis berikan cukup motivasi untuk terus maju dan berproses menuju pribadi yang lebih baik. Terimakasih telah menjadi bagian menyenangkan sekaligus menyakitkan dari pendewasaan ini. Pada akhirnya setiap orang ada masanya dan setiap masa ada orangnya.

Penulis menyadari bahwa penulisan proyek akhir ini masih jauh dari sempurna dikarenakan penulis adalah manusia biasa yang tidak luput dari kesalahan. Karena yang benar hanya datang dari Allah SWT dan yang salah datang dari penulis sendiri. Oleh karena itu, sangat diharapkan segala petunjuk, kritik dan saran yang membangun dari pembaca agar dapat menunjang pengembangan dan perbaikan penulis selanjutnya. Besar harapan penulis makalah proyek akhir dan alat yang dibuat ini dapat memberikan manfaat bagi pihak yang berkepentingan pada khususnya bagi perkembangan ilmu teknologi pada umumnya.

Sungailiat, 24 Juli 2024

Penulis

DAFTAR ISI

LEMBAR PENGESAHAN	Error! Bookmark not defined.
PERNYATAAN BUKAN PLAGIAT	Error! Bookmark not defined.
ABSTRAK	iii
<i>ABSTRACT</i>	iv
KATA PENGANTAR.....	v
DAFTAR ISI	vii
DAFTAR TABEL	x
DAFTAR GAMBAR	xi
LAMPIRAN.....	xii
BAB I	1
1.1 Latar Belakang Masalah.....	1
1.2 Perumusan Masalah	2
1.3 Tujuan Proyek Akhir	3
1.4 Batasan Masalah.....	3
BAB II	4
2.1 Sistem Otomatis	4
2.1.1 Sistem Kontrol	4
2.1.2 Dampak <i>Air Conditioner</i> bagi Pengguna Ruangan	5
2.2 Sistem Berbasis <i>Image Processing</i> (Pengolahan Citra).....	6
2.2.1 Kontroler <i>Raspberry Pi</i>	8
2.2.2 Algoritma YOLO V5 Sebagai Deteksi Objek (Manusia)	10
BAB III	16
3.1 <i>Flow Chart</i> Perencanaan dan Pembuatan Alat.....	16

3.2	Pengumpulan Data	17
3.3	Desain <i>Hardware</i> dan <i>Software</i> Sistem Kontrol	17
3.4	Pembuatan <i>Hardware</i> dan <i>Software</i> Sistem Kontrol.....	18
3.5	Uji Coba	19
3.6	Perbaikan.....	19
3.7	Pengolahan Data.....	20
3.8	Pembuatan Laporan.....	20
BAB IV	21
4.1	Diagram Blok	21
4.2	<i>Hardware</i> Sistem Kontrol	22
4.2.1	Scematic <i>Hardware</i> Sistem Kontrol.....	22
4.2.2	Pemilihan Komponen Sistem Kontrol	23
4.3	Komponen Sistem Kontrol.....	24
4.3.1	Modul DHT 22	24
4.3.2	Modul IR Transmitter.....	26
4.3.3	<i>Raspberry Pi</i>	30
4.3.4	NodeMCU ESP8266	30
4.4	Komponen Sistem Pengambilan Data.....	32
4.4.1	Modul <i>IR Receiver</i>	32
4.5	Pengujian Algoritma Deteksi Objek	33
4.5.1	YOLOV5.....	33
4.6	Pengujian Kontrol <i>Fan AC</i> Berdasarkan Jumlah Orang	37
4.7	Tampilan Monitor <i>Display</i> Pada <i>Raspberry Pi</i>	39
BAB V	40
5.1	Kesimpulan	40

5.2	Saran.....	41
	DAFTAR PUSTAKA.....	42



DAFTAR TABEL

Tabel 4. 1 Tabel Kriteria Komponen Yang Digunakan	23
Tabel 4. 2 Spesifikasi Pin Modul DHT 22	24
Tabel 4. 3 Spesifikasi Pin Modul IR Transmitter	26
Tabel 4. 4 Pengiriman Data ke AC Dengan Jarak	27
Tabel 4. 5 Pengiriman Data ke AC Dengan Sudut	29
Tabel 4. 6 Spesifikasi Pin Modul NodeMCU.....	31
Tabel 4. 7 Spesifikasi Pin Modul IR Receiver	32
Tabel 4. 8 Hasil Pengujian Deteksi Objek	36
Tabel 4. 9 Hasil Pengujian Kontrol Fan AC.....	37

DAFTAR GAMBAR

Gambar 2. 1 <i>Air Conditioner</i>	6
Gambar 2. 2 Raspberry Pi	10
Gambar 2. 3 Proses Konvolusi Gambar	11
Gambar 3. 1 Flow Chart Proses Pelaksanaan Penelitian	16
Gambar 4. 1 Diagram Blok Hardware	21
Gambar 4. 2 Scematic Sistem Kontrol	22
Gambar 4. 3 Software Rangkaian DHT 22	24
Gambar 4. 4 Pengujian Serial Monitor DHT22	25
Gambar 4. 5 Software Rangkaian IR Transmitter	26
Gambar 4. 6 Tampilan Uji Coba Raspberry Pi	30
Gambar 4. 7 Software Rangkaian Pengujian NodeMCU ESP8266	31
Gambar 4. 8 Software Rangkaian IR Receiver	32
Gambar 4. 9 Software Rangkaian IR Receiver	33
Gambar 4. 10 Hasil Pengujian YOLOV5 dalam Jarak Dekat	34
Gambar 4. 11 Hasil Pengujian YOLOV5 dengan Jarak Berbeda	34
Gambar 4. 12 Hasil Pengujian 1 Grid YOLOV5	35
Gambar 4. 13 Hasil Pengujian 2 Grid YOLOV5	35
Gambar 4. 14 Tampilan Layar Monitor	39

LAMPIRAN

Lampiran 1 Daftar Riwayat Hidup.....	46
Lampiran 2	48
Lampiran 3	50
Lampiran 4	53



BAB I

PENDAHULUAN

1.1 Latar Belakang Masalah

Dengan kemajuan teknologi dan pembangun yang berkembang semakin pesat, penggunaan daya listrik yang ada di Indonesia juga meningkat dengan sangat tinggi. Hal ini disebabkan oleh banyaknya penambahan beban dan penggunaan listrik yang berlebihan [1]. Dengan pasokan listrik yang terbatas, kebutuhan listrik terus meningkat. Akibatnya, perlu adanya pengurangan penggunaan energi listrik. Menurut laporan DEPDIKNAS [2], AC masih menjadi komponen terbesar dalam konsumsi listrik di bangunan gedung, mencapai 50% dari total penggunaan listrik nasional. Salah satu solusi efektif untuk mengurangi pemborosan energi adalah dengan mematikan AC saat tidak sedang digunakan [2], [3].

Suhu dan kelembapan dalam suatu ruangan sangat berpengaruh pada efektifitas kegiatan, maka dari itu kegiatan atau bahkan bekerja di tempat yang terlalu panas dapat menyebabkan tubuh lebih cepat merasa lelah karena performa fisik menurun. Sebaliknya, bekerja di lingkungan yang terlalu dingin dapat membatasi gerakan motorik karena tegangan fisik, yang dapat mengganggu konsentrasi saat menjalankan tugas. Keseimbangan suhu yang tepat di tempat kerja sangat penting untuk mendukung kenyamanan dan produktivitas pengguna ruangan [4]. Standar yang ditetapkan oleh SNI 03-6572-2002 ada beberapa tingkatan temperatur yang nyaman untuk orang Indonesia atas tiga bagian, yaitu 20,5°C - 22,8°C, nyaman optimal, antara temperatur efektif 22,8°C - 25,8°C, dan hangat nyaman, antara temperatur efektif 25,8°C – 27,1°C [5].

Saat ini, kemajuan teknologi dalam pengolahan citra (*image processing*) telah menjadi magnet bagi banyak orang untuk dipelajari karena relevansinya dalam kehidupan sehari-hari. Seiring dengan perkembangan teknologi ini, kreativitas dalam mengenali dan mengidentifikasi objek juga semakin berkembang, menjadi bagian integral dari pengolahan citra *modern* [6]. *Image Processing* adalah suatu metode untuk mengolah atau mengubah gambar dua dimensi. Proses ini

melibatkan pengolahan sinyal dari gambar masukan untuk menghasilkan gambar lain dengan menggunakan metode tertentu. Metode-metode ini bisa mencakup filterisasi, segmentasi, ekstraksi fitur, dan teknik lain yang bertujuan untuk memperbaiki kualitas gambar, menganalisis konten visual, atau mengekstraksi informasi penting dari gambar tersebut [7].

Beberapa studi sebelumnya telah menginvestigasi topik ini, seperti yang telah dilakukan oleh Aqmaluh Agung Pradana yang merancang alat yang dapat berfungsi sebagai *remote control* dengan menggunakan teknologi IoT dan *infrared* untuk menggantikan *remote* konvensional AC, yang memungkinkan seseorang untuk mengontrol AC dari mana saja dan kapan saja selama mereka memiliki akses ke internet [8]. Penelitian serupa juga dilakukan oleh Hamam Nasirrudin dan Budi Prijo Sembodo yang menggunakan sensor ultrasonik HCSR04 untuk mendeteksi sesuatu dalam sistem dan sinyal dari sensor diterima oleh Arduino Uno dan diolah menjadi data. Kemudian data yang dihasilkan akan memerintahkan AC dan lampu [9].

Berdasarkan penelitian sebelumnya penulis berinovasi untuk membuat alat yang bisa digunakan untuk mengontrol *fan* AC berdasarkan jumlah orang yang terdeteksi masuk ke dalam ruangan menggunakan *image processing*. Pada sistem ini *image processing* digunakan untuk mengamati dan mengevaluasi suatu objek yang berfungsi untuk menghitung jumlah orang yang masuk dan keluar dari ruangan.

1.2 Perumusan Masalah

Berdasarkan latar belakang dari proyek akhir ini membentuk rumusan masalah berikut:

1. Bagaimana merancang dan membuat alat yang dapat mengendalikan *fan air conditioner* (AC) secara otomatis?
2. Bagaimana membuat sistem yang dapat membaca jumlah orang dalam ruangan menggunakan *image processing*?
3. Bagaimana hasil uji pengontrolan *fan* AC berdasarkan data dari *image processing*?

1.3 Tujuan Proyek Akhir

Adapun tujuan dari proyek akhir ini adalah:

1. Merancang dan membuat alat yang dapat mengendalikan *fan air conditioner* (AC) secara otomatis.
2. Membuat sistem yang dapat membaca jumlah orang dalam ruangan menggunakan *image processing*.
3. Menguji pengontrolan *fan AC* berdasarkan data dari *image processing*.

1.4 Batasan Masalah

Berikut adalah batasan masalah dari proyek akhir ini:

1. Alat ini hanya bisa digunakan menggunakan AC merek *Panasonic*.
2. Jumlah orang yang terdeteksi maksimal 10 orang.
3. Suhu ruangan dijaga dalam suhu referensi 20 - 25°C.
4. Perubahan jumlah orang yang terdeteksi mengganti kecepatan *fan AC*.
5. Jumlah orang dan pengaturan *fan AC* hanya sebagai uji coba.

BAB II

DASAR TEORI

2.1 Sistem Otomatis

Otomatis mempunyai arti dengan bekerja sendiri atau dengan sendirinya. Pengertian pengaturan otomatis atau sistem pengaturan otomatis berasal dari tiga suku kata yaitu sistem, pengaturan dan otomatis. Sistem adalah sebuah susunan komponen-komponen fisik yang saling terhubung dan membentuk satu kesatuan untuk melakukan aksi tertentu. Pengaturan adalah suatu aktivitas mengatur, mengendalikan, mengarahkan, memerintah. Sedangkan otomatis adalah dengan bekerja sendiri atau dengan sendirinya. Dalam hal ini istilah pengaturan atau kontrol mengandung tiga aspek atau unsur utama yaitu rencana yang jelas, dapat melakukan pengukuran, dan dapat melakukan tindakan. Pengaturan otomatis yang dimaksud adalah “membuat sesuatu sesuai dengan harapan ataupun rencanan kita dan juga berjalan dengan sendirinya tanpa campur tangan manusia secara langsung” maka kita dapat menganggap suatu sistem kontrol otomatis adalah suatu sistem yang dapat membuat agar keluaran (*output*) sistem sesuai dengan rencana dan keinginan yang diharapkan [10].

2.1.1 Sistem Kontrol

Sistem adalah entitas kompleks yang terdiri dari serangkaian prosedur atau komponen yang saling terhubung untuk mencapai tujuan tertentu. Apabila salah satu komponen mengalami kerusakan atau gangguan, sistem dapat mengalami disfungsi atau tidak beroperasi sebagaimana mestinya. Ketergantungan antar-komponen dalam sistem memastikan bahwa setiap elemen berperan penting dalam menjaga kinerja keseluruhan sistem [11]. Sistem kontrol adalah sistem pengaturan atau pengendalian terhadap satu atau beberapa besaran (variabel, parameter) sehingga berada pada suatu harga atau dalam satu rangkuman harga (*range*)

tertentu. Sistem kontrol (control system) merupakan suatu kumpulan cara atau metode yang dipelajari dari kebiasaan-kebiasaan manusia dalam bekerja, dimana manusia membutuhkan suatu pengamatan kualitas dari apa yang telah mereka kerjakan sehingga memiliki karakteristik sesuai dengan yang diharapkan pada mulanya. Dari penjelasan tentang sistem kontrol dapat disimpulkan sistem kontrol adalah gabungan dari beberapa komponen yang saling berhubungan yang bekerja secara terus menerus untuk mencapai tujuan tertentu sesuai yang diharapkan pada mulanya [12].

2.1.2 Dampak *Air Conditioner* bagi Pengguna Ruangan

Temperatur ruangan sangat penting untuk menciptakan lingkungan yang nyaman dan produktif. Istilah "suhu ruangan" merujuk pada suhu di dalam suatu tempat yang sering digunakan untuk berbagai aktivitas seperti pertemuan, presentasi, atau diskusi.

Bekerja dalam lingkungan yang kurang mendukung, seperti suhu terlalu tinggi atau kelembapan yang tinggi, dapat signifikan memengaruhi produktivitas serta mengurangi kemampuan fisik, bahkan menyebabkan kelelahan yang cepat terjadi. Sebaliknya, ruangan yang terlalu dingin juga dapat mengurangi fleksibilitas motorik tubuh karena kekakuan fisik yang ditimbulkan. Oleh karena itu, suhu yang sangat memungkinkan untuk melakukan aktifitas di dalam ruangan berkisar antara 18°C-28°C dan kelembaban udara sekitar 40%-60% [13].

Penggunaan *Air Conditioner* (AC) memiliki potensi dampak buruk bagi penghuni ruangan, jika tidak dioperasikan sesuai dengan standar yang ditetapkan. AC dapat mengganggu sistem udara dalam bangunan, yang pada akhirnya dapat menyebabkan kondisi yang dikenal sebagai *Sick Building Syndrome* (SBS). SBS merujuk pada gangguan kesehatan yang dialami seseorang ketika berada di dalam gedung atau bangunan, yang disebabkan oleh ventilasi yang buruk dan kontaminasi udara. Polutan udara seperti debu, asap, dan zat-zat kimia lainnya dapat menyebabkan atau memperburuk SBS. Maka, penting untuk mengelola

penggunaan AC dengan baik agar tidak hanya menjaga kenyamanan tetapi juga kesehatan penghuni gedung [14].

Dampak dari penggunaan *Air Conditioner* tak hanya berpengaruh pada kesehatan, melainkan pemborosan konsumsi energi listrik. Penggunaan *air Conditioner* yang tepat dapat menjadikan *Air Conditioner* awet dan hemat listrik, sebaliknya penggunaan *Air Conditioner* yang tidak sesuai dapat memperpendek umur *Air Conditioner* dan pemborosan pada penggunaan listrik. Pemborosan energi terjadi ketika kompresor AC bekerja secara berlebihan, menyebabkan penyerapan daya listrik yang besar. Hal ini mengurangi efisiensi AC dan mengakibatkan pemborosan energi listrik yang tidak perlu. Penting untuk menggunakan AC dengan cara yang tepat, seperti mempertahankan suhu yang sesuai dan melakukan perawatan teratur, untuk mengoptimalkan kinerja perangkat dan menghemat energi listrik secara signifikan [15].



Gambar 2. 1 *Air Conditioner*

<https://www.electriq.co.uk/p/iqool24/electriq-iqool24-air-conditioner>

2.2 Sistem Berbasis *Image Processing* (Pengolahan Citra)

Image Processing adalah teknologi visual yang memungkinkan pengamatan dan analisis suatu objek tanpa interaksi langsung dengan objek

tersebut, melalui variasi citra warna yang diolah menggunakan perangkat lunak. Perkembangan teknologi dalam bidang pengolahan citra (*image processing*) terus mengalami kemajuan pesat.

Teknik-teknik baru telah dikembangkan untuk mempermudah pekerjaan manusia dalam pengolahan dan analisis citra, bahkan ketika citra yang digunakan tidak ideal karena berbagai gangguan seperti bayangan, foto atau gambar yang buram, atau ketidakjelasan objek. Hal ini dapat mengakibatkan masalah dan mempengaruhi hasil interpretasi serta perencanaan yang dilakukan. Oleh karena itu, berbagai teknik pengolahan citra diperlukan untuk mendapatkan citra yang optimal dan akurat [16].

Sesuai dengan perkembangan komputer, pengolahan citra mempunyai dua tujuan utama, yaitu :

- 1) Memperbaiki kualitas citra, dimana citra yang dihasilkan dapat menampilkan informasi secara jelas. Hal ini berarti manusia sebagai pengolah informasi (*human perception*).
- 2) Mengekstraksi informasi ciri yang menonjol pada suatu citra, dimana hasilnya adalah informasi citra dimana manusia mendapatkan informasi ciri dari citra secara numerik atau dengan kata lain *computer* (mesin) melakukan interpretasi terhadap informasi yang ada pada citra melalui besar-besaran data yang dapat dibedakan secara jelas [17].

Peneliti dari berbagai perguruan tinggi di Indonesia telah melakukan banyak penelitian tentang *image processing*, khususnya dalam konteks proyek akhir mereka. Hasil-hasil penelitian ini berkontribusi signifikan dalam mengembangkan teknologi dan aplikasi baru di bidang *image processing*. Beberapa penelitian sebelumnya, *image processing* sudah dibahas dalam berbagai pokok permasalahan.

Penelitian yang berkaitan dengan *image processing* yaitu penentuan kualitas warna batu *blue sapphire* dengan *image processing* menggunakan metode RGB to HVS. Penelitian ini dilakukan dengan menggunakan *microscope* digital dan *Raspberry Pi*, yang dapat digunakan secara portabel untuk memproses gambar batu *blue sapphire*. Adapun tujuan dari penelitian ini adalah untuk mengatasi

permasalahan dalam menentukan kualitas warna pada batu *blue sapphire*, dikarenakan banyak orang yang kesulitan untuk menentukan warna yang berkualitas baik pada batu permata *blue sapphire* [18].

Penelitian selanjutnya yaitu pendeteksi banjir dengan *image processing* berbasis *Convolutional Neural Network* (CNN) pada kamera pengawas. Teknologi *image processing* memungkinkan pengenalan gambar dari input. Algoritma *Convolutional Neural Network* (CNN) menonjol dalam penelitian ini berkat struktur lapisan neuron yang dalam dan proses pelatihan data yang berulang. Di sisi lain, pemrosesan citra yang diterima dari sensor ultrasonik dan kamera pengawas dapat digunakan untuk mengukur ketinggian air, yang pada gilirannya memberikan pembacaan yang akurat terkait potensi bencana banjir. Sehingga dapat memberikan peringatan dini untuk dapat mengurangi dan/atau bahkan menghilangkan ancaman bencana [19].

Kemudian penelitian yang dilakukan yaitu sistem parkir menggunakan teknik *image processing* yang bertujuan untuk mengembangkan sistem pemetaan slot parkir yang efektif untuk pengelola dan pengguna apartemen menggunakan spesifikasi segmentasi analisis *image processing*. Desain sistem ini berfokus pada pengolahan plat kendaraan yang terdeteksi oleh kamera pengawas melalui analisis segmentasi gambar. Proses data dilakukan melalui antarmuka Visual Studio yang terkoneksi dengan MySQL sebagai server basis data, sementara aplikasi *Blynk* digunakan untuk memvisualisasikan status slot parkir dan lokasi mereka secara *real-time* [20].

2.2.1 Kontroler *Raspberry Pi*

Raspberry Pi adalah sebuah komputer papan tunggal dengan ukuran yang mirip dengan kartu kredit yang memiliki kemampuan mampu menjalankan berbagai aplikasi perkantoran, permainan komputer, serta memutar media, termasuk video beresolusi tinggi [21]. Perangkat ini dapat dianggap sebagai versi *Personal Computer* (PC) yang bisa diprogram menggunakan bahasa pemrograman

Python untuk melakukan penangkapan gambar apabila terdeteksi kedatangan seseorang [22].

Salah satu keunggulan dari *Raspberry Pi* adalah kemampuannya dalam *home automation* dimana perangkat ini dapat mengontrol lampu, penyiraman tanaman, kipas angin, AC yang dapat diatur dari LCD maupun *handphone*. Hal ini menjadikan *Raspberry Pi* sangat cocok digunakan sebagai alat pendidikan untuk berbagai usia dan tingkat keterampilan. Minat yang tinggi pada *Raspberry Pi* jauh melampaui ekspektasi awal, menunjukkan popularitas dan kegunaan yang luar biasa dari perangkat ini dalam berbagai aplikasi teknologi [23].

Penelitian mengenai sistem kontrol *Raspberry Pi* sudah banyak dilakukan, terutama pengontrolan yang menggunakan sistem *image processing*. Ada banyak persamaan dan perbedaan, serta topik masalah yang dibahas, mikrokontroler yang digunakan, dan komponen lainnya yang membentuknya.

Penelitian yang berkaitan dengan sistem kontrol *Raspberry Pi* menggunakan *image processing* diantaranya yaitu alat identifikasi jenis kayu berbasis *image processing* dengan metode *K-Nearest Neighbor* (KNN). Dalam pengembangannya, *image processing* digunakan untuk memproses data gambar yang kemudian dianalisis menggunakan metode KNN. Sistem ini dirancang dengan *Raspberry Pi* sebagai pengganti laptop, dimana program *Python* dijalankan untuk melakukan proses analisis menggunakan metode HSV, GLCM, dan KNN yang telah diimplementasikan di dalamnya. Alat ini bertujuan untuk memudahkan pengguna dalam membedakan delapan jenis kayu: glugu, jati, kamper, kruing, meranti, nangka, randu, dan sengon, berdasarkan analisis gambar [24].

Kemudian penelitian pada implementasi deteksi warna pada *public monitoring system* dengan *Raspberry Pi* yang merupakan inovasi dalam perangkat pengawasan yang hemat energi. Tujuan utama penelitian ini adalah mengidentifikasi warna merah untuk deteksi kebakaran dan warna coklat untuk area persawahan dalam gambar, yang memberikan informasi penting untuk pencegahan kebakaran. *Raspberry Pi* dipilih karena kecepatan prosesor yang mencukupi untuk aplikasi pemrosesan gambar. Pengembangan deteksi kebakaran dengan *Raspberry Pi* mengilustrasikan potensi aplikasi ini, terutama dengan menggunakan

Raspberry Pi Camera untuk memungkinkan pemrosesan gambar dan IoT secara *real-time* dalam mendeteksi kondisi tertentu [25].

Selanjutnya yaitu rancang bangun alat deteksi fertilitas telur unggas berbasis *image processing* dengan tujuan mengurangi kesalahan dalam menentukan fertilitas. Alat ini dikembangkan dengan metode *area contours*, yang memungkinkan mesin untuk meniru kemampuan penglihatan manusia dalam mendeteksi kontur pada gambar. Pengambilan gambar dilakukan menggunakan kamera *Raspberry Pi* sebagai bagian dari implementasi sistem ini [26].



Gambar 2. 2 *Raspberry Pi*

<https://www.extremetech.com/computing/311086-Raspberry-pi-4-now-available-with-8gb-of-ram-64-bit-os>

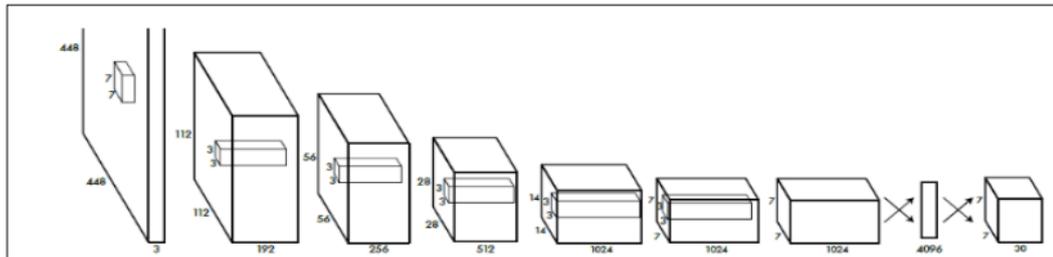
2.2.2 Algoritma YOLO V5 Sebagai Deteksi Objek (Manusia)

Pendeteksian objek merupakan teknologi yang sangat berguna dalam mengidentifikasi dan menemukan objek berdasarkan karakteristik seperti warna, bentuk, dan data yang terkumpul. Sistem pendeteksi *You Only Look Once* (YOLO) menonjol karena kecepatan dan akurasi yang tinggi dalam mengenali objek pada gambar atau citra, sehingga sangat cocok untuk aplikasi pendeteksian objek secara *real-time* pada video.

Kecepatan memegang peranan krusial dalam pendeteksian objek secara *real-time*, terutama karena video dapat mengolah lebih dari 24 *frame per second* (FPS). Ketika proses pendeteksian objek terlalu lambat, setiap *frame* mengalami penundaan, yang berpotensi merusak kualitas output video secara keseluruhan [27].

YOLOV5 yang merupakan versi terbaru dari YOLO, telah diperbarui dari YOLOV4 dengan peningkatan signifikan dalam kecepatan eksekusi hingga 10 fps. Algoritma YOLOV5 mampu menangani berbagai tantangan dalam deteksi objek seperti tumpang tindih, variasi ukuran objek, dan perubahan kondisi pencahayaan. Ini menjadikannya sebagai metode deteksi objek yang akurat dan efisien yang cocok untuk aplikasi deteksi objek secara *real-time* pada perangkat dengan sumber daya terbatas. Contoh penerapannya meliputi sistem keamanan, kendaraan tanpa pengemudi, dan deteksi objek dalam video [28].

Dalam menggunakan YOLO, objek dibagi menjadi beberapa bagian yang disebut *grid*. Setiap *grid* dapat terdiri dari 9 *sub-grid* atau lebih, di mana proses konvolusi dilakukan untuk menghasilkan prediksi garis. YOLO menggunakan informasi dari garis prediksi ini untuk menemukan dan mengidentifikasi objek yang terletak di dalam *grid-grid* tersebut.



Gambar 2. 3 Proses Konvolusi Gambar

Setelah *grid* ditempatkan di atas objek, dilakukan proses perhitungan matriks di setiap *grid* untuk mengisi matriks tersebut. Proses ini melibatkan penentuan jumlah objek di dalam *grid*, koordinat *grid*, titik sudut, dan kriteria pencarian lainnya. Selanjutnya, matriks diisi dengan prediksi objek yang memiliki probabilitas tinggi di dalam *grid*, menangkap karakteristik khusus yang dimiliki oleh objek yang sedang dicari. Setelah mendapatkan hasil prediksi di setiap *grid*, proses eliminasi prediksi dengan probabilitas rendah digunakan untuk menghasilkan satu prediksi akhir [29].

2.2.2.1 Pengembangan Sistem Deteksi Objek

2.2.2.1.1 Monitoring Ruangan untuk Deteksi Manusia Berbasis CNN dengan Fitur *Push Notification* (Penelitian Tahun 2019)

Mahasiswa dari Program Studi Teknik Informatika, Universitas Ma Chung melakukan penelitian dengan judul “Monitoring Ruangan untuk Deteksi Manusia Berbasis CNN dengan Fitur *Push Notification*”. Latar belakang dari penelitian ini yaitu kasus pencurian yang umumnya di lingkungan tempat tinggal seperti pencurian rumah kosong, pencurian kendaraan bermotor, hingga pembobolan rumah. Sehingga memerlukan tindakan yang dapat dilakukan dengan monitoring rumah agar pemilik rumah dapat mengawasi rumah dari jarak jauh. CCTV dan IP *Camera* yang dinilai memiliki beberapa kekurangan seperti mudah dipengaruhi interferensi, bisa disadap, memerlukan kabel dan memiliki *delay* yang bergantung pada panjang kabel CCTV atau spesifikasi pada IP *Camera*. Sehingga dalam penelitian ini, dibuat sebuah sistem monitoring ruangan yang dapat mendeteksi manusia dengan menggunakan perangkat *Raspberry Pi 3* dan akan mengirimkan *push notification* ketika sistem mendeteksi gerakan manusia dalam video [30].

Berdasarkan penelitian yang telah dilakukan dalam proses pembuatan sistem “Monitoring Ruangan untuk Deteksi Manusia Berbasis CNN dengan Fitur *Push Notification*” dapat disimpulkan bahwa :

- 1) Bobot YOLOv2 yang dilatih dapat digunakan untuk mendeteksi objek berupa manusia secara *realtime* dengan nilai akurasi sebesar 72,1%, sensitivitas sebesar 56,49% dan spesifitas sebesar 94,9%.
- 2) *Push notification* yang disampaikan kepada pengguna dilakukan dengan *login* pada aplikasi di *smartphone*.

2.2.2.1.2 Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk *Mobile Robot* (Penelitian Tahun 2021)

Institut Teknologi Sepuluh Nopember merupakan institusi yang berada di Surabaya, yang memiliki 39 Departemen diantaranya adalah Departemen Teknik

Elektro melakukan penelitian yaitu “Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk *Mobile Robot*”. Dalam perkembangan *mobile robot* yang dapat berfungsi sebagai pengikut garis (*line follower*), peneliti melakukan pengembangan dengan menciptakan *mobile robot* yang berfungsi sebagai pengikut manusia. Dalam hal ini *mobile robot* harus mampu mengenali objek “manusia” agar dapat mengikuti objek tersebut sebagai asisten yang dapat membantu manusia dalam melakukan pekerjaannya, dengan menggunakan sensor kamera serta metode YOLO yang berbasis CNN.

Dalam penelitian ini, beberapa baris kode dalam pemrograman bahasa *Python* pada YOLOv4 diubah dengan tujuan untuk menambahkan jangkauan deteksi. Perubahan ini dilakukan untuk memperluas kemampuan olah data pada *mobile robot*, sehingga memungkinkan robot untuk mendeteksi objek dari jarak yang lebih jauh dan dengan akurasi yang lebih tinggi. Dengan demikian, modifikasi ini diharapkan dapat meningkatkan wawasan dan kinerja robot dalam berbagai tugas yang memerlukan analisis visual dan pemrosesan data yang lebih canggih [31].

Berdasarkan pengujian sistem yang dilakukan dapat disimpulkan bahwa:

- 1) Hasil uji performa YOLOv4 diperoleh nilai mAP sebesar 87,03% dan waktu pemrosesan selama 116 detik dengan jumlah total gambar sebanyak 904 gambar.
- 2) Modul dapat mendeteksi baik *single object* ataupun *multiple object*.
- 3) Pengujian deteksi didalam *range* sudut tertentu dapat mengklasifikasikan objek yang berada di dalam dan luar *range*.

2.2.2.1.3 Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO (Penelitian Tahun 2022)

Salah satu sistem deteksi objek manusia yang telah dibuat oleh mahasiswa dari Jurusan Teknik Mesin Politeknik Bandung dengan judul “Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO”. Penelitian ini dilakukan dengan latar belakang penggunaan mesin yang dapat meningkatkan kinerja manusia dan mengurangi

resiko terjadinya *human error* sehingga menghasilkan pekerjaan yang lebih konsisten. Pada penelitian ini dilakukan implementasi pengolahan citra pada quadcopter dengan menggunakan algoritma YOLO untuk mendeteksi objek. Pengimplementasian ini diharapkan dapat membantu beberapa pekerjaan manusia, seperti membantu proses evakuasi bencana, pemetaan lahan, pemantauan dari udara, dan lainnya. Berdasarkan pengujian sistem yang telah dilakukan diperoleh kesimpulan sebagai berikut :

- 1) Sistem pendeteksian manusia yang telah dibuat berhasil mendeteksi objek berupa manusia.
- 2) Nilai mAP yang diperoleh dari hasil pengujian kinerja YOLOv5 sebesar 86.8% yang berarti akurasi sistem deteksi objek sudah baik.
- 3) Pengujian sistem berdasarkan jarak dari kamera ke objek dengan rentang jarak 0.5 m sampai 4.5 m.
- 4) Sistem dapat mendeteksi objek lebih dari satu dalam satu *frame* [32].

2.2.2.1.4 Rancang Bangun Sistem Deteksi Manusia dengan YOLO pada Video CCTV (Penelitian Tahun 2024)

Penelitian dengan judul “Rancang Bangun Sistem Deteksi Manusia dengan YOLO pada Video CCTV” dilakukan oleh mahasiswa dari Universitas Nusantara PGRI Kediri. Penelitian ini dilakukan dengan tujuan untuk mengembangkan sistem deteksi objek manusia berbasis *computer vision* pada CCTV untuk meningkatkan efisiensi penyimpanan data. Masalah klasik sistem pada CCTV yang dinilai memiliki penyimpanan *overload* akibat rekaman terus-menerus tanpa seleksi. Sehingga peneliti merancang sebuah sistem yang dapat mengoptimalkan penyimpanan dengan hanya merekam saat ada kehadiran manusia dengan menerapkan algoritma YOLO.

Berdasarkan hasil perancangan dan pengujian alat Rancang Bangun Sistem Deteksi Manusia dengan YOLO pada Video CCTV yang telah dilakukan diperoleh kesimpulan sebagai berikut :

- 1) Hasil pengujian menunjukkan bahwa modul berhasil mendeteksi semua objek manusia dengan benar, sehingga menghasilkan akurasi 100%.
- 2) Menerapkan metode YOLOv8 untuk deteksi objek manusia menggunakan dataset dari *Google Image/Open Image* [33].

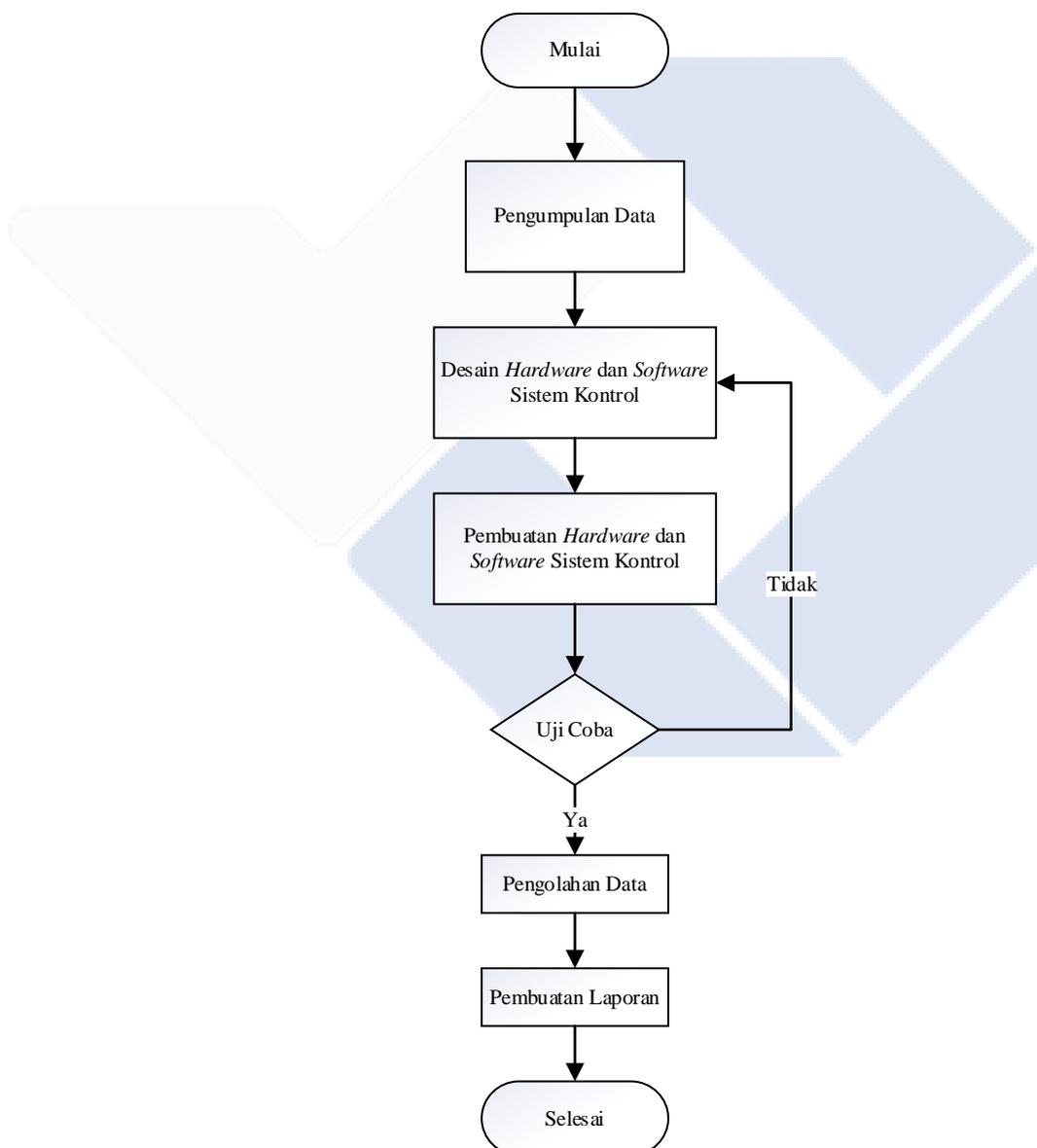


BAB III

METODE PELAKSANAAN

Dalam proses penelitian proyek akhir ini, terdiri dari beberapa tahapan yang dirancang untuk mempermudah selama pengerjaan.

3.1 *Flow Chart* Perencanaan dan Pembuatan Alat



Gambar 3. 1 Diagram Alir Proses Pelaksanaan Penelitian

3.2 Pengumpulan Data

Dalam pengerjaan proyek akhir membutuhkan data-data yang meliputi *study literatur*. *Study literatur* adalah proses pembelajaran yang mencakup pencarian informasi dari berbagai sumber, seperti internet, buku-buku, jurnal ilmiah, dan publikasi lainnya. Proses ini bertujuan untuk mengumpulkan informasi yang relevan dan terbaru untuk mendukung penelitian, memberikan landasan teoritis, serta memperkaya pemahaman tentang topik yang diteliti.

1) *Study Literatur*

Study literatur adalah proses pengumpulan jurnal yang diperoleh melalui pencarian di *Google*. Selain itu, penulis mencari referensi dan tutorial di *YouTube* terkait alat yang akan dibuat. Penulis juga menggunakan beberapa referensi dari jurnal dan penelitian lainnya.

3.3 Desain *Hardware* dan *Software* Sistem Kontrol

Desain *hardware* dan pemograman akan digunakan dalam penelitian ini untuk merencanakan pembuatan alat dan kontrol. Dalam proses perencanaan rancangan *hardware*, hal-hal berikut dilakukan:

1) Komponen yang akan digunakan pada pembuatan sistem kontrol.

Pada pembuatan sistem kontrol, komponen yang digunakan yaitu *Raspberry Pi* untuk menjalankan pengolahan citra guna mendeteksi keberadaan manusia dan *NodeMCU ESP8266* yang digunakan sebagai kontroler kedua untuk mengatur *IR Transmitter* yang mengolah data *image processing*.

Sedangkan untuk perencanaan pemograman sebagai berikut:

1) Perencanaan pengaturan *fan AC* berdasarkan jumlah orang dalam ruangan.

Perencanaan dalam pengaturan *fan AC* memiliki beberapa opsi, yaitu :

1. Apabila terdeteksi 1-3, maka *fan AC* akan berubah setelan ke *speed low*.
2. Apabila terdeteksi 4-6, maka *fan AC* akan berubah setelan ke *speed medium*.
3. Apabila terdeteksi 7-10, maka *fan AC* akan berubah setelan ke *speed high*.

Dalam semua opsi tersebut, suhu AC akan terjaga dalam suhu referensi yang berkisar antara 20 - 25°C.

2) Program *image processing*

Program *image processing* yang dilakukan menggunakan bahasa *Python* untuk memproses data yang telah didapatkan pada sistem pengolahan citra berupa jumlah objek (manusia) dari hal gambar yang terdeteksi oleh *camera webcam*.

3.4 Pembuatan *Hardware* dan *Software* Sistem Kontrol

Pembuatan *hardware* dilakukan dengan cara pemilihan tipe komponen yang dibutuhkan sesuai dengan fungsi alat. Seluruh komponen yang digunakan dalam pembuatan *hardware* ini harus dipesan secara online karena sedikit susah didapatkan dan memiliki resiko kerusakan.

Adapun tahapan dalam pembuatan *hardware* ini antara lain :

- 1) Menghubungkan komunikasi *Raspberry Pi* dengan *NodeMCU* menggunakan USB.
- 2) Menghubungkan *camera webcam* dengan *Raspberry Pi*.
- 3) Menghubungkan *NodeMCU ESP8266* dengan sensor *IR Transmitter*.

Sedangkan untuk pembuatan program dilakukan dengan menggunakan metode deteksi objek berbasis *deep learning* yang dikembangkan untuk mendeteksi dalam gambar dan video secara *real time*. Pemograman ini berfungsi sebagai kontrol *fan AC*. Dalam proses pemograman menggunakan bahasa *Python*.

Tahapan dari pembuatan program ini meliputi :

- 1) Pemograman *NodeMCU ESP8266* untuk mendapatkan data dari remot AC dengan menggunakan sensor *IR Receiver*.
- 2) Pemograman *Raspberry Pi* untuk mendeteksi objek dan menampilkan suhu di layar monitor.
- 3) Pemograman *NodeMCU ESP8266* untuk mengontrol AC menggunakan *IR Transmitter*.

3.5 Uji Coba

Langkah selanjutnya adalah melakukan uji coba terhadap penelitian setelah selesai merancang *hardware* dan pemrograman. Uji coba ini bertujuan untuk mengevaluasi keberhasilan alat yang telah dibuat dalam konteks penelitian tersebut.

Dalam proses uji coba yang dilakukan meliputi :

- 1) Uji coba pengambilan data remot AC menggunakan *NodeMCU ESP8266* dengan sensor *IR Receiver* dan kemudian dicek kembali data yang dihasilkan menggunakan *IR Transmitter* ke AC. Jika tidak berhasil maka dilakukan pengambilan data kembali.
- 2) Uji coba mendeteksi objek berbasis *deep learning* yang dikembangkan untuk mendeteksi objek dalam gambar dan video secara *real time* lalu melihat apakah ada objek yang terdeteksi atau tidak.
- 3) Uji coba program. Uji coba ini dilakukan untuk mengecek apakah data dari objek yang terdeteksi terkirim ke sensor *IR Transmitter* untuk mengatur AC.

3.6 Perbaikan

Proses ini diperlukan ketika selama uji coba penelitian menghadapi kegagalan atau hasil yang tidak sesuai dengan harapan. Untuk mengatasi masalah tersebut, perbaikan pada *hardware* dan program menjadi langkah yang diperlukan.

Setelah perbaikan selesai dilakukan, dilanjutkan dengan proses uji coba ulang untuk memverifikasi perbaikan yang telah dilakukan.

3.7 Pengolahan Data

Setelah alat dapat bekerja dengan baik serta proses pengujian alat telah dilakukan untuk menjalankan implementasi dari sistem yang sudah dirancang sesuai dengan kebutuhan, maka tahapan selanjutnya adalah pengolahan data. Data yang diambil berupa jumlah orang yang terdeteksi dalam satu frame, pengontrolan *fan* AC berdasarkan jumlah orang yang terdeteksi, dan pengiriman data dari *IR Transmitter* ke AC dengan beberapa parameter jarak dan sudut.

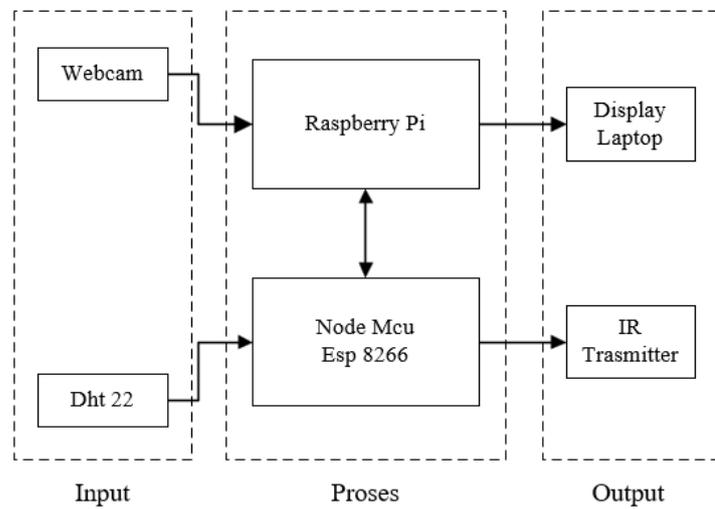
3.8 Pembuatan Laporan

Pembuatan laporan merupakan proses terakhir dari penelitian yang bertujuan untuk menampilkan hasil keseluruhan dengan detail proyek yang telah dibuat. Laporan ini meliputi latar belakang, rumusan masalah, batasan masalah, landasan teori, metodologi penelitian, analisis data, serta kesimpulan dan rekomendasi. Selain itu, beberapa lampiran terkait dengan pembuatan penelitian ini juga disertakan untuk mendukung penjelasan yang lebih lengkap.

BAB IV PEMBAHASAN

4.1 Diagram Blok

Diagram blok Pengontrolan *Air Conditioner* Berdasarkan Jumlah Orang dalam Ruang Berbasis Pengolahan Citra ditunjukkan pada gambar 4.1 berikut :



Gambar 4. 1 Diagram Blok Hardware

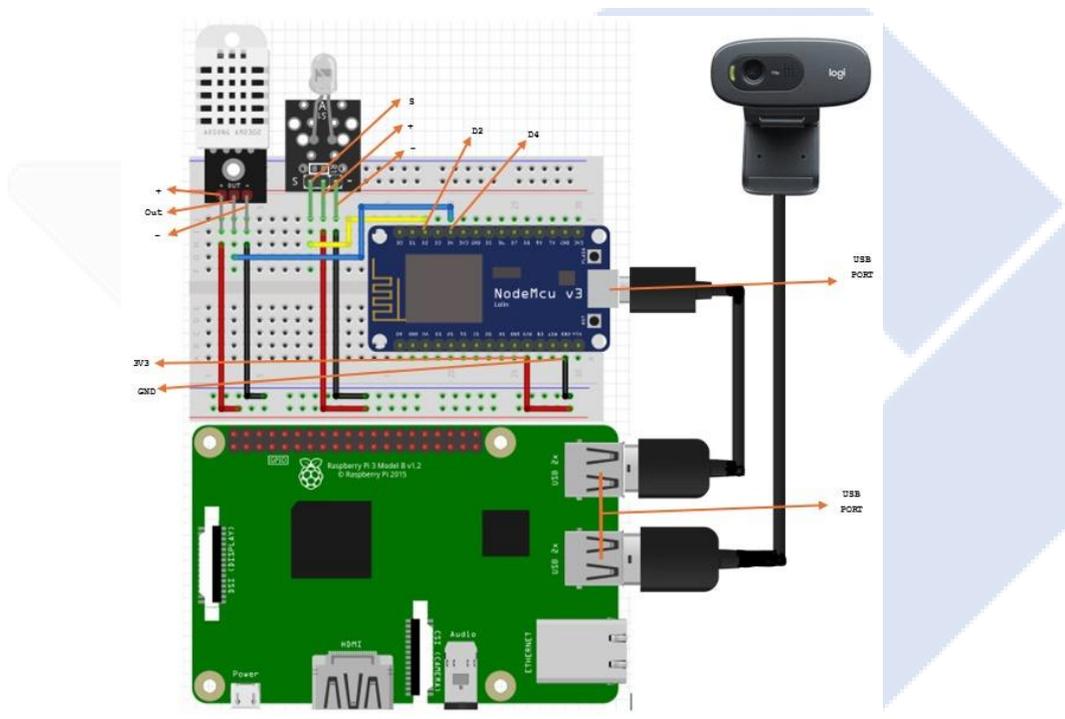
Berdasarkan informasi dari gambar diagram blok diatas, dapat dilihat bahwa kamera *webcam* sebagai input digunakan untuk mendeteksi objek (manusia) yang masuk ke dalam ruangan. Sensor DHT 22 digunakan untuk mengetahui suhu ruangan secara *realtime* yang nantinya akan di proses oleh *NodeMCU* dan dikirimkan ke bagian output yang berupa *IR Transmitter*. Setelah objek yang terdeteksi oleh kamera *webcam* di proses oleh *Raspberry Pi* menggunakan salah satu model deteksi YOLOV5 objek *deep learning* yang dikembangkan untuk mendeteksi objek secara *realtime*. Setelah data yang dihasilkan diproses oleh *NodeMCU ESP8266* maka akan ada sebuah perintah ke *IR Transmitter* untuk dikirimkan sinyal data ke AC, yang mengakibatkan *fan AC* akan berubah secara

otomatis tergantung jumlah objek yang terdeteksi dengan suhu ruangan dijaga dalam suhu referensi 20 - 25°C.

4.2 *Hardware* Sistem Kontrol

4.2.1 *Scematic Hardware* Sistem Kontrol

Scematic hardware merupakan proses yang dilakukan dalam membuat desain sistem kontrol. Perencanaan desain ini meliputi bagaimana penghubungan antar komponen yang digunakan. Pembuatan desain *scematic* ini menggunakan aplikasi *Fritzing*. Secara umum dapat dilihat pada Gambar 4.2 berikut.



Gambar 4. 2 *Scematic* Sistem Kontrol

Perancangan *scematic* ini berfungsi untuk dapat memudahkan dalam memahami dan menganalisis bagaimana sebuah rangkaian bekerja. Dengan menghubungkan *input*, *output*, ataupun *ground* yang dibutuhkan ke antar komponen lainnya. Hal ini dapat dijelaskan bahwa, pin 3.3V yang merupakan *power* pada *NodeMCU* disambungkan ke pin VCC sensor DHT 22 dan *IR Transmitter*. Penghubungan ini dilakukan dengan cara paralel. Kemudian pin GND (*ground*) pada *NodeMCU* disambungkan ke GND (*ground*) sensor suhu DHT 22

dan *IR Transmitter*. Selanjutnya, untuk pin output dari sensor DHT 22 dihubungkan ke pin D4, GPIO 02 dari *NodeMCU*. Untuk pin sinyal *IR Transmitter* disambungkan ke pin D2, GPIO 04 dari *NodeMCU*. Kemudian untuk USB Port 1 pada *Raspberry Pi* dihubungkan ke kamera *webcam* dan USB Port 2 pada *NodeMCU* sebagai komunikasi serial antar controller.

4.2.2 Pemilihan Komponen Sistem Kontrol

Pemilihan komponen meliputi pemilihan tipe dari komponen yang akan digunakan. Dalam pemilihan komponen ini penulis mempertimbangkan beberapa kriteria. Hal ini dilakukan karena mengingat fungsi dari tipe komponen yang harus sesuai dengan tujuan dari pembuatan alat ini. Dengan mempertimbangkan hal-hal berikut, komponen yang akan digunakan untuk membuat alat memiliki beberapa kriteria seperti yang ditunjukkan dalam Tabel 4.1.

Tabel 4. 1 Tabel Kriteria Komponen Yang Digunakan

No	Komponen	Alasan
1	<i>Raspberry Pi</i>	Adanya pin GPIO yang dapat memudahkan koneksi antar komponen lainnya
2	<i>Webcam</i>	Memiliki kualitas gambar yang baik, harga terjangkau, dan <i>plug</i> dan <i>play</i> melalui port USB <i>Raspberry Pi</i>
3	KY-005 <i>IR Transmitter Module</i>	<i>Plug</i> dan <i>play Raspberry Pi</i>
4	DHT 22	<i>Plug</i> dan <i>play Raspberry Pi</i> , memiliki rentang suhu dan akurasi suhu yang akurat dari DHT 11
5	Digital <i>IR Receiver Module</i>	Memudahkan pengambilan data remot AC menggunakan <i>IR Receiver</i>

No	Komponen	Alasan
6	NodeMCU v3	Pilihan kontroler terbaik untuk pengambilan data <i>remote</i> AC menggunakan <i>IR Receiver</i>

4.3 Komponen Sistem Kontrol

4.3.1 Modul DHT 22

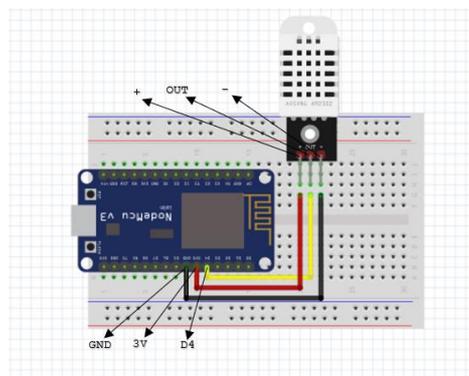
Modul DHT 22 digunakan sebagai pendeteksi suhu dan kelembapan. Data suhu dan kelembapan akan dikirimkan ke *NodeMCU* menggunakan pin D4 untuk digunakan sebagai data suhu secara *realtime* untuk mengukur suhu ruangan.

Pin modul DHT 22 yang dispesifikasikan menggunakan *NodeMCU* dapat dilihat pada tabel sebagai berikut :

Tabel 4. 2 Spesifikasi Pin Modul DHT 22

No	Pin DHT 22	Pin NodeMCU
1	<i>Out</i>	D4
2	VCC	3.3V
3	GND	GND

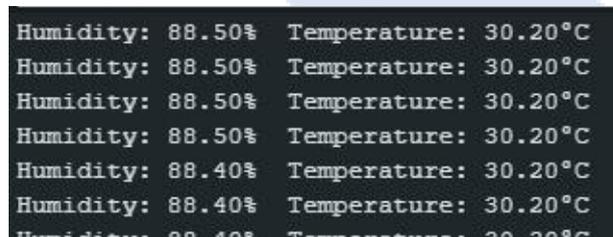
Berikut merupakan gambar *software* sambungan pin DHT 22 dengan *NodeMCU*, dapat dilihat pada Gambar 4.3.



Gambar 4. 3 *Software* Rangkaian DHT 22

Pengujian dilakukan pada Sensor DHT 22, seperti yang telah dijelaskan pada gambar diatas untuk memverifikasi kinerja komponen tersebut. Proses pengujian ini melibatkan pembuatan program NodeMCU yang dirancang dengan menggunakan *software IDE Arduino*, dengan tujuan untuk mengukur suhu dan kelembapan di dalam ruangan. Berikut program NodeMCU untuk pengujian Sensor DHT 22.

```
#include <Arduino.h>
#include "DHT.h"
#define DHTPIN D4
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);
float data_humidity;
float data_temperature;
void setup()
{ // Debug console
  Serial.begin(115200);
  dht.begin(); }
void loop()
{ delay(500);
  Blynk.run();
  data_humidity = dht.readHumidity();
  data_temperature = dht.readTemperature();
  if (isnan(data_humidity) || isnan(data_temperature))
  { Serial.println(F("Failed to read from DHT sensor!"));
    return; }
  Serial.print(F("Humidity: "));
  Serial.print(data_humidity);
  Serial.print(F("%   Temperature: "));
  Serial.print(data_temperature);
  Serial.println(F("°C "));
```



```
Humidity: 88.50% Temperature: 30.20°C
Humidity: 88.50% Temperature: 30.20°C
Humidity: 88.50% Temperature: 30.20°C
Humidity: 88.50% Temperature: 30.20°C
Humidity: 88.40% Temperature: 30.20°C
Humidity: 88.40% Temperature: 30.20°C
Humidity: 88.40% Temperature: 30.20°C
```

Gambar 4. 4 Pengujian Serial Monitor DHT22

Berdasarkan hasil pengujian menjelaskan bahwa sensor DHT22, yang digunakan untuk mengukur suhu dan kelembapan dalam proyek ini beroperasi sesuai dengan harapan dan rencana yang telah disusun sebelumnya.

4.3.2 Modul IR Transmitter

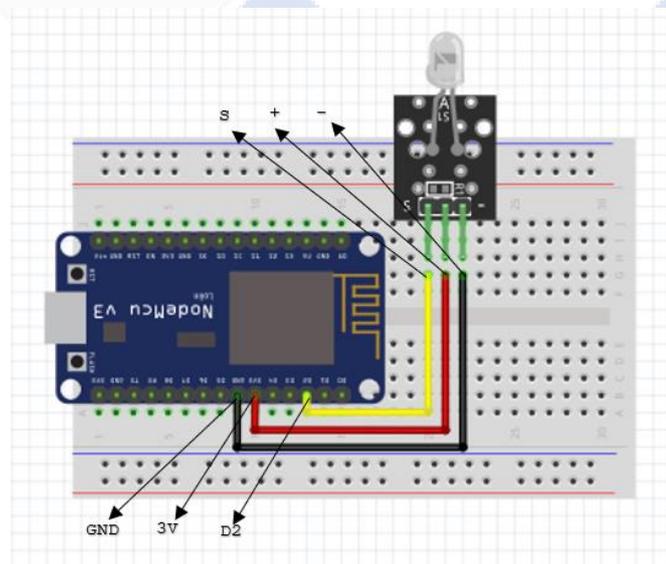
Modul *IR Transmitter* digunakan sebagai pemancar sinyal data ke AC. Kemudian data yang dihasilkan akan dikirimkan dari *NodeMCU* ke *IR Transmitter* menggunakan pin D2 yang nantinya akan dikirimkan ke AC.

Pin modul *IR Transmitter* yang dispesifikasikan menggunakan *NodeMCU* dapat dilihat pada tabel sebagai berikut :

Tabel 4. 3 Spesifikasi Pin Modul *IR Transmitter*

No	Pin <i>IR Transmitter</i>	Pin <i>NodeMCU</i>
1	Data	D2
2	VCC	3.3V
3	GND	GND

Berikut merupakan gambar *software* sambungan pin *IR Transmitter* dengan *NodeMCU*, dapat dilihat pada Gambar 4.5.



Gambar 4. 5 *Software* Rangkaian *IR Transmitter*

Untuk mengetahui apakah komponen ini berfungsi dengan baik, pengujian dilakukan pada Modul *IR Transmitter*. Ini dilakukan dengan membuat

program dengan menggunakan *software Arduino IDE* untuk mengirimkan data asli AC *panasonic* ke *IR Receiver*, seperti yang ditunjukkan pada gambar di atas. Program *NodeMCU* untuk pengujian modul *Ir Transmitter* dapat dilihat pada Lampiran 2.

Implementasi dalam proses pengiriman data ke AC dengan parameter jarak berbeda-beda yang dapat mempengaruhi sistem dalam memancarkan sinyal data ke AC dan juga bertujuan untuk mengetahui keoptimalan sistem dalam bekerja.

Tabel 4. 4 Pengiriman Data ke AC Dengan Jarak

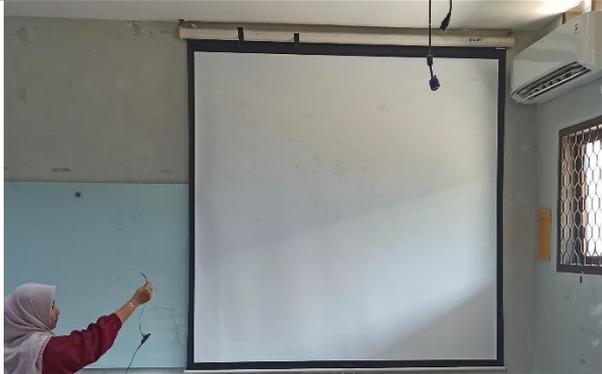
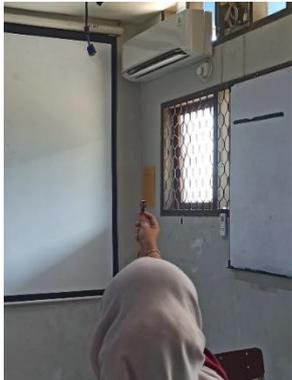
No	Jarak	Gambar Pengiriman Data ke AC Dengan Jarak	Kesimpulan
1.	1 Meter		Pemancaran sinyal data ke AC berhasil dikirim.
2.	2 Meter		Pemancaran sinyal data ke AC berhasil dikirim.

No	Jarak	Gambar Pengiriman Data ke AC Dengan Jarak	Kesimpulan
3.	3 Meter		Pemancaran sinyal data ke AC berhasil dikirim.
4.	4 Meter		Pemancaran sinyal data ke AC berhasil dikirim.
5.	5 Meter		Pemancaran sinyal data ke AC berhasil dikirim. Akan tetapi, dalam pengiriman sinyal berjalan kurang optimal.
6.	6 Meter		Pemancaran sinyal data ke AC tidak berhasil dikirim.

Dalam proses pengiriman sinyal data ke AC, bukan hanya jarak *IR Transmitter* yang mempengaruhi pemrosesan. Akan tetapi, posisi *IR Transmitter* juga menjadi salah satu faktor yang mempengaruhi hal tersebut. Berdasarkan dari

pengujian yang telah dilakukan, proses pengiriman data ke AC menggunakan *IR Transmitter* dengan beberapa jarak yang telah dilakukan dapat diambil kesimpulan bahwa modul dapat memancarkan sinyal data ke AC dengan maksimal jarak 6 meter.

Tabel 4. 5 Pengiriman Data ke AC Dengan Sudut

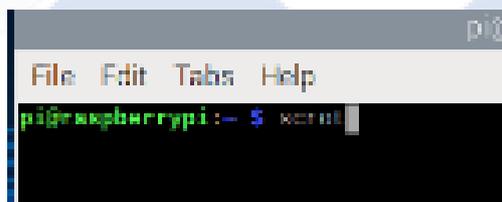
No	Sudut	Gambar Pengiriman Data ke AC Dengan Sudut	Kesimpulan
1.	30°		Pemancaran sinyal data ke AC berhasil dikirim.
2.	45°		Pemancaran sinyal data ke AC berhasil dikirim.
3.	60°		Pemancaran sinyal data ke AC berhasil dikirim.

No	Sudut	Gambar Pengiriman Data ke AC Dengan Sudut	Kesimpulan
4.	90°		Pemancaran sinyal data ke AC berhasil dikirim.

Dalam proses pengiriman sinyal data ke AC, posisi *IR Transmitter* harus sejajar dengan AC untuk meminimalisir kegagalan dalam proses pengiriman.

4.3.3 *Raspberry Pi*

Raspberry Pi digunakan untuk pengolahan *image processing* menggunakan salah satu model deteksi YOLOV5 untuk mendeteksi objek (manusia) berbasis *deep learning* yang kemudian datanya dikirimkan ke NodeMCU untuk diproses sekaligus menampilkan suhu pada layar monitor.



Gambar 4. 6 Tampilan Uji Coba *Raspberry Pi*

Dari gambar diatas dilakukan pengujian pada *Raspberry Pi* untuk melakukan ujicoba apakah komponen ini bekerja dengan optimal. Pada pengujian ini *Raspberry Pi* dicoba untuk dinyalakan apakah bisa untuk menampilkan gambar.

4.3.4 **NodeMCU ESP8266**

NodeMCU ESP8266 digunakan untuk pengolahan data *image processing* dari *Raspberry Pi* yang nantinya di proses untuk mengontrol AC menggunakan *IR*

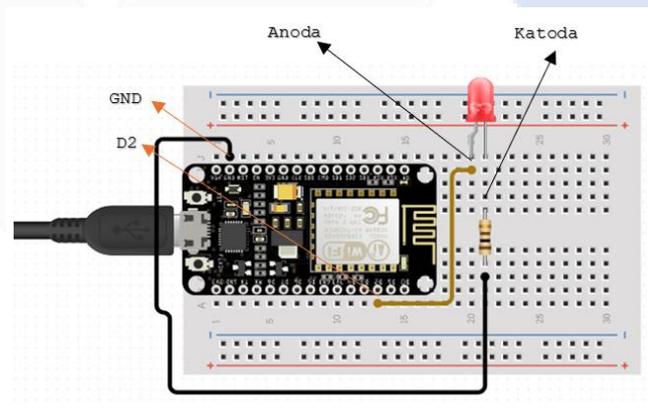
Transmitter sesuai jumlah manusia yang terdeteksi dan sesuai dengan suhu referensi 20 - 25°C.

Pin Led yang dispesifikasikan untuk menggunakan NodeMCU dapat dilihat pada tabel sebagai berikut :

Tabel 4. 6 Spesifikasi Pin Modul NodeMCU

No	Pin Led	Pin NodeMCU
1	GND	GND
2	VCC	D2

Berikut merupakan gambar *software* sambungan pin LED dengan NodeMCU, dapat dilihat pada Gambar 4.7.



Gambar 4. 7 Software Rangkaian Pengujian NodeMCU ESP8266

Pengujian yang dilakukan pada NodeMCU ESP8266, seperti yang ditunjukkan di atas untuk memastikan bahwa komponen ini beroperasi dengan baik. Untuk pengujian ini, software IDE Arduino digunakan untuk membuat program yang dapat menyalakan lampu LED. Berikut program *NodeMCU ESP8266* untuk pengujian *NodeMCU ESP8266*.

```
// Inisialisasi Port
int led = D2;
// initialize digital pin LED_BUILTIN as an output.
void setup() {
  pinMode (led, OUTPUT);
}
```

```
// the loop function runs over and over again forever
void loop (){
  digitalWrite (led, HIGH);
  delay (1000);
  digitalWrite (led, LOW);
  delay (1000);}

```

4.4 Komponen Sistem Pengambilan Data

4.4.1 Modul *IR Receiver*

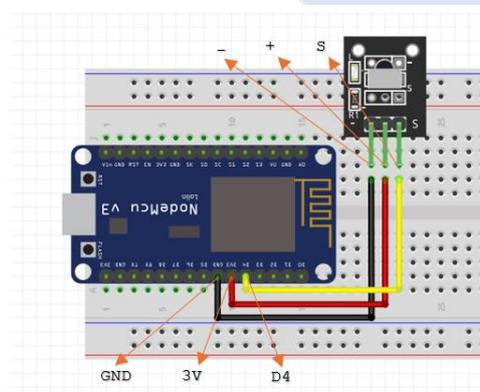
Modul *IR Receiver* digunakan untuk mengambil raw data dari *remote* AC. Kemudian raw data yang dihasilkan digunakan untuk mengontrol AC menggunakan *IR Transmitter*.

Pin modul *IR Receiver* yang dispesifikasikan menggunakan NodeMCU dapat dilihat pada tabel sebagai berikut :

Tabel 4. 7 Spesifikasi Pin Modul IR Receiver

No	Pin <i>IR Receiver</i>	Pin NodeMCU
1	Data	D4
2	VCC	3.3V
3	GND	GND

Berikut merupakan gambar *software* sambungan pin DHT 22 dengan NodeMCU, dapat dilihat pada Gambar 4.8.



Gambar 4. 8 *Software* Rangkaian *IR Receiver*

Pengujian dilakukan pada Modul *IR Receiver* untuk mengetahui apakah komponen ini berfungsi dengan baik, seperti yang ditunjukkan di atas. Program dibuat untuk menerima data asli dari *remote AC Panasonic*, yang kemudian akan muncul di serial monitor menggunakan *software* Arduino IDE. Program NodeMCU untuk pengujian modul *IR Receiver* bisa dilihat pada Lampiran 3.

```
Protocol : PANASONIC_AC
Code      : 0x0220E0040000000040220E0040001308C4F000004400000050000
Msg Desc.: Model: 4 (JXX), Power: On, Mode: 0 (Auto), Temp: 24C,
uint16_t rawData[439] = {3536, 1746, 472, 400, 424, 3240, 422,
uint8_t state[27] = {0x02, 0x20, 0x20, 0x04, 0x00, 0x00, 0x00, 0x00
```

Gambar 4. 9 Software Rangkaian *IR Receiver*

Pada gambar 4.9 menunjukkan hasil pengujian pada *IR Receiver* berdasarkan program yang telah dibuat. Dapat dilihat AC yang digunakan, status AC, mode *fan* AC, dan temperatur.

4.5 Pengujian Algoritma Deteksi Objek

4.5.1 YOLOV5

YOLOV5 digunakan sebagai algoritma deteksi objek dalam gambar atau video secara *real time* yang nantinya diproses oleh *Raspberry Pi*. Program untuk pengujian modul YOLOV5 dapat dilihat pada Lampiran 4.

Implementasi sistem deteksi objek (manusia) ini diuji coba secara *real-time* dengan beberapa parameter jarak, intensitas cahaya, hingga penggunaan aksesoris yang memungkinkan dapat mempengaruhi sistem dalam mendeteksi objek (manusia). Adapun tujuan dari uji coba ini adalah untuk mengetahui seberapa optimal sistem dapat bekerja.



Gambar 4. 10 Hasil Pengujian YOLOV5 dalam Jarak Dekat

Pada pengujian yang telah dilakukan di jarak ± 2 meter dengan kondisi cahaya normal didapatkan hasil objek (manusia) yang terdeteksi penuh sesuai dengan jumlah orang yang benar berada dalam cangkupan *frame camera*. Pada hasil tersebut, terdeteksi objek (manusia) sebanyak 7 orang. Dengan bermacam-macam posisi dan aksesoris yang digunakan, seperti menggunakan helm, masker, bahkan posisi objek dengan berbagai gaya. Dari hal tersebut, dapat diketahui bahwa aksesoris yang digunakan ataupun posisi dari objek, tidak mempengaruhi hasil pendeteksian. Selagi objek yang masuk ke dalam *frame camera* terlihat jelas.



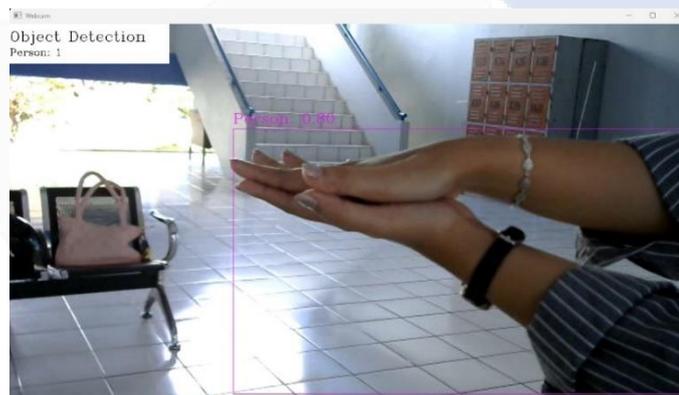
Gambar 4. 11 Hasil Pengujian YOLOV5 dengan Jarak Berbeda

Berdasarkan dari pengujian pada jarak yang berbeda, *YOLOV5* tetap terdeteksi objek (manusia) sebanyak 2 orang walaupun objek tersebut terlihat kurang begitu jelas dan intensitas cahaya yang kurang. Dengan kata lain, algoritma *YOLOV5* adalah metode deteksi objek yang efisien dan akurat, sehingga memungkinkan deteksi objek secara real time. Maka dari itu, beberapa tantangan dalam deteksi

objek, seperti jarak dan cahaya tidak mempengaruhi validasi hasil objek yang terdeteksi selama objek masuk ke dalam *frame camera*.



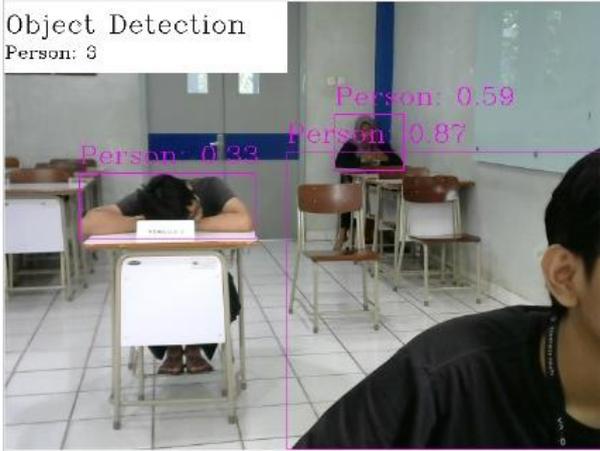
Gambar 4. 12 Hasil Pengujian 1 Grid YOLOV5

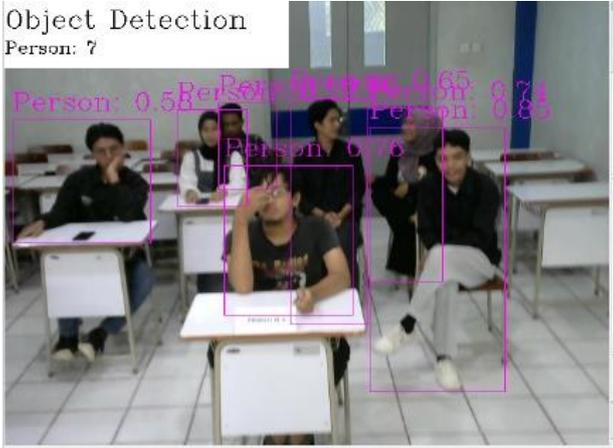


Gambar 4. 13 Hasil Pengujian 2 Grid YOLOV5

Berdasarkan gambar hasil pengujian algoritma YOLOV5 tersebut dapat diketahui bahwa YOLOV5 dapat digunakan dalam mendeteksi objek (manusia), walaupun hanya sebagian tubuh yang hanya masuk ke dalam *frame camera*. Hal ini dikarenakan, cara kerja YOLOV5 yang membagi objek menjadi beberapa bagian sehingga mendapatkan ciri-ciri yang dimiliki oleh objek yang dicari.

Tabel 4. 8 Hasil Pengujian Deteksi Objek

No	Jumlah Orang	Gambar Pengujian Deteksi Objek	Kesimpulan
1.	0		Tidak ada objek (manusia) yang terdeteksi
2.	3		3 dengan 3 terdeteksi dengan benar
3.	4		4 dengan 4 terdeteksi dengan benar

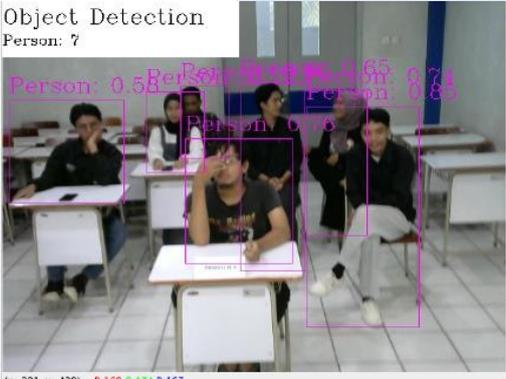
No	Jumlah Orang	Gambar Pengujian Deteksi Objek	Kesimpulan
4.	7	<div style="display: flex; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Object Detection Person: 7</div>  </div>	7 dengan 7 terdeteksi dengan benar

4.6 Pengujian Kontrol *Fan AC* Berdasarkan Jumlah Orang

Pengujian ini merupakan tahap lanjutan dari pengujian yang telah dilakukan pada YOLO V5 mengenai deteksi objek, penulis melakukan pengujian pada sistem kontrol *Fan AC* yang akan berubah sesuai dengan sistem yang telah ditetapkan berdasarkan jumlah orang yang terdeteksi oleh *camera webcam* dalam ruangan.

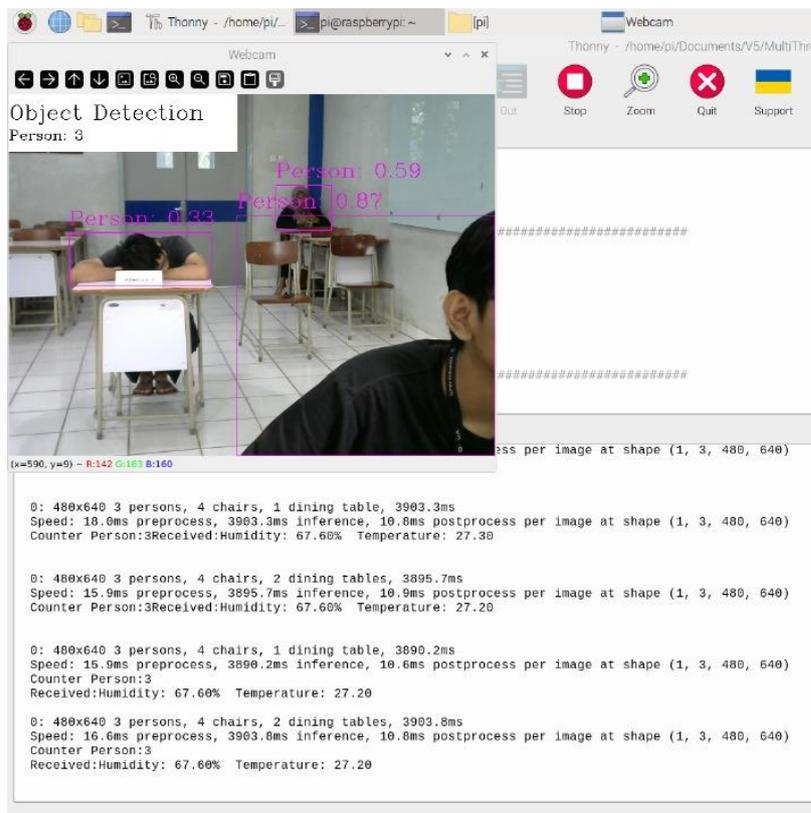
Tabel 4. 9 Hasil Pengujian Kontrol *Fan AC*

No	Jumlah Orang	Gambar Pengujian Kontrol <i>Fan AC</i>	Kesimpulan
1.	0	<div style="display: flex; align-items: flex-start;"> <div style="border: 1px solid black; padding: 5px; margin-right: 10px;">Object Detection Person: 0</div>  </div> <pre style="font-family: monospace; font-size: 0.8em; margin-top: 10px;"> Timestamp : 001883.119 Library : v2.8.6 Protocol : PANASONIC_AC Code : 0x0220E004000000060220E004003828803500000EE00000089000092 (216 Bits) Msg Desc.: Model: 6 (RRR), Power: Off, Mode: 3 (Cool), Temp: 20C, Fan: 0 (Quiet),</pre>	Terdeteksi tidak ada orang dalam ruangan, dengan benar status AC yaitu OFF, <i>fan AC</i> dengan status 0 (<i>low</i>), dan suhu AC yang terjaga dalam suhu referensi 20°C.

No	Jumlah Orang	Gambar Pengujian Kontrol <i>Fan AC</i>	Kesimpulan
2.	3	<p>Object Detection Person: 3</p>  <pre>Timestamp : 002100.830 Library : v2.8.6 Protocol : PANASONIC_AC Code : 0x0220E004000000060220E004003928803500000EE000008900093 (216 Bits) Msg Desc.: Model: 6 (RKR), Power: On, Mode: 3 (Cool), Temp: 20C, Fan: 0 (Quiet),</pre>	<p>Terdeteksi 3 orang dalam ruangan, dengan benar status AC yaitu ON, <i>fan AC</i> dengan status 0 (<i>low</i>), dan suhu AC yang terjaga dalam suhu referensi 20°C.</p>
3.	4	<p>Object Detection Person: 4</p>  <pre>Timestamp : 002200.719 Library : v2.8.6 Protocol : PANASONIC_AC Code : 0x0220E004000000060220E004003928805500000EE00000890000B3 (216 Bits) Msg Desc.: Model: 6 (RKR), Power: On, Mode: 3 (Cool), Temp: 20C, Fan: 2 (Medium),</pre>	<p>Terdeteksi 4 orang dalam ruangan, dengan benar status AC yaitu ON, <i>fan AC</i> dengan status 2 (<i>medium</i>), dan suhu AC yang terjaga dalam suhu referensi 20°C.</p>
4.	7	<p>Object Detection Person: 7</p>  <pre>Timestamp : 000394.419 Library : v2.8.6 Protocol : PANASONIC_AC Code : 0x0220E004000000060220E004003928807500000EE00000890000D3 (216 Bits) Msg Desc.: Model: 6 (RKR), Power: On, Mode: 3 (Cool), Temp: 20C, Fan: 4 (Maximum),</pre>	<p>Terdeteksi 7 orang dalam ruangan, dengan benar status AC yaitu ON, <i>fan AC</i> dengan status 4 (<i>maximum</i>), dan suhu AC yang terjaga dalam suhu referensi 20°C.</p>

4.7 Tampilan Monitor *Display* Pada *Raspberry Pi*

Dalam pengerjaan proyek akhir ini menggunakan aplikasi *Thony Python* untuk menjalankan *image processing*. *Thony Python* digunakan agar lebih mempermudah dalam memonitor dengan menampilkan jumlah objek (manusia) yang terdeteksi, suhu bahkan kelembapan pada ruangan. Tampilan pada aplikasi *Thony Python* ditunjukkan pada Gambar 4.14.



Gambar 4. 14 Tampilan Layar Monitor

Dari tampilan aplikasi *Thony Python* pada layar monitor tersebut, dapat dilihat bahwa pada aplikasi tersebut dapat menampilkan jumlah objek (manusia) yang terdeteksi. Bahkan dapat menampilkan kelembapan dan temperatur di dalam ruangan berdasarkan sensor DHT 22.

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Ada beberapa kesimpulan yang dapat dibuat berdasarkan hasil pengujian dan analisis yang telah dilakukan dalam penelitian "Pengontrolan *Air Conditioner* Berdasarkan Jumlah Orang Dalam Ruangan Berbasis Pengolahan Citra" :

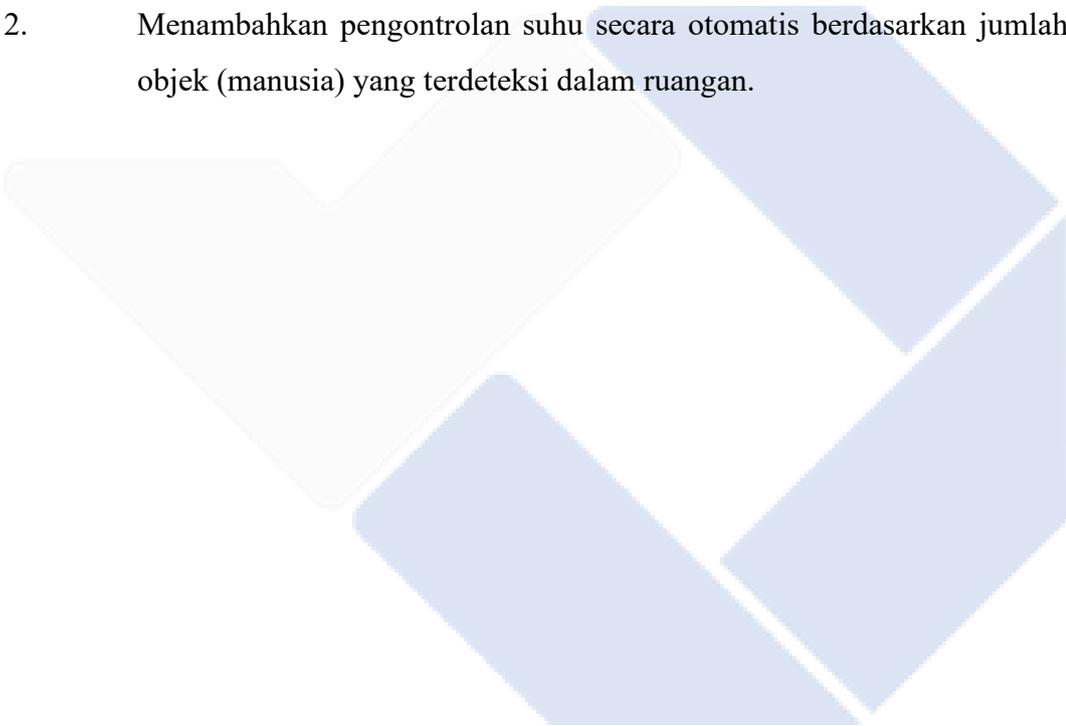
1. Pengaturan *fan AC* secara otomatis yang telah dibuat sudah efisien dengan menggunakan *Raspberry Pi* sebagai kontroler.
2. Sistem yang dapat membaca jumlah orang dalam ruangan menggunakan *image processing* dengan algoritma YOLO V5.
3. Hasil uji pengontrolan *fan AC* sudah berfungsi dengan baik. *Speed fan AC* dapat berubah secara otomatis berdasarkan jumlah orang dalam ruangan dan dapat menjaga suhu referensi ruangan sekitar 20-25°C.
4. Pengontrolan *fan AC* berdasarkan data dari *image processing* dilakukan dengan mengirimkan data *remote AC* sesuai dengan objek yang telah terdeteksi yang kemudian di proses sesuai dengan beberapa opsi pengaturan yang telah ditentukan untuk mengatur *speed AC*.
5. Sistem dapat mendeteksi objek (manusia) secara *realtime*.
6. Sistem dapat menghitung jumlah objek (manusia) yang terdeteksi dalam ruangan dengan menampilkan hasil data pada layar monitor.
7. Faktor yang mempengaruhi sistem bekerja kurang optimal dalam perhitungan jumlah objek yang terdeteksi, yaitu faktor antar objek (manusia) yang terlalu berdekatan dan pencahayaan dalam ruangan yang kurang mendukung.
8. Sistem dapat mengatur *fan AC* secara otomatis berdasarkan jumlah orang yang terdeteksi dalam ruangan sesuai dengan beberapa opsi pengaturan yang telah ditentukan.
9. Menjalankan sistem *image processing* menggunakan *Raspberry Pi* Model B membuat pendeteksian objek menjadi *delay* dikarenakan

Raspberry tidak terlalu kuat untuk menjalankan deteksi objek dengan lancar.

5.2 Saran

Dari hasil proyek akhir yang telah dilakukan, ada beberapa saran yang ingin disampaikan untuk pengembangan yang lebih lanjut, antara lain :

1. Menggunakan *Raspberry Pi* versi terbaru agar dalam proses pendeteksian objek tidak terjadi *delay* yang terlalu lama.
2. Menambahkan pengontrolan suhu secara otomatis berdasarkan jumlah objek (manusia) yang terdeteksi dalam ruangan.



DAFTAR PUSTAKA

- [1] R. Fetra dan Hambali, “Sistem Otomasi Penyalaan Lampu dan AC (Air Conditioner) pada Ruang Dosen Berbasis Arduino UNO,” 2020, [Daring]. Tersedia pada: <http://ejournal.unp.ac.id/index.php/jtev/index>
- [2] A. Pratama dan Elfizon, “Sistem Pengontrolan Air Conditioner Berbasis Arduino,” 2021.
- [3] I. Heryanto, G. Suweken, dan K. Aryanto, “Modul Portable Rumah Cerdas Untuk Penghematan Penggunaan Energi Listrik Menggunakan Arduino Uno dan Sensor Ping Sebagai Pengontrol,” 2020.
- [4] F. Irawan dan F. Merdy, “Pengontrolan Suhu dan Perhitungan Jumlah Orang yang Masuk Ruangan Rapat Berbasis IoT,” 2023.
- [5] I. Jati dan M. Rivai, “Implementasi Thermal Camera Pada Pengaturan Pendingin Ruangan,” 2019.
- [6] Siaulhak, S. Kasma, dan Suparman, “Sistem Pengiriman File Menggunakan Steganografi Pengolahan Citra Digital Berbasis Matriks Laboratory,” 2023.
- [7] Widyawati, A. Fathoni, Sutanto, dan Renaldi, “Identifikasi Ukuran Pakaian Berbasis Image Processing,” vol. 5, no. 1, 2021.
- [8] A. Pradana, “Sistem Kontrol dan Monitoring Air Conditioner Berbasis Internet of Things (IOT),” 2022.
- [9] H. Nasirrudin dan B. Sembodo, “Control System Automatic for Light and Air Conditioner at Living Room using Arduino Uno Controller,” vol. 04, no. 01, hlm. 2022, 2022.
- [10] R. Ridho Prabowo dan R. Taufiq Subagio, “Sistem Monitoring dan Pemberian Pakan Otomatis Pada Budidaya Ikan Menggunakan Wemos dengan Konsep Internet of Things (IoT),” 2020.
- [11] R. Prabowo, Kusnadi, dan R. Subagio, “S(IoT),” 2020.
- [12] R. Friadi, “Sistem Kontrol Intensitas Cahaya, Suhu dan Kelembaban Udara Pada Greenhouse Berbasis Raspberry PI,” 2019.

- [13] H. I. Islam *dkk.*, “Sistem Kendali Suhu Dan Pemantauan Kelembaban Udara Ruang Berbasis Arduino Uno Dengan Menggunakan Sensor Dht22 Dan Passive Infrared (PIR),” Universitas Negeri Jakarta, 2016, hlm. SNF2016-CIP-119-SNF2016-CIP-124. doi: 10.21009/0305020123.
- [14] M. Julyasri, “Pengaruh Ruang Ber-AC (Air Conditioner) Terhadap Gejala Sick Building Syndrome (SBS) di Gedung Pepadun Pemerintah Kota Lalampung dan Bagian Umum Gedung Rektorat Universitas Islam Negeri Raden Intan Lampung,” 2023.
- [15] M. Rozaq, B. Sukoco, dan D. Nugroho, “Analisa Pengaruh Setting Suhu Air Conditioner Terhadap Konsumsi Energi Listrik Pada Air Conditioner Kapasitas 5 PK Type PSF 5001,” 2019.
- [16] M. Dermawan, D. Witarsyah, dan H. Fakhruroja, “Penerapan Image Processing untuk Mengetahui Tingkat Kematangan Kopi Menggunakan Algoritma K-Nearest Neighbor (KNN) pada Perkebunan Kopi Malabar Bandung,” 2023.
- [17] S. Maslikah, R. Alfita, dan A. F. Ibadillah, “Sistem Deteksi Kantuk Pada Pengendara Roda Empat Menggunakan Eye Blink Detection,” 2022.
- [18] M. Arifin, K. Joni, dan M. Ulum, “Penentuan Kualitas Warna Batu Blue Sapphire Dengan Image Processing Menggunakan Metode RGB To HSV,” 2019.
- [19] M. A. Hidayat, N. Husni, dan F. Damsi, “Pendeteksi Banjir dengan Image Processing Berbasis Convolutional Neural Network (CNN) pada Kamera Pengawas,” vol. 2, hlm. 10–18, 2022.
- [20] N. Utomo, B. Budiana, A. Jefiza, dan F. Nakul, “Sistem Pemetaan Parkir Menggunakan Teknik Image Processing,” 2023.
- [21] M. Kurniawan, U. Sunarya, dan R. Tulloh, “Internet of Things : Sistem Keamanan Rumah berbasis Raspberry Pi dan Telegram Messenger,” *ELKOMIKA: Jurnal Teknik Energi Elektrik, Teknik Telekomunikasi, & Teknik Elektronika*, vol. 6, no. 1, hlm. 1, Apr 2018, doi: 10.26760/elkomika.v6i1.1.

- [22] A. Juliansyah, Ramlah, dan D. Nadiani, “Sistem Pendeteksi Gerak Menggunakan Sensor PIR dan Raspberry Pi (Motion Detection System Using PIR Sensors and Raspberry Pi),” vol. 2, no. 4, hlm. 199–205, 2021.
- [23] Desmira, A. Fatoni, dan F. Gumilang, “Perancangan Smart Home Dengan Raspberry Berbasis Wireless Menggunakan Mikrokontroler dan Fuzzy Logic (Pada Mess PLTU Labuan),” 2016.
- [24] I. Yaqin, R. Alfita, dan K. A. Wibisono, “Alat Identifikasi Jenis kayu Berbasis Image Processing dengan Metode K-Nearest Neighbor (KNN),” *Seminar Nasional Fortei Regional*, vol. 7, 2022.
- [25] Heriyanto, S. Nugroho, dan B. Yuwono, *Implementasi Deteksi Warna pada Public Monitoring System dengan Rassyberry Pi*. 2020.
- [26] Y. Apriliansah, E. Kurniawan, dan R. I. Vidyastari, “Rancang Bangun Alat Deteksi Fertilitas Telur Unggas Berbasis Image Processing,” *Digital Transformation Technology (Digitech) | e*, vol. 3, no. 1, 2023, doi: 10.47709/digitech.v3i1.2624.
- [27] D. I. Mulyana dan M. A. Rofik, “Implementasi Deteksi Real Time Klasifikasi Jenis Kendaraan Di Indonesia Menggunakan Metode YOLOV5,” 2022.
- [28] D. Permana dan J. Sutopo, “Aplikasi Pengenalan Abjad Sistem Isyarat Bahasa Indonesia (SIBI) dengan Algoritma YOLOv5,” vol. 11, no. 2, 2023.
- [29] M. Ashar dan D. Suarna, “Implementasi Algoritma YOLOv5 dalam Mendeteksi Penggunaan Masker Pada Kantor Biro Umum Gubernur Sulawesi Barat,” *Media Online*, vol. 3, no. 3, hlm. 298–302, 2022, [Daring]. Tersedia pada: <https://djournals.com/klik>
- [30] W. Swastika, A. W. Nur, dan O. H. Kelana, “Monitoring Ruangan Untuk Deteksi Manusia Berbasis CNN Dengan Fitur Push Notification,” *Teknika*, vol. 8, no. 2, hlm. 92–96, Okt 2019, doi: 10.34148/teknika.v8i2.166.
- [31] Khairunnas, E. Yuniarno, dan A. Zaini, “Pembuatan Modul Deteksi Objek Manusia Menggunakan Metode YOLO untuk Mobile Robot,” 2021.
- [32] A. N. Sugandi dan B. Hartono, “Implementasi Pengolahan Citra pada Quadcopter untuk Deteksi Manusia Menggunakan Algoritma YOLO,” 2022.

- [33] M. Yusup Efendi, R. Wulanningrum, A. Bagus Setiawan, dan U. Nusantara PGRI Kediri, “Rancang Bangun Sistem Deteksi Manusia dengan YOLO pada video CCTV,” Online, 2024.



Lampiran 1 Daftar Riwayat Hidup

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Eri Nur Mushthafa
Tempat, Tanggal Lahir : Sungailiat, 9 November 2003
Alamat : Jl. Sam Ratulangi No.37
No. HP : 082237970935
Email : mlerinurmushthafa@gmail.com
Jenis Kelamin : Laki-Laki
Agama : Islam



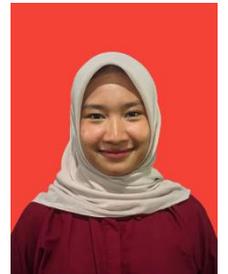
2. Riwayat Pendidikan

1. SD Negeri 1 Sungailiat Lulus 2015
2. SMP Negeri 1 Sungailiat Lulus 2018
3. SMA Negeri 1 Sungailiat Lulus 2021
4. Politeknik Manufaktur Negeri Bangka Belitung 2021-Sekarang

DAFTAR RIWAYAT HIDUP

1. Data Pribadi

Nama Lengkap : Herda Fitri Permatasari Riva'i
Tempat, Tanggal Lahir : Sungailiat, 5 Desember 2003
Alamat : Jl. Kartini No. 7 Kp. Jawa
No. HP : 085788907623
Email : herdafitri0512@gmail.com
Jenis Kelamin : Perempuan
Agama : Islam



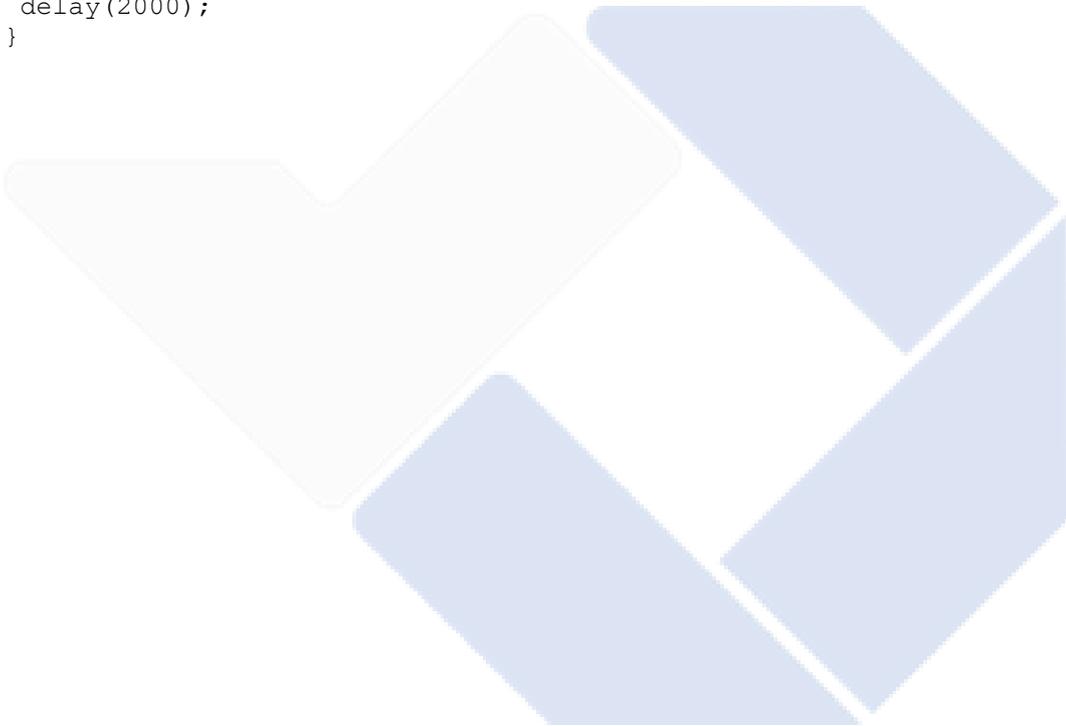
2. Riwayat Pendidikan

5. SD Negeri 4 Sungailiat	Lulus 2015
6. SMP Negeri 1 Sungailiat	Lulus 2018
7. SMAS Setia Budi Sungailiat	Lulus 2021
8. Politeknik Manufaktur Negeri Bangka Belitung	2021-Sekarang

Lampiran 2

```
#include <Arduino.h>
#include <IRremoteESP8266.h>
#include <IRsend.h>
const uint16_t kIrLed = 4; // ESP8266 GPIO pin to use.
Recommended: 4 (D2).
IRsend irsend(kIrLed); // Set the GPIO to be used to sending the
message.
// Example of data captured by IRrecvDumpV2.ino
uint16_t rawDataAcON[439] = {3538, 1722, 498, 398, 474, 1168,
494, 398, 474, 372, 498, 374, 496, 376, 496, 398, 472, 376,
496, 398, 472, 374, 496, 398, 472, 374,
496, 398, 472, 1168, 496, 376, 494,
398, 474, 374, 496, 376, 496, 376, 496, 374, 496, 374, 496,
1170, 494, 1166, 498, 1166, 498, 376, 494, 376, 496, 1168,
496,
398, 472, 400, 472, 376, 494, 398,
472, 374, 498, 374, 498, 374, 496, 376, 496, 398, 474, 376,
494, 376, 496, 376, 496, 398, 472, 376, 496, 376, 494, 376,
494, 376, 496, 374, 498, 374, 496, 376,
496, 376, 496, 374, 496, 376, 496, 374, 496, 374, 496, 398,
472, 376, 494, 400, 472, 374, 496, 374, 496, 1170, 494,
1170, 494, 398, 472, 374, 496, 374,
496, 374, 496, 398, 472, 9912, 3562, 1746, 472, 374, 496,
1166, 496, 376, 496, 376, 494, 376, 496, 400, 472, 376, 494,
376,
496, 374, 498, 398, 474, 376, 494, 398,
472, 374, 496, 1168, 496, 376, 494, 376, 496, 376, 496, 374,
496, 376, 496, 374, 496, 376, 496, 1168, 496, 1192, 472,
1170, 494, 400, 472, 376, 496, 1190,
472, 400, 472, 376, 494, 398, 472, 374, 496, 376, 496, 376,
496, 398, 472, 398, 472, 398, 472, 376, 496, 374, 496, 374,
496, 376, 494, 1192, 472, 398, 472,
398, 472, 376, 496, 376, 496, 376, 496, 376, 496, 374, 496,
376, 494, 376, 496, 376, 494, 376, 496, 1166, 496, 1192,
472,
374, 498, 374, 496, 374, 496, 376,
496, 376, 496, 376, 494, 398, 472, 400, 472, 400, 472, 1166,
498, 1168, 496, 1168, 496, 1192, 472, 1168, 494, 1170, 494,
1168, 496, 376, 496, 374, 496, 376,
494, 374, 496, 398, 472, 376, 494, 376, 494, 398, 472, 398,
472, 376, 496, 374, 496, 400, 472, 376, 494, 374, 496, 378,
494, 398, 472, 374, 496, 400, 472, 378,
494, 1192, 472, 1166, 496, 376, 496, 374, 496, 378, 492, 376,
494, 376, 496, 376, 494, 376, 496, 376, 496, 398, 472,
376, 496, 1168, 496, 1192, 472, 376, 496,
378, 494, 400, 472, 374, 496, 374, 496, 374, 498, 376, 494,
374, 496, 376, 496, 374, 496, 376, 496, 374, 496, 376,
494, 376, 496, 376, 496, 374, 496, 376,
494, 1168, 496, 374, 496, 1192, 472, 374, 496, 376, 496, 376,
496, 376, 494, 1168, 496, 398, 472, 400, 472, 400, 472,
```

```
        400, 472, 400, 472, 376, 494, 376,
496, 374, 496, 376, 494, 400, 472, 376, 496, 376, 494, 376,
494, 400, 470, 376, 496, 376, 494, 1168, 494, 376, 496, 376,
        494, 400, 472, 376, 496, 1168, 496,
1170, 494, 1168, 496}; // PANASONIC_AC
void setup() {
  irsend.begin();
  #if ESP8266
  Serial.begin(115200, SERIAL_8N1, SERIAL_TX_ONLY);
  #else // ESP8266
  Serial.begin(115200, SERIAL_8N1);
  #endif // ESP8266
  void loop() {
    Serial.println("SENDING RAW DATA TO TRANSMITTER");
    irsend.sendRaw(rawDataAcON, 439, 38); // Send a raw data capture
    at 38kHz.
    delay(2000);
  }
}
```



Lampiran 3

```
#include "BaseOTA.h"
#include <Arduino.h>
#include <assert.h>
#include <IRrecv.h>
#include <IRremoteESP8266.h>
#include <IRac.h>
#include <IRtext.h>
#include <IRutils.h>
#ifdef ARDUINO_ESP32C3_DEV
const uint16_t kRecvPin = 10; // 14 on a ESP32-C3 causes a boot
loop.
#else // ARDUINO_ESP32C3_DEV
const uint16_t kRecvPin = 14;
#endif // ARDUINO_ESP32C3_DEV
const uint32_t kBaudRate = 115200;
const uint16_t kCaptureBufferSize = 1024;
#ifdef DECODE_AC
// Some A/C units have gaps in their protocols of ~40ms. e.g.
Kelvinator
// A value this large may swallow repeats of some protocols
const uint8_t kTimeout = 50;
#else // DECODE_AC
// Suits most messages, while not swallowing many repeats.
const uint8_t kTimeout = 15;
#endif // DECODE_AC
const uint16_t kMinUnknownSize = 12;
const uint8_t kTolerancePercentage = kTolerance; // kTolerance is
normally 25%
#define LEGACY_TIMING_INFO false
IRrecv irrecv(kRecvPin, kCaptureBufferSize, kTimeout, true);
decode_results results; // Somewhere to store the results
void setup() {
  OTAwifi(); // start default wifi (previously saved on the ESP)
for OTA
#ifdef ESP8266
  Serial.begin(kBaudRate, SERIAL_8N1, SERIAL_TX_ONLY);
#elif ARDUINO_USB_CDC_ON_BOOT
  Serial.begin(kBaudRate);
#else // ESP8266
  Serial.begin(kBaudRate, SERIAL_8N1);
#endif // ESP8266
  while (!Serial) // Wait for the serial connection to be
established.
    delay(50);
  assert(irutils::lowLevelSanityCheck() == 0);
  Serial.printf("\n" D_STR_IRRECVDUMP_STARTUP "\n", kRecvPin);
  OTAinit(); // setup OTA handlers and show IP
#ifdef DECODE_HASH
  // Ignore messages with less than minimum on or off pulses.
  irrecv.setUnknownThreshold(kMinUnknownSize);
#endif // DECODE_HASH
  irrecv.setTolerance(kTolerancePercentage); // Override the
default tolerance.
  irrecv.enableIRIn(); // Start the receiver
```

```

}
// The repeating section of the code
void loop() {
  // Check if the IR code has been received.
  if (irrecv.decode(&results)) {
    // Display a crude timestamp.
    uint32_t now = millis();
    Serial.printf(D_STR_TIMESTAMP " : %06u.%03u\n", now / 1000,
now % 1000);
    // Check if we got an IR message that was too big for our
capture buffer.
    if (results.overflow)
      Serial.printf(D_WARN_BUFFERFULL "\n", kCaptureBufferSize);
    // Display the library version the message was captured with.
    Serial.println(D_STR_LIBRARY " : v"
_IRREMOTEESP8266_VERSION_STR "\n");
    // Display the tolerance percentage if it has been changed from
the default.
    if (kTolerancePercentage != kTolerance)
      Serial.printf(D_STR_TOLERANCE " : %d%%\n",
kTolerancePercentage);
    // Display the basic output of what we found.
    Serial.print(resultToHumanReadableBasic(&results));
    // Display any extra A/C info if we have it.
    String description = IRACUtils::resultAcToString(&results);
    if (description.length()) Serial.println(D_STR_MSGDESC ": " +
description);
    yield(); // Feed the WDT as the text output can take a while
to print.
#ifdef LEGACY_TIMING_INFO
    // Output legacy RAW timing info of the result.
    Serial.println(resultToTimingInfo(&results));
    yield(); // Feed the WDT (again)
#endif // LEGACY_TIMING_INFO
    // Output the results as source code
    Serial.println(resultToSourceCode(&results));
    Serial.println(); // Blank line between entries
    yield(); // Feed the WDT (again)
  }
  OTALoopHandler();}
Base OTA.h
// Copyright 2020 Christian Nilsson (@nikize)
// Based on public Arduino BasicOTA example
#ifndef EXAMPLES_IRRECVDUMPV3_BASEOTA_H_
#define EXAMPLES_IRRECVDUMPV3_BASEOTA_H_
#ifndef OTA_ENABLE
#define OTA_ENABLE false
#endif // OTA_ENABLE
#ifdef OTA_ENABLE
#include <WiFi.h>
#include <ESPmDNS.h>
#include <WiFiUdp.h>
#include <ArduinoOTA.h>
void OTAwifi() {
  // start default wifi (previously saved on the ESP) for OTA
  WiFi.mode(WIFI_STA);

```

```

    WiFi.begin();
}
void OTAINIT() {
    // See BasicOTA ESP example for source and settings
    ArduinoOTA
        .onStart([]() {
            String type;
            if (ArduinoOTA.getCommand() == U_FLASH)
                type = "sketch";
            else
                type = "filesystem";
            Serial.println("Start updating " + type);
        })
        .onEnd([]() {
            Serial.println("\nEnd");
        })
        .onProgress([](unsigned int progress, unsigned int total) {
            Serial.printf("Progress: %u%%\r", (progress / (total /
100)));
        })
        .onError([](ota_error_t error) {
            Serial.printf("Error[%u]: ", error);
            if (error == OTA_AUTH_ERROR) Serial.println("Auth Failed");
            else if (error == OTA_BEGIN_ERROR) Serial.println("Begin
Failed");
            else if (error == OTA_CONNECT_ERROR) Serial.println("Connect
Failed");
            else if (error == OTA_RECEIVE_ERROR) Serial.println("Receive
Failed");
            else if (error == OTA_END_ERROR) Serial.println("End
Failed");
        });
    ArduinoOTA.begin();
    Serial.println();
    if (WiFi.waitForConnectResult() == WL_CONNECTED) {
        Serial.print("IP address: ");
        Serial.println(WiFi.localIP());
    } else {
        Serial.println("Wifi Connection Failed.");
    }
}
void OTALoopHandler() {
    ArduinoOTA.handle();
}
#else // OTA_ENABLE
void OTAwifi() {}
void OTAINIT() {}
void OTALoopHandler() {}
#endif // OTA_ENABLE
#endif // EXAMPLES_IRRECVDUMPV3_BASEOTA_H_

```

Lampiran 4

```
#Import Threading Multi Function
import threading
import time
import queue
#####
#####
#Open CV
import cv2
from ultralytics import YOLO
#####
#####
#Communication Serial to NodeMCU
import serial
import time
#####
#####
def PersonDetectionYOLO (GlobalCounterPerson):
    # Load YOLO v8 model
    model = YOLO ("YOLOV5nu.pt")
    # Class ID for person
    person_id = 0
    # Initialize webcam
    cam = cv2.VideoCapture(0)
    cam.set(3,640)
    cam.set(4,640)
    while cam.isOpened():
        ret, frame = cam.read()
        if not ret:
            break
        # Get model predictions
        results = model(frame)
        # Print results for debugging
        #print(results)
        #init Counter for 0
        CountPerson =0
        for result in results:
            boxes = result.boxes.xyxy # Bounding box coordinates
            labels = result.boxes.cls # Class labels
            scores = result.boxes.conf # Confidence scores
            for box, label, score in zip(boxes, labels, scores):
                if label == person_id:
                    CountPerson=CountPerson+1
                    #Write Variable to Global Variable
                    # Convert box coordinates to integers
                    x1, y1, x2, y2 = map(int, box)
                    # Draw bounding box
                    cv2.rectangle(frame, (x1, y1), (x2, y2), (255,
0, 255), 1)

                    # Put text on the frame
                    cv2.putText(frame, f'Person: {score:.2f}',
(x1, y1 - 10), cv2.FONT_HERSHEY_COMPLEX, 0.9, (255, 0, 255), 1)
                    GlobalCounterPerson.put(CountPerson)
                    # Display the frame
```

```

        cv2.rectangle(frame, (0, 0), (300, 75), (255, 255, 255),
cv2.FILLED)
        cv2.putText(frame, "Object Detection", (0, 33),
cv2.FONT_HERSHEY_COMPLEX, 0.9, (0, 0, 0), 1)
        cv2.putText(frame, f'Person: {CountPerson}', (0, 60),
cv2.FONT_HERSHEY_COMPLEX, 0.6, (0, 0, 0), 1)
        cv2.imshow("Webcam", frame)
        #Print Person
        print(f'Counter Person:{CountPerson}')
        # Exit program on 'q' key press
        if cv2.waitKey(1) == ord('q'):
            print("Quit")
            break
    # Release the camera and destroy all windows
    cam.release()
    cv2.destroyAllWindows()
def CommunicationSerial(GlobalCounterPerson):
    ser=serial.Serial(
        port='COM3',
        baudrate=9600,
        timeout=2 )
    if not ser.is_open:
        ser.open()
    while True:
        TempPerson=GlobalCounterPerson.get()
#
        if (TempPerson>=1 and TempPerson<=3):
            ser.write("a".encode('utf-8'))
            #print("LOW")
        elif (TempPerson>=4 and TempPerson<=6):
            ser.write("b".encode('utf-8'))
            #print("MEDIUM")
        elif (TempPerson>=7):
            ser.write("c".encode('utf-8'))
            #print("HIGH")
        else:
            ser.write("n".encode('utf-8'))

        if ser.in_waiting>0:
            data=ser.readline().decode('utf-8').rstrip()
            print(f'Received:{data}')
        time.sleep(1)
    ser.close()
#OverShared Variable
#GlobalVarTemp=queue.Queue()
#GlobalVarHumd=queue.Queue()
GlobalCounterPerson=queue.Queue()
#Create Threading
THPersonDetect=threading.Thread(target=PersonDetectionYOLO
, args=(GlobalCounterPerson,))
#THReadSensor=threading.Thread(target=ReadSensorDHT, args=(GlobalVa
rTemp, GlobalVarHumd,))
THSerialCom=threading.Thread(target=CommunicationSerial, args=(Glob
alCounterPerson,))
#THTampil=threading.Thread(target=Tampil, args=(GlobalVarTemp, Globa
lVarHumd,))

```

```
#Start Function
THPersonDetect.start()
#THReadSensor.start()
THSerialCom.start()
#THTampil.start()
#THReadSensor.join()
THPersonDetect.join()
THSerialCom.join()
#THTampil.join()
#GlobalVarTemp.join()
#GlobalVarHumd.join()
GlobalCounterPerson.join()
```

