

**RANCANG BANGUN ROBOT KONTROL *MOBILE*
PENGAWASAN BERBASIS *IOT (INTERNET OF THINGS)*
MENGUNAKAN KAMERA *ESP-32***

TUGAS AKHIR



ITEBA
Institut Teknologi Batam

**OLEH:
ANDRIAN SYAH
1922009**

**PROGRAM STUDI TEKNIK KOMPUTER
FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI BATAM
2023**

**RANCANG BANGUN ROBOT KONTROL *MOBILE*
PENGAWASAN BERBASIS *IOT*(*INTERNET OF THINGS*)
MENGUNAKAN KAMERA *ESP-32***

TUGAS AKHIR

**DIAJUKAN SEBAGAI SALAH SATU SYARAT UNTUK
MENDAPATKAN GELAR SARJANA(STRATA-1) PADA PROGRAM
STUDI TEKNIK KOMPUTER**



ITEBA

Institut Teknologi Batam

**OLEH:
ANDRIAN SYAH
NIM 1922009**

**FAKULTAS TEKNOLOGI INFORMASI
INSTITUT TEKNOLOGI BATAM
2023**

KATA PENGANTAR

Puji syukur saya panjatkan kehadiran Tuhan Yang Maha Esa atas segala rahmat-Nya sehingga Tugas Akhir ini dapat tersusun sampai dengan selesai dengan judul “Rancang Bangun Robot Kontrol *Mobile* Pengawasan Berbasis *IoT* (Internet Of Things) Menggunakan Kamera *ESP-32*”, salah satu syarat untuk memperoleh gelar Program Studi S1 Teknik Komputer, Fakultas Teknologi Informasi, Institut Teknologi Batam.

Penulis merasa bahwa masih banyak kekurangan dalam penyusunan Penelitian ini karena keterbatasan pengetahuan dan pengalaman penulis. Dalam pembuatan Penelitian ini juga tidak terlepas doa dan bantuan serta bimbingan dari berbagai pihak, oleh karena itu penulis ingin berterimakasih kepada :

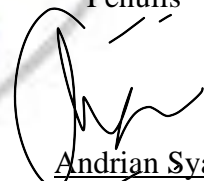
1. Allah Subhanahu wa Ta’ala, yang telah memberikan nikmat, rahmat, dan karunia kepada penulis.
2. Kedua orang tua yaitu Bapak Sumardi dan Ibu Kadariah atas segala do’a yang senantiasa mengiringi setiap langkah penulis dalam penyusunan Tugas Akhir ini.
3. Bapak Prof. Dr. Ing. Ir. H. Hairul Abral selaku Rektor Institut Teknologi Batam.
4. Ibu Dr. Ir. Ririt Dwiputri Permatasari, S.T., M.SI. selaku Dekan Fakultas Teknologi Informasi, Institut Teknologi Batam.
5. Bapak Muhammad Abrar Masril, M.Kom. selaku Ketua Program Studi Teknik Komputer, Fakultas Teknologi Informasi, Institut Teknologi Batam. Dan sebagai Dosen Pembimbing II Pembimbing Skripsi Tugas Akhir.
6. Bapak Deosa Putra Caniago, M.Kom selaku dosen pembimbing I yang senantiasa membimbing penulis dalam menyelesaikan pelaksanaan dan penulisan Tugas Akhir.
7. Bapak Zainal Munir, S.T., Me.TC selaku dosen penguji I pada sidang akhir penulis.
8. Bapak Joni Eka Candra, M.T. selaku dosen penguji II pada sidang akhir penulis.

9. Badan Riset dan Inovasi Nasional (BRIN) yang sudah memberikan support Dana Pendidikan dalam penelitian penulis sehingga dapat menyelesaikan Tugas Akhir tepat waktu.
10. PT.Pundi Mas Berjaya Selaku Perusahaan Tempat Kerja Praktik Penulis.
11. Rekan-Rekan Magang PT.Pundi Mas Berjaya khususnya Deven Lee, Azri Irawan, dan Rekan-rekan peserta magang MBKM Batch 3.
12. Rekan-rekan yaitu Hani Khairiyah, Kevin Antoni Kasimilale, Maranti Nainggolan, Muhamad Arie, Rizky Fadillah dan Seluruh Rekan-Rekan Mahasiswa Program Studi Teknik Komputer Institut Teknologi Batam Angkatan 2019 yang telah Bersama-sama mendorong, memotivasi, dan memberikan masukan kepada penulis.
13. Seluruh Pihak yang telah membantu penulis yang tidak dapat penulis sebutkan satu per satu.

Akhirnya penulis berharap semoga Tugas Akhir ini dapat bermanfaat bagi semua pihak.

Batam, 15 Oktober 2023

Penulis



Andrian Syah
NIM: 1922009

HALAMAN PENGESAHAN

Tugas akhir ini berjudul Rancang Bangun Robot Kontrol *Mobile* Pengawasan Berbasis *IoT(Internet Of Things)* Menggunakan Kamera Esp-32 ditulis dan diserahkan oleh Andrian Syah – 1922009 sebagai salah satu syarat untuk meraih gelar Sarjana Teknik Komputer, telah diperiksa dan oleh karena itu direkomendasikan untuk disahkan dan diterima.



Deosa Putra Caniago, M.Kom
NIDN : 1003129101

Tanggal: 18 Oktober 2023

Pembimbing I



Muhammad Abrar Masril, M.Kom
NIDN : 1027089203

Tanggal: 18 Oktober 2023

Pembimbing II

PENGUJI

Disahkan oleh penguji pada ujian Tugas Akhir
18/10/2023



Zainul Munir, S.T., Me.TC
NIDN : 1014088002

Tanggal: 18 Oktober 2023

Ketua Penguji

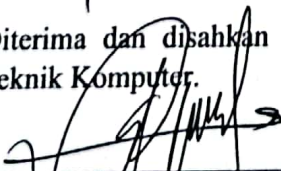


Joni Eka Candra, S.T., M.T.
NIDN : 1025068201

Tanggal: 18 Oktober 2023

Anggota Penguji

Diterima dan disahkan sebagai salah satu syarat untuk meraih gelar Sarjana Teknik Komputer.



Muhammad Abrar Masril, M.Kom
NIDN : 1027089203

Tanggal: 18 Oktober 2023

Ketua Program Studi Teknik Komputer

ABSTRAK

Teknologi Robot kini tidak hanya membantu manusia dalam melakukan segala aktifitas, namun teknologi yang ada sekarang membantu manusia untuk dapat memudahkan di dalam melakukan pengawasan dalam berbagai lingkungan, dari banyak nya kasus kejahatan dan beberapa kasus tindak kriminal yang ada di Tanjungpinang penelitian ini dilakukan di perumahan, untuk membantu keamanan yang sebelumnya konvensional menjadi lebih *flexible*, dengan metode pada penelitian ini adalah *Experiment* yang digunakan adalah Robot Kontrol *Mobile* berbasis IoT(*Internet Of Things*), robot ini menggunakan *Mikrokontroller ESP32Cam* dengan Output yang digunakan menggunakan Teknologi *WebSocket* sehingga pengguna dapat menggunakan secara efisien dan *realtime*, robot ini dilengkapi juga dengan *Pant Tilt Servo* yang digunakan untuk menggerakkan kamera Sekaligus *Mikrokontroller*. Pengujian dan pengambilan *sample* data yang digunakan adalah dengan menguji kekuatan sinyal, jarak tempuh, dan sistem kontrol. Hasil yang diperoleh dalam beberapa pengujian alat adalah pada jarak tempuh 16 Meter dengan Kekuatan Sinyal -70 dbm tanpa adanya halangan, dengan hasil kesimpulan bahwa robot dapat di kendalikan secara nirkabel dan dapat dikontrol menggunakan *WebSocket*.

Kata Kunci : *ESP32Cam, IoT, Mikrokontroller, Robot Mobile, WebSocket*

ABSTRACT

Robot technology nowadays not only assists humans in various activities but also aids in surveillance across different environments, given the increasing cases of crime and criminal incidents in Tanjungpinang. This research is conducted in residential areas to enhance security, which was previously conventional, and make it more flexible. The research method used is experimentation, employing an IoT (Internet of Things)-based Mobile Control Robot. The robot is equipped with an ESP32Cam microcontroller and utilizes WebSocket technology for real-time and efficient control. Additionally, the robot features a Pant Tilt Servo for camera movement along with the microcontroller. Testing and data collection involve assessing signal strength, range, and control system. The results of several tests indicate that the robot has a range of 16 meters with a signal strength of -70 dBm without obstructions. In conclusion, the robot can be wirelessly controlled and managed using WebSocket.

Keywords : ESP32Cam, IoT, Mikrokontroller, Robot Mobile, WebSocket

DAFTAR ISI

HALAMAN SAMPUL	i
HALAMAN JUDUL	ii
KATA PENGANTAR	iii
HALAMAN PENGESAHAN	v
ABSTRAK	vi
ABSTRACT	vii
DAFTAR ISI	viii
DAFTAR GAMBAR	xi
DAFTAR TABEL	xii
BAB I PENDAHULUAN	1
1.1 Latar Belakang	1
1.2 Rumusan Masalah	2
1.3 Batasan Masalah.....	2
1.4 Tujuan Penelitian	3
1.5 Manfaat Penelitian	3
1.6 Metode Penulisan	3
BAB II TINJAUAN PUSTAKA	7
2.1 <i>Internet Of Things(IoT)</i>	7
2.1.1 Keamanan Berbasis IoT	6
2.1.2 Keamanan Lingkungan	6
2.2 <i>Flowchart</i>	7
2.3 <i>Metode Waterfall</i>	9
2.4 <i>Data Flow Diagram</i>	10
2.5 <i>Contex Diagram</i>	11
2.6 <i>Arduino IDE</i>	11
2.7 <i>NodeMCU ESP-8266</i>	13
2.7.1 <i>ESP-32 CAM</i>	14
2.8 <i>Motor DC</i>	16
2.8.1 <i>Gearbox</i>	17
2.9 <i>Driver Motor L298N</i>	17
2.10 <i>Motor Servo SG90</i>	19
2.11 <i>Regulator Converter</i>	20

2.12 Web Application.....	21
2.13 Penelitian Terdahulu	21
BAB III METODE PENELITIAN	26
3.1 Metode Penelitian.....	26
3.2 Kerangka Kerja Penelitian	26
3.3 Metode Perancangan Sistem	26
3.4 Lokasi Penelitian.....	27
3.5 Alat dan Bahan.....	27
BAB IV PERANCANGAN DAN ALAT	25
4.1 Analisis Sistem.....	25
4.1.1 <i>Contex Diagram</i>	25
4.1.2 <i>Data Flow Diagram</i>	32
4.1.3 <i>Blok Diagram</i>	33
4.2 Perancangan Sistem	34
4.2.1 Perancangan Perangkat Keras (<i>Hardware</i>).....	34
4.2.2 Perancangan <i>Hardware</i> pada ESP32cam.....	34
4.2.3 Perancangan <i>Pant Tilt Servo</i>	35
4.2.4 Perancangan Motor <i>DC Gearbox</i>	35
4.2.5 Perancangan Keseluruhan <i>Hardware</i>	36
4.3 Perancangan Perangkat Lunak (<i>Software</i>)	37
4.3.1 Implementasi <i>Arduino IDE</i>	38
4.3.2 Implementasi Kontrol Robot berbasis <i>Websockets</i>	39
BAB V HASIL DAN PEMBAHASAN	45
5.1 Implementasi	45
5.2 Hasil Pengujian	45
5.2.1 Pengujian Jarak	45
5.2.1.1 Langkah Pengujian.....	45
5.2.1.2 Hasil Pengujian Jarak Tempuh.....	45
5.2.2 Pengujian Integrasi Jaringan	46
5.2.2.1 Langkah Pengujian Integrasi Jaringan	46
5.2.2.2 Hasil Pengujian Integrasi Jaringan.....	46
5.2.3 Pengujian Sistem Kontrol	48
5.2.3.1 Langkah Pengujian Sistem Kontrol	48
5.2.3.2 Hasil Pengujian Sistem Kontrol.....	48

BAB VI KESIMPULAN DAN SARAN.....	50
6.1 Kesimpulan	50
6.2 Saran.....	50

DAFTAR PUSTAKA

LAMPIRAN *SOURCECODE*

LAMPIRAN GAMBAR

DAFTAR GAMBAR

Gambar 2.1 Logo Arduino	11
Gambar 2.2 Aplikasi Arduino IDE	12
Gambar 2.3 Pinout ESP-8266	13
Gambar 2.4 Pinout ESP-32 Cam.....	14
Gambar 2.5 Motor DC	16
Gambar 2.6 Motor DC dengan Gearbox	17
Gambar 2.7 PinOut Motor Driver L298N	17
Gambar 2.8 Diagram PinOut Driver motor L298N.....	18
Gambar 2.9 Motor Servo SG90.....	19
Gambar 2.10 Converter LM2596	20
Gambar 2.11 Diagram Schematic Converter LM2596	20
Gambar 3.1 Diagram Kerangka Kerja	26
Gambar 3.2 Metode Waterfall	26
Gambar 4.1 Context Diagram Sistem Robot Pengawasan	25
Gambar 4.2 Data Flow Diagram Sistem Robot Pengawas.....	32
Gambar 4.3 Block Diagram Sistem.....	33
Gambar 4.4 Schematic ESP-32 Cam.....	34
Gambar 4.5 Schematic Pant Tilt.....	35
Gambar 4.6 Schematic Motor DC Gearbox.....	36
Gambar 4.7 Perancangan Schematic Keseluruhan.....	36
Gambar 4.8 Flowchart Keluruhan Sistem	37
Gambar 4.9 Contoh Keseluruhan program	38
Gambar 4.10 Kontrol Web User	39

DAFTAR TABEL

Table 2.1 <i>Flowchart</i> Simbol.....	7
Table 2.2 Spesifikasi <i>ESP-32 Cam</i>	15
Table 2.3 Spesifikasi Motor DC	16
Table 2.4 Spesifikasi Motor Driver L298N	18
Table 2.5 Spesifikasi Motor Servo SG90.....	19
Table 2.6 Spesifikasi LM2596	21
Table 2.7 Literatur Penelitian Terdahulu.....	21
Table 3.1 Jenis refrensi yang dikumpulkan.....	24
Table 3.2 Perangkat Keras	27
Table 5.1 Hasil Pengujian Jarak Tempuh	45
Table 5.2 Pengujian Integrasi Jaringan	47
Table 5.3 Tabel Uji Sistem Kontrol	48

BAB I

PENDAHULUAN

1.1 Latar Belakang

Dalam beberapa tahun terakhir dunia teknologi informasi berkembang sangat pesat, dengan adanya teknologi seperti saat ini memberikan kemudahan bagi setiap kalangan yang ada di dunia informasi dan teknologi, tentu saja pada bidang-bidang tertentu dari perkembangan teknologi informasi tersebut banyak dampak positif terutama dalam bidang pendidikan dan industri yang dapat dicontohkan seperti Teknologi Robotika (Arby et al., 2022).

Robotika merupakan suatu cabang teknologi yang berperan sebagai perangkat yang di program otomatis dan beberapa peralatan manipulator yang dapat diprogram untuk melakukan beberapa fungsi seperti memindahkan barang, komponen, peralatan maupun alat khusus lainnya. (Zhang et al., 2019). Pada dasarnya robot adalah otomasi internal yang bergerak menggunakan tenaga tertentu seperti listrik, hidrolik atau pneumatik yang dikendalikan menggunakan program dengan urutan tertentu (Ulinuha, 2020).

Beberapa robot memiliki beberapa sistem tertanam yang digunakan untuk mengontrol bagian-bagian yang sesuai dengan permintaan, dari adanya *Internet Of Things (IoT)*, *IoT* adalah konsep membuat suatu teknologi yang digunakan untuk mengontrol suatu perangkat menggunakan sistem server dan sebuah remote yang dapat diakses seperti komunikasi, mengendalikan, menghubungkan, selama masih terhubung dengan internet (Isrofi et al., 2021).

Dari Penjelasan Robot dan Teknologi *IoT* tersebut dapat diambil kesimpulan bahwa faktor pendorong fleksibilitas tentu saja merupakan faktor yang paling umum dalam pembuatan robot, karena robot didesain dengan menerapkan konsep menggunakan sebuah sensor, aktuator, dan komponen yang saling terkoneksi antara satu dan lain nya (Latifa & Slamet Saputro, 2018).

Berdasarkan berita artikel (A. Setiawan, 2022), tingkat kriminalitas pada tahun 2022 yang ada di kota Tanjungpinang meningkat menjadi 292 kasus dibanding pada tahun 2021 yang masih didalam situasi pandemi yaitu sebanyak

214 kasus. Dari permasalahan tersebut penulis mengambil objek penelitian pada sebuah perumahan yang mengalami permasalahan pengawasan lingkungan yang masih bersifat konvensional atau *siskamling* (*sistem keamanan keliling*) (Eulaerts & Joanny, 2022).

Dari permasalahan diatas penulis menjadikan robot pengawasan menjadi solusi atas permasalahan yang berkaitan dengan sistem keamanan yang ada di perumahan tersebut. robot yang ada dan dapat diarahkan melalui *Websocket*, cara kerja robot ini adalah dapat diarahkan untuk berjalan dan mengontrol kamera . Untuk tingkat keamanan pada robot ini memiliki kelebihan dibandingkan dengan penggunaan *CCTV* , penggunaan robot ini lebih fleksibel dan efisien.

Berdasarkan Latar belakang Permasalahan yang ada tersebut, maka penulis tertarik mengangkat Judul Penelitian tugas akhir ini yaitu **“Rancang Bangun Robot Kontrol Mobile Pengawasan Berbasis IoT(Internet Of Things) Menggunakan Kamera ESP-32”**.

1.2 Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan diatas, dapat dirumuskan permasalahan sebagai berikut :

1. Bagaimana Cara Merancang dan membangun sebuah robot kontrol *mobile* pengawasan menggunakan ESP-32 Cam ?
2. Bagaimana Cara Mengendalikan Robot menggunakan *Websocket Controller* dengan kontrol menggunakan jaringan nirkabel ?

1.3 Batasan Masalah

Agar penulis tidak keluar dari permasalahan yang ada dan hasil penelitian dapat diperoleh dengan baik, sempurna, dan terarah, maka penulis membatasi lingkup penelitian sebagai berikut :

1. Pada Penelitian ini hanya akan memfokuskan pada perancangan dan pembuatan Robot Pengawasan dengan pengaplikasian Modul *Mikrokontroller* kamera ESP-32.

2. Website yang dibangun akan mengambil data dari *Mikrokontroller ESP-32 Cam* dan menampilkan data tersebut pada bentuk aplikasi untuk memudahkan pengguna dalam mengendalikan Robot.

1.4 Tujuan Penelitian

Adapun beberapa Tujuan penulis pada penelitian ini sebagai berikut :

1. Penelitian ini bertujuan untuk membuat rancangan robot yang dapat di implementasikan untuk pengawasan di lingkungan seperti Perumahan dan Industri. Robot ini akan dirancang dengan kemampuan mobilitas agar dapat bergerak secara bebas di area yang akan diawasi.
2. Pengguna akan dapat mengendalikan robot pengawas dengan menggunakan perangkat kontrol *mobile*, seperti *smartphone* ataupun *tablet*.

1.5 Manfaat Penelitian

1. Memperoleh pengetahuan dan pengalaman dalam merancang dan membangun sebuah sistem *IoT* pada robot Pengawasan .
2. Masyarakat dapat memanfaatkan Robot Kontrol *Mobile* Pengawasan sebagai alat bantu untuk melakukan pengawasan pada lingkungan tertentu, seperti rumah atau kantor.

1.6 Metode Penulisan

Penelitian ini mencoba untuk membahas terkait dengan pokok permasalahan secara cepat dan sistematis. Untuk pembahasan akan dibuat dan disusun dengan sistematika penulisan yang telah di tetapkan sebagai berikut:

Bab I : Pendahuluan

Berisi Latar Belakang, Perumusan Masalah, Batasan Masalah, Tujuan Penelitian, Manfaat Penelitian, dan Sistematika Penulisan.

Bab II : Landasan Teori

Pada bab ini dijelaskan teori yang mendukung dalam beberapa penelitian yang sebelum nya mengenai penggunaan *Mikrokontroller* yang bersifat kompleks untuk memberikan *command* pada robot Pengawasan.

Bab III : Metode Penelitian

Pada Bab ini menjelaskan terkait metode Analisa yang digunakan dan penggunaan secara subjektif mengenai penggunaan sensor dan *Mikrokontroller* yang digunakan pada rangkaian robot Pengawasan.

Bab IV : Analisa dan Perancangan

Pada bab ini membahas terkait perancangan rangkaian robot *pengawasan* beserta skema diagram yang digunakan dalam rangkaian *Internet Of Things* dari Robot Pengawasan beserta sistem yang akan digunakan dalam pengujian pada bab selanjutnya.

Bab V : Implementasi dan Hasil

Pada Bab ini akan dilakukan beberapa skema pengujian dari hasil rancangan rangkaian yang telah dibuat untuk membandingkan penelitian teknologi yang terdahulu dengan rangkaian yang baru pada saat penelitian ini dilakukan.

Bab VI : Kesimpulan dan Saran

Pada Bab ini menjelaskan terkait dengan kesimpulan dan hasil dari penelitian mengenai rancang bangun Robot Pengawasan yang ada pada saat ini dengan sedikit penambahan rangkaian yang berguna untuk efisiensi perangkat dan yang akan diimplementasikan , serta memberikan saran bagi peneliti berikutnya.

BAB II

TINJAUAN PUSTAKA

2.1 Internet Of Things(IoT)

Internet of Things adalah teknologi jaringan dinamis yang secara otomatis mengkonfigurasi dirinya sendiri berdasarkan komunikasi standar. Dengan dukungan teknologi *cloud computing*, yang dimungkinkan untuk saling mengambil atau menyimpan data yang terhubung ke beberapa jaringan Internet (Caniago, 2022). Penggunaan *IoT* pada saat ini sangat berkembang pesat berbagai teknologi listrik dapat di kendalikan dari jarak jauh. Salah satu perkembangan *IoT* yang sering ditemukan adalah penggunaan *smartphone* yang mampu mengendalikan dan menerima informasi peralatan listrik atau sensor dari jarak yang jauh selama terhubung dengan internet.

Teknologi *Internet of Things* sendiri sudah berkembang diberbagai bidang mulai dari bidang agrikultur, transportasi, logistik, industri, pendidikan, keamanan bahkan dalam bidang militer, perlahan-lahan penggunaan teknologi *internet of things* akan menjadi suatu kebiasaan yang berkembang dalam kehidupan bermasyarakat. *Internet of things* sendiri terdapat lima unsur yang membentuk teknologi internet of thing yaitu(Adikara et al., 2021):

- 1) *Artificial intelligence* atau bisa disebut dengan kecerdasan buatan yang dimana suatu mesin atau alat mampu bergerak secara autonom dengan algoritma yang telah dimasukan, sehingga mesin mampu mengumpulkan data, algoritma, dan jaringan.
- 2) Konektifitas atau hubungan koneksi antar jaringan yang dimana memungkinkan terbentuknya jaringan skala kecil yang menghubungkan perangkat antar sistem.
- 3) Perangkat Berukuran Kecil Perkembangan teknologi saat ini membuat perangkat menjadi kecil sehingga teknologi *internet of things* dapat digunakan dengan nyaman dan efisien.

2.1.1 Keamanan Berbasis IoT

Pengertian sistem keamanan secara umum adalah pengamanan suatu benda dimana benda tersebut mengandung unsur-unsur penting yang perlu diamankan, misalnya rumah, ruangan, gedung atau yang lainnya. Sistem keamanan sangat penting untuk mencegah pencurian atau tindakan kriminal lainnya. Hal ini dibuat untuk mencegah tingkat kejahatan pencurian semakin meningkat setiap tahunnya (H et al., 2019). Kita sendiri bisa melihat berbagai macam teknologi keamanan yang tersebar dipasaran, variasi teknologi di kembangkan menyesuaikan kebutuhan dan keinginan dari masyarakat itu sendiri.

Perkembangan teknologi saat ini sangat pesat, salah satunya ialah perkembangan teknologi keamanan berbasis *internet of things* yang sangat bermanfaat dan inovasi terus diciptakan (Agustina, 2023). Dengan perkembangan teknologi keamanan ini diharapkan bisa menekan angka kriminalitas khususnya kasus pencurian. Dalam penulisan laporan tugas akhir ini penulis menggunakan teknologi *internet of things* yang diterapkan pada robot yang dapat di Kontrol melalui jarak jauh.

2.1.2 Keamanan Lingkungan

Keamanan Lingkungan dibuat agar terciptanya aman dan tentram di lingkungan. ada nya beberapa teknologi yang digunakan pada saat ini menunjukan perkembangan yang sangat pesat. Sehingga ada beberapa teknologi yang digunakan sebagai penunjang untuk melakukan aktifitas keamanan dan ketentraman pada lingkungan tersebut, Berikut ini beberapa keamanan lingkungan yang diterapkan antara lain :

1) CCTV (Closed Circuit Television)

Pemasangan kamera *CCTV* di area strategis di lingkungan RT atau RW dapat membantu memantau aktivitas dan mendeteksi potensi ancaman keamanan. *CCTV* dapat dipasang di titik-titik masuk, jalan utama, atau area yang sering dilalui warga. Dengan adanya *CCTV*,

aktivitas yang mencurigakan dapat terdeteksi dan tindakan dapat diambil lebih cepat.

2) Aplikasi Pengawasan Lingkungan



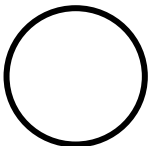
Penggunaan aplikasi pengawasan lingkungan yang dapat diakses oleh warga dapat membantu dalam melaporkan dan memantau situasi keamanan. Aplikasi tersebut dapat digunakan untuk melaporkan kejadian mencurigakan, meminta bantuan, atau berbagi informasi penting kepada warga lainnya.

2.2 Flowchart

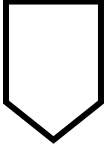




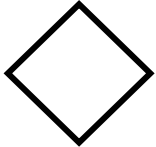


Flowchart ini berfungsi untuk menggambarkan dan menyederhanakan suatu rangkaian proses atau prosedur sehingga mudah untuk dipahami dan mudah dilihat berdasarkan urutan langkah-langkah dari suatu proses. *Flowchart* digunakan untuk mempermudah pemahaman apa saja yang diperlukan dalam menjalankan suatu program (Ilham & Sulasmoro, n.d. 2021).

Flowchart memiliki simbol yang memiliki masing-masing dari setiap simbol memiliki arti disetiap bentuk simbolnya, simbol tersebut akan disusun membentuk suatu urutan sebuah konstruksi atau program. Untuk memahami bentuk dan arti *Flowchart* dapat dilihat pada Tabel berikut ini.



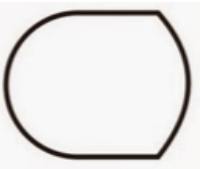


Tabel 2.1 *Flowchart* Simbol

No	Simbol	Nama	Keterangan
1		<i>Flow Direction</i>	Simbol yang digunakan untuk menghubungkan simbol dengan simbol lainnya
2		Terminal	Simbol yang digunakan untuk menunjukkan mulai (<i>start</i>) atau akhir (<i>end</i>) suatu kegiatan.
3		<i>Connector</i>	Simbol yang menunjukan keluar masuk atau sambungan pada halaman yang sama.

Lanjutan Tabel 2.1 Flowchart Simbol

No	Simbol	Nama	Keterangan
4		<i>Connector</i>	Simbol yang menunjukkan keluar masuk atau sambungan padahalaman yang berbeda.
5		<i>Processing</i>	Simbol yang menunjukkan pengolahan yang dilakukan komputer
6		<i>Manual Operation</i>	Simbol yang menunjukkan pengolahan yang tidak dilakukan oleh komputer.
7		<i>Simbol Document</i>	Simbol menyatakan input atau output dalam bentuk kertas.
8		<i>Simbol Punch Card</i>	Simbol menyatakan input atau output dalam bentuk kartu.
9		<i>Simbol Decision</i>	Simbol menyatakan pemilihan proses pada kondisi yang ada.
10		<i>Simbol Input-Output</i>	Simbol menyatakan proses input atau output tanpa bergantung pada jenis peralatannya.
11		<i>Simbol Manual Input</i>	Simbol menyatakan pemasukan <i>data</i> secara manual.

Lanjutan Tabel 2.1 *Flowchart* Simbol

No	Simbol	Nama	Keterangan
12		Simbol <i>Preparation</i>	Simbol yang menyatakan mempersiapkan penyimpanan data yang akan digunakan.
13		Simbol <i>Defined Process</i>	Simbol pelaksanaan suatu bagian
14		Simbol <i>Display</i>	Simbol menyatakan peralatan output yang digunakan seperti printer, layar dan sebagainya.
15		Simbol <i>Disk and Online Storage</i>	Simbol menyatakan input berasal dari disk.
16		Simbol mengetik <i>tape Unit</i>	Simbol menyatakan input dari pita magnetic.


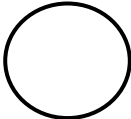


2.3 Metode Waterfall

Waterfall merupakan model *SDLC* yang biasa digunakan dalam pengembangan perangkat lunak atau perangkat keras yang berfungsi untuk menyediakan pendekatan alur rangkaian yang akan digunakan (Hidayat et al., 2017). Pada metode ini mempunyai rangkaian dan alur kerja yang sistem nya jelas dan terukur, Pada setiap komponen memiliki tugas masing-masing sesuai dengan instruksi yang diberikan. Kelebihan Metode *Waterfall* adalah Hasil dokumentasi yang baik dan dapat menghemat biaya.

2.4 Data Flow Diagram

DFD (Data Flow Diagram) adalah diagram yang menggambarkan aliran data dari proses yang biasa disebut sebagai sistem informasi. Diagram aliran data juga menyediakan informasi tentang input dan output dari setiap entitas dan proses itu sendiri. Juga tidak ada kontrol atas alirannya dalam diagram aliran data, sehingga tidak ada aturan yang terkait dengan keputusan atau pengulangan (Wardoyo & Liana Putri, 2023).



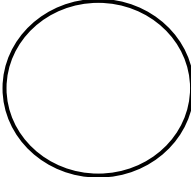
Tabel 2.2 Komponen *Data Flow Diagram*

Nama Simbol	Gambar Simbol	Keterangan
<i>External Entity</i>		Simbol ini digunakan untuk menggambarkan asal dan tujuan data
Proses		Simbol ini digunakan untuk proses pengolahan data transformasi data
<i>Data Flow</i>		Simbol ini digunakan untuk menggambarkan aliran data yang berjalan
<i>Data Store</i>		Simbol ini digunakan untuk menggambarkan data <i>flow</i> yang sudah disimpan atau diarsipkan

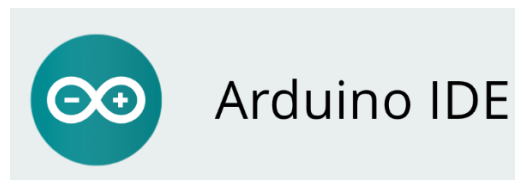
2.5 Context Diagram

Diagram konteks adalah diagram yang terdiri dari suatu proses dan menggambarkan ruang lingkup suatu sistem. Diagram konteks merupakan level tertinggi dari DFD yang menggambarkan seluruh input ke sistem atau output dari sistem. Pada *context* diagram akan memberi gambaran tentang keseluruhan sistem. Sistem dibatasi oleh boundary (dapat digambarkan dengan garis putus). Dalam diagram konteks hanya ada satu proses. Tidak boleh ada store dalam diagram konteks(Nurmalina, n.d.).

Tabel 2.3 Simbol Context Diagram

Nama Simbol	Gambar Simbol
Terminator	
Aliran Data/ Data <i>Flow</i>	
<i>Process</i>	

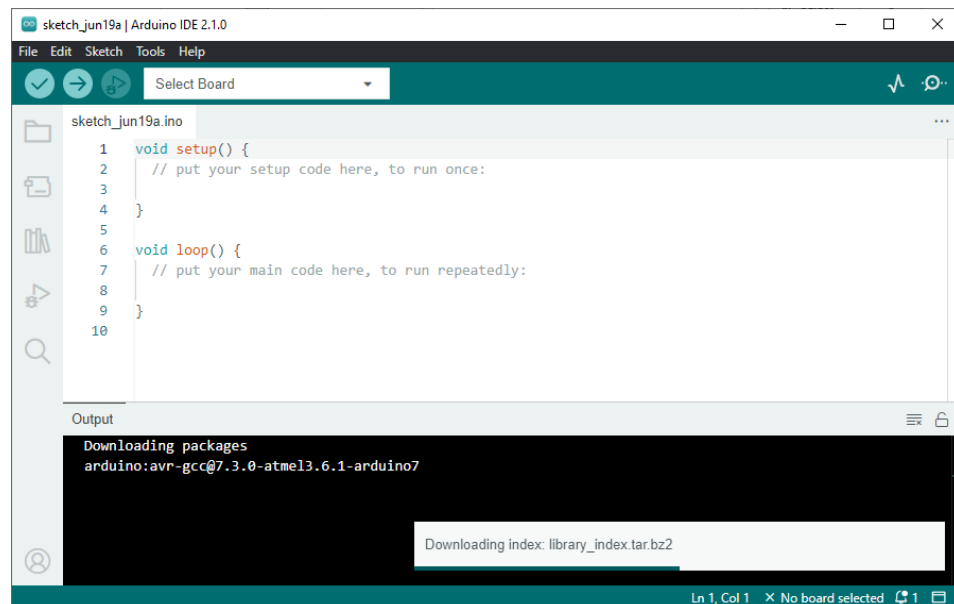
2.6 Arduino IDE



Gambar 2.1 Logo Arduino IDE

Arduino IDE merupakan *Software* yang digunakan untuk membuat sketsa (*sketch*). Sketsa merupakan program yang dibuat untuk mengontrol papan Arduino

dan berbagai jenis board yang ada pada *library* aplikasi tersebut, bahasa yang digunakan dalam penulisan sketsa pada aplikasi Arduino IDE adalah bahasa C (Hergika et al., 2021). Berikut ini beberapa fitur yang ada pada aplikasi Arduino IDE :

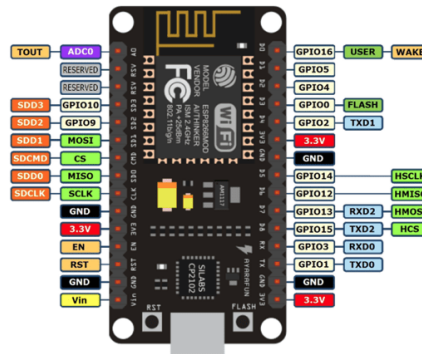


Gambar 2.2 Aplikasi Arduino IDE

- 1) *File* memiliki pilihan untuk membuat sketsa baru, menyimpan sketsa, membuka *preferences*, pilihan untuk keluar dari program dan yang lainnya.
- 2) *Edit* terdapat pilihan-pilihan seperti *Copy*, *Paste*, *Cut*, *Select All* untuk menyeleksi semua kode yang sudah ditulis dan yang lainnya.
- 3) *Sketch* terdapat pilihan seperti *Verify* yang digunakan untuk memverifikasi *sketch* yang telah dibuat, kemudian pilihan *Upload* yang digunakan untuk mengunggah *sketch* yang telah dibuat dan dikompilasi ke Arduino.
- 4) *Tools* terdapat beberapa pilihan submenu. Submenu yang biasa digunakan adalah pilihan untuk memilih jenis *Board* Arduino dan ESP yang digunakan terhubung dengan komputer dan pilihan untuk *port COM* di mana Arduino atau board yang lain tersebut terhubung dengan komputer.

- 5) *Help* terdapat beberapa pilihan yang dapat digunakan untuk mencari informasi berkaitan dengan board yang digunakan.

2.7 NodeMCU ESP-8266



Gambar 2.3 Pinout ESP-8266

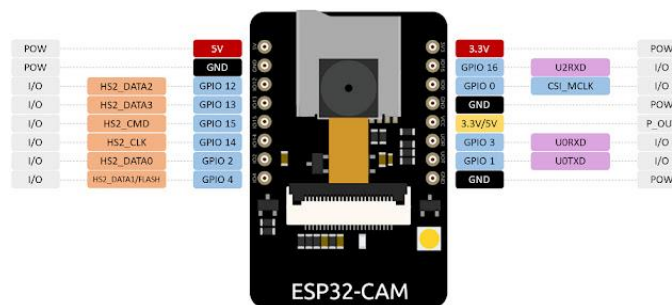
NodeMCU dilengkapi dengan chip ESP8266 yang dirancang khusus untuk memenuhi kebutuhan perangkat yang terhubung saat ini. Chip ini menawarkan solusi jaringan Wi-Fi yang lengkap dan mandiri, sehingga dapat berfungsi sebagai host aplikasi atau menangani semua fungsi jaringan *Wi-Fi* dari prosesor aplikasi lainnya. *NodeMCU* memiliki kemampuan pemrosesan dan penyimpanan yang kuat di dalamnya, memungkinkan integrasi dengan perangkat sensor tertentu. Hal ini sangat penting dalam pengembangan *Internet of Things (IoT)* karena *NodeMCU* dapat menyederhanakan dan mempermudah proses pengembangan aplikasi IoT(Parihar, 2019).

Adapun fungsi mengenai pin utama pada *NodeMCU* antara lain :

1. Vin (*Voltage In*): Pin Vin adalah pin yang digunakan untuk menyuplai tegangan ke *NodeMCU*. Tegangan yang biasanya digunakan adalah 5V. Pin ini dapat digunakan jika Anda ingin memberi daya pada *NodeMCU* melalui pin ini.
2. Pin *GND*(*Ground*) adalah pin yang digunakan untuk menghubungkan sirkuit dengan referensi ground. Pin ini digunakan untuk menjaga potensial pada sirkuit tetap stabil dan merujuk pada nol *Volt*.

3. Pin 3V3 adalah pin keluaran yang memberikan tegangan output 3.3V. Pin ini dapat digunakan untuk memberi daya pada sensor atau perangkat lain yang membutuhkan tegangan 3.3V.
4. Pin D0 hingga D8 adalah pin input/output digital yang dapat digunakan sebagai GPIO (*General Purpose Input/Output*). Pin ini dapat dikonfigurasi sebagai input atau output, dan digunakan untuk menghubungkan *NodeMCU* dengan berbagai perangkat eksternal seperti sensor, LED, motor, dan lainnya.
5. Pin A0 adalah pin analog input. Pin ini dapat digunakan untuk membaca tegangan analog dari sensor seperti sensor suhu, sensor kelembaban, atau sensor cahaya. *NodeMCU* menggunakan konverter analog ke digital internal untuk mengubah tegangan analog menjadi nilai digital.
6. Pin *RX* (*Receive*) dan *TX* (*Transmit*) adalah pin yang digunakan untuk komunikasi serial. Pin *RX* digunakan untuk menerima data serial, sedangkan pin *TX* digunakan untuk mengirim data serial. Anda dapat menggunakan pin ini untuk berkomunikasi dengan perangkat lain yang mendukung komunikasi serial seperti komputer, modul *Bluetooth*, atau modul *GSM*.

2.7.1 ESP-32 CAM



Gambar 2.4 Pinout ESP-32 Cam

ESP-32 CAM merupakan perangkat yang dikembangkan dari *ESP-32* dengan penambahan modul kamera dan *Slot Memory Card*. *ESP-32* merupakan perangkat yang sering digunakan dalam pembelajaran atau pengembangan teknologi *internet of things (IoT)* (Rusimamto et al., 2021). *ESP-32* sangat unggul dibanding dengan pedahulunya yaitu *ESP8266*. Salah satu keunggulan

dari *Mikrokontroller ESP-32 CAM* dengan *ESP8266* dapat dilihat dari jumlah yang disediakan. *ESP-32 CAM* terdapat 32 Pin sedang *ESP8266* terdapat 17 pin.

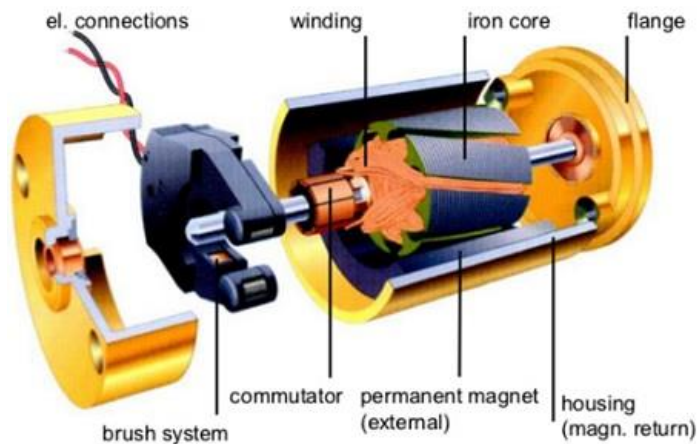
Tabel 2.4 Spesifikasi *ESP-32 Cam*

Fitur	Spesifikasi
Mikrokontroler	Tipe: ESP32
	Kecepatan CPU: Hingga 240 MHz
	Memori Flash: 4 MB
Kamera	Sensor gambar: OV2640
	Resolusi: 2 Megapiksel (1600 x 1200 piksel)
	Format gambar: JPEG
	Antarmuka: UART, SPI, I2C, I2S, ADC, DAC, PWM, GPIO
Komunikasi	Wi-Fi: 802.11 b/g/n
	Bluetooth: Bluetooth v4.2 BR/EDR dan BLE
	Protokol jaringan: TCP/IP, HTTP, FTP, MQTT, SSL/TLS, UDP, DNS

ESP-32 CAM terdapat susunan pin yang memiliki fungsi masing-masing. Pin yang disusun dalam *ESP-32 CAM* antara lain:

- 1) *Power* pin Merupakan pin yang terdiri atas 5V, 3.3 V dan *GND* yang digunakan sebagai sumber daya yang dihubungkan ke komponen lain.
- 2) *Serial* Pin *Serial* digunakan sebagai *upload* karena *ESP32 CAM* tidak memiliki *built-in Programmer*. Pin serial terdapat pada pin GPIO 1 sebagai *U0TXD* dan pin GPIO 3 sebagai *U0RXD*.
- 3) *GPIO 0* Digunakan sebagai *upload* untuk *setting ESP32 CAM* dengan cara dihubungkan ke *GND*.
- 4) *Module Camera* Modul kamera digunakan untuk merekam gambar atau video.
- 5) *SD Card Reader* Digunakan untuk menyimpan *data*.

2.8 Motor DC



Gambar 2.5 Motor DC

Penggunaan motor listrik DC saat ini paling banyak digunakan pada sektor industri. Hal ini terjadi karena beberapa keunggulan motor DC. Keunggulan motor DC antara lain torsi yang baik, putaran yang panjang, pengereman yang baik, pengaturan kecepatan yang baik sehingga mudah dikendalikan. Dalam aplikasi lapangan, motor DC umumnya digunakan di banyak sektor penggerak listrik yang memerlukan pengaturan kecepatan, termasuk mesin derek dan rolling mill serta peralatan mesin presisi. Pengaturan kecepatan motor DC beroperasi berdasarkan teori kontrol umpan balik (F. B. Setiawan et al., 2022).

Spesifikasi yang dimiliki oleh Motor DC yaitu pada tabel sebagai berikut :

Tabel 2.5 Spesifikasi Motor DC

Spesifikasi	Nilai
Tegangan	12 Volt
Tipe motor	Motor DC
Tipe gearbox	Gearbox
Rasio gearbox	01:50
Kecepatan putar tanpa beban	1000 RPM

Lanjutan Tabel 2.6 Spesifikasi Motor DC

Torsi maksimum	2 kg.cm
Arus beban maksimum	1 Ampere
Efisiensi	80%
Jenis penggerak	Brushed
Berat	200 gram
Dimensi	5 cm x 5 cm x 5 cm
Aplikasi tipikal	Robotika, otomasi, peralatan elektronik

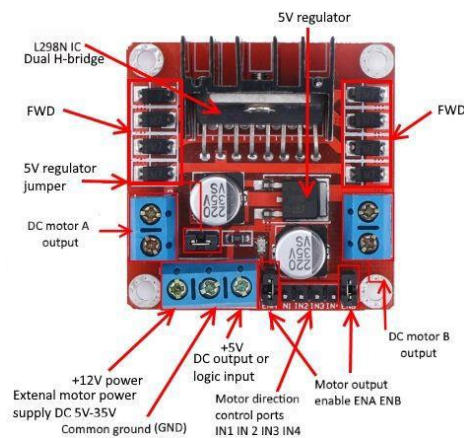
2.8.1 Gearbox



Gambar 2.6 Motor DC dengan Gearbox

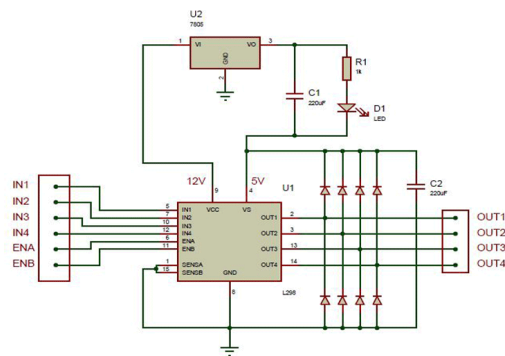
Gearbox pada motor DC berfungsi untuk mengubah kecepatan dan torsi keluaran motor sesuai dengan kebutuhan aplikasi. Gearbox terdiri dari beberapa gigi dengan perbandingan gear yang berbeda-beda. Gigi-gigi ini dapat digunakan untuk mempercepat atau memperlambat kecepatan rotasi motor DC, sambil meningkatkan atau mengurangi torsi yang dihasilkan.

2.9 Driver Motor L298N



Gambar 2.7 PinOut Motor Driver L298N

Motor driver L298N adalah sebuah perangkat elektronik yang digunakan untuk mengendalikan dan menggerakkan motor DC (Direct Current) atau motor langkah. Motor driver L298N adalah jenis motor driver H-bridge, yang berarti dapat mengendalikan arah putaran motor dan mengatur kecepatannya. Perangkat ini menggunakan teknologi H-bridge, yang terdiri dari empat transistor daya (biasanya MOSFET atau BJT) yang dikonfigurasi dalam bentuk jembatan, sehingga memungkinkan aliran arus searah pada motor DC (Peerzada et al., 2021).



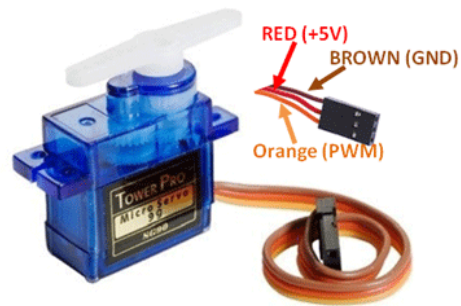
Gambar 2.8 Diagram PinOut Driver motor L298N

Spesifikasi yang dimiliki oleh Driver motor L298N yaitu sebagai berikut :

Tabel 2.7 Spesifikasi Motor Driver L298N

Fitur	Spesifikasi
Tipe	LM2596
Tegangan masukan	4.5 V - 40 V
Tegangan keluaran	1.25 V - 37 V
Arus keluaran	Hingga 3 A
Efisiensi	Hingga 92%
Tegangan dropout	Sekitar 1.5 V
Frekuensi operasi	150 kHz
Proteksi	Perlindungan termal, perlindungan arus pendek
Kontrol	Aplikasi potensiometer atau pin kontrol
Dimensi	Sekitar 43 mm x 20 mm x 14 mm

2.10 Motor Servo SG90



Gambar 2.9 Motor Servo SG90

Motor servo adalah sebuah motor dengan sistem *closed feedback* di mana posisi dari motor akan di informasikan kembali ke rangkaian kontrol yang ada di dalam motor servo. Dengan input ke kontrolnya yang bisa berupa sinyal analog ataupun sinyal digital, pada dasarnya motor servo banyak digunakan sebagai aktuator yang membutuhkan posisi putaran motor yang presisi (Salim et al., 2020).

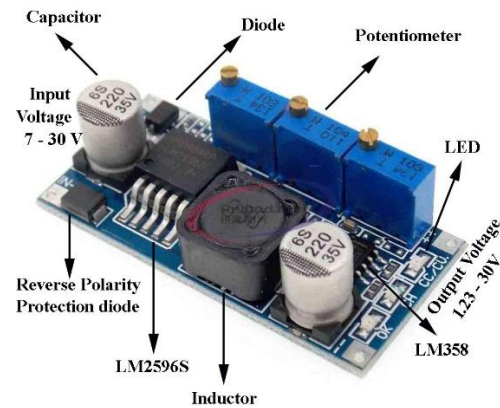
Berikut ini fungsi dari PinOut yang ada gambar diatas :

- 1) VCC(Merah): Pin ini terhubung ke sumber tegangan positif (biasanya 5V) untuk memberikan daya pada motor servo.
- 2) GND(Coklat): Pin ini terhubung ke ground (tanah) untuk menyediakan referensi ground bagi motor servo.
- 3) Sinyal (Orange): Pin ini digunakan untuk mengontrol posisi sudut motor servo. Sinyal PWM (Pulse Width Modulation) dikirim melalui pin ini untuk mengatur sudut rotasi motor.

Tabel 2.8 Spesifikasi Motor Servo SG90

Fitur	Spesifikasi
Tipe	Motor Servo SG90
Tegangan operasi	4.8 V - 6 V
Stall torque	1.8 kg/cm
Kecepatan tanpa beban	0.12 s/60°
Torsi kerja	1.6 kg/cm
Sudut rotasi	180°
Tegangan sinyal kontrol	5 V
Jumlah gigi	3
Dimensi	22.2 mm x 11.8 mm x 31 mm
Berat	Sekitar 9 g

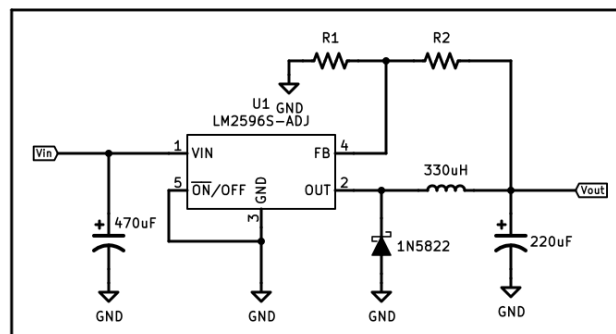
2.11 Regulator Converter



Gambar 2.10 Converter LM2596

Regulator converter merupakan jenis regulator *switching DC-DC* yang digunakan untuk menurunkan tegangan. Dibandingkan dengan regulator linear, buck converter memiliki beberapa keunggulan. Salah satunya adalah efisiensi daya yang lebih tinggi, terutama ketika terdapat perbedaan tegangan yang cukup besar antara tegangan masukan dan keluaran(Alsumady et al., 2021).

Berikut ini merupakan *Schematic Diagram* dari Regulator Converter LM2596



Gambar 2.11 Diagram Schematic Converter LM2596

Berikut ini merupakan Spesifikasi dari Converter LM2596 dalam bentuk tabel sebagai berikut :

Tabel 2.9 Spesifikasi LM2596

Fitur	Spesifikasi
Tipe	LM2596
Tegangan masukan	4.5 V - 40 V
Tegangan keluaran	1.25 V - 37 V
Arus keluaran	Hingga 3 A
Efisiensi	Hingga 92%
Tegangan dropout	Sekitar 1.5 V
Frekuensi operasi	150 kHz
Kontrol	Aplikasi potensiometer atau pin kontrol
Dimensi	Sekitar 43 mm x 20 mm x 14 mm

2.12 Web Application

Website Merupakan Sebuah halaman *Web* merupakan berkas yang ditulis sebagai berkas teks biasa (*plain text*) yang diatur dan dikombinasikan sedemikian rupa dengan instruksi-instruksi berbasis *HTML*(*Hypertext Markup Language*). yang dapat diakses jaringan wilayah lokal. (Hidayanto, 2015). Pada Penelitian ini penulis akan membuat sebuah *Website* yang dimana berfungsi menjadi aplikasi berbasis *Web* untuk menggerakkan robot Modul ESP-32 Cam.

2.13 Penelitian Terdahulu

Hasil penelitian terdahulu menjadi referensi untuk memperkaya teori dalam mengkaji penelitian ini. Dengan penelitian terdahulu penulis dapat mengetahui apa saja yang dapat dikembangkan dalam penelitian ini. Berikut literatur penelitian terdahulu yang digunakan penulis sebagai referensi dapat dilihat pada Tabel dibawah ini.

Tabel 2.10 Literatur Penelitian Terdahulu

NO	Nama Peneliti(Tahun)	Judul Penelitian	Hasil Penelitian
1	Angga Masri S.M Koroy, Gamaria Mandar dan Abdur Haris Muhammad (2020)	Rancang Bangun Sistem Keamanan Rumah Menggunakan ESP32 CAM	Penerapan <i>ESP32</i> <i>CAM</i> dan aplikasi telegram sebagai monitoring rumah

Lanjutan Tabel 2.11 Literatur Penelitian Terdahulu

2	Joni Prayitno , Harso Kurniadi(2018)	Pembuatan Sistem Kendali Robot Menggunakan Kamera Berbasis Android	Penerapan Robot Menggunakan <i>IP Cam</i> dengan aplikasi Android.
3	Al Mahdali, Lutfi ,Reinsinka Deby Melky, Sedik(2022)	Pengembangan <i>Wireless Remotely Operated Vehicle</i> Berbasis <i>Web</i>	Penerapan Robot Menggunakan Kontrol <i>Web</i> .
4	Azis Isrofi, Shoffin Nahwa Utama, Oddy Virgantara Putra (2021)	Rancang Bangun Robot Pemotong Rumput Otomatis Menggunakan Wireless Kontroler Modul Esp32-Cam Berbasis <i>Internet Of Things (IoT)</i>	Penerapan Robot Pemotong Rumput Otomatis dengan Beberapa pengujian berdasarkan waktu.
5	Winda Yulita, Aidil Afriansyah (2022)	Alat Pemantau Keamanan Rumah Berbasis Esp32-Cam	Penelitian mengenai persentase keberhasilan sesuai dengan pengujian dan penerimaan perintah seperti menghidupkan flash, menangkap pergerakan, pengambilan foto dan video secara manual.

BAB III

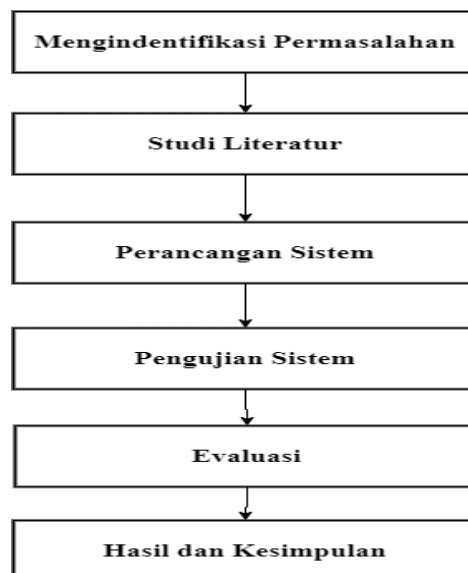
METODE PENELITIAN

3.1 Metode Penelitian

Metode penelitian merupakan cara atau teknik yang digunakan untuk mengumpulkan data atau informasi dalam sebuah penelitian. Metode yang digunakan sangat penting untuk memperoleh data yang valid dan variabel yang relevan terhadap topik penelitian. Metode penelitian harus sesuai dengan tujuan penelitian, waktu penelitian, sumber data, dan jenis data yang akan dikumpulkan (Darna & Herlina, 2018). Adapun dengan adanya metode penelitian, alur dan hasil penelitian yang penulis lakukan lebih jelas dan terstruktur sesuai dengan tema yang diambil oleh penulis.

3.2 Kerangka Kerja Penelitian

Dalam penelitian ini, kerangka kerja digunakan untuk persiapan penelitian yang lebih matang, lebih mudah dipahami serta memperjelas metodologi penelitian yang digunakan pada proses penelitian dalam pembuatan rancang bangun robot kontrol pengawasan. dengan kerangka kerja penelitian sebagai berikut.



Gambar 3.1 Diagram Kerangka Kerja

Kerangka kerja penelitian dapat diuraikan sebagai berikut:

1) Mengidentifikasi masalah

Pada Tahap ini, dilakukan indentifikasi terhadap pemasalahan yang diangkat dalam penelitian yaitu di perumahan griya senggarang permai yang menggunakan CCTV sehingga sehingga solusi permasalahan nya menambahkan robot pengawasan yang *flexible* .

2) Studi Literatur

Pada tahap ini, dilakukan pengumpulan informasi yang relevan terhadap topik penelitian. Tahap penelitian ini merupakan tahap pencarian dan pengumpulan artikel atau jurnal dari penelitian-penelitian sebelumnya yang terkait dengan penelitian ini.

Tabel 3.1 Jenis refrensi yang dikumpulkan

No	Nama	Jenis
1	<i>Flowchart</i>	Simbol
2	Arduino IDE	Aplikasi Open Source
3	NodeMCU	ESP-32 CAM
4	Motor DC	Gearbox
5	Driver Motor	L298N
6	Motor Servo	SG90
7	Regulator Converter	L25M96
8	<i>Web</i> Aplikasi	- HTML - CSS

Lanjutan Tabel 3.1 Jenis refrensi yang dikumpulkan

9	Penelitian Terdahulu	<ul style="list-style-type: none"> - Penerapan ESP-32 untuk Monitoring Rumah. - Penerapan ESP 32 dengan <i>IP Cam</i> Menggunakan Aplikasi <i>Android</i>. - Penerapat Robot Menggunakan Kontrol <i>Web</i>. - Penerapan Robot untuk pengujian berdasarkan waktu.
---	----------------------	---

3) Perancangan Sistem

Pada tahap ini, dilakukan perancangan *Hardware*, pembuatan *activity diagram*, *flowchart diagram* serta pembuatan aplikasi integrasi berbasis *Web Appllication* untuk menggerakkan robot sesuai dengan interaksi pengguna.

4) Pengujian Sistem

Pada tahap ini penulis melakukan pengujian terhadap kinerja dari masing-masing komponen yang membangun robot pengawasan mengenai penggunaan perangkat *Hardware* dan aplikasi (*Software*).

5) Evaluasi

Pada tahap ini merupakan evaluasi dari tahap pengujian sistem, jika terjadi kendala terkait dengan *Software* ataupun *Hardware* , maka penulis akan memperbaiki dan menganalisa kembali kendala tersebut.

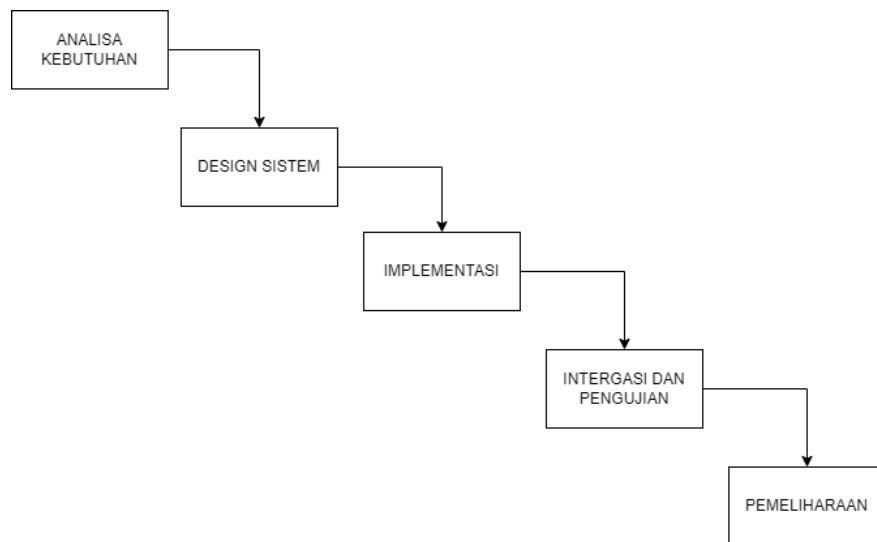
6) Hasil dan Kesimpulan

Pada tahap ini adalah Hasil dan Kesimpulan yang di dapat dari tahapan awal sampai dengan akhir penelitian, yang berguna untuk penelitian selanjutnya.

3.3 Metode Perancangan Sistem

Adapun Metode perancangan sistem digunakan untuk mengambil sebuah keputusan, perencanaan, pengembangan, dan menentukan proses serta data yang diberikan oleh suatu sistem.

Dalam penelitian ini penulis menggunakan metode *waterfall*, Metode air terjun atau yang sering disebut *método waterfall* sering dinamakan siklus hidup klasik (*classic life cycle*), nama model ini sebenarnya adalah “*Linear Sequential Model*” dimana hal ini menggambarkan pendekatan yang sistematis dan juga berurutan (Wahid, 2020).



Gambar 3.2 Metode Waterfall

Adapun Metode *Waterfall* penulis uraikan sebagai berikut :

1. Analisa Kebutuhan

Pada tahap ini penulis menganalisa kebutuhan yang sesuai dengan judul penelitian yaitu Rancang bangun robot pengawasan berbasis *Internet Of Things*, Penulis juga menganalisa penggunaan alat dan bahan.

2. *Design Sistem*

Tahapan pembuatan desain *Hardware, Software*, dan perancangan sistem pada penelitian dan sebelum masuk proses pemograman agar desain yang sudah dibuat jelas dan terstruktur.

3. Implementasi

Pada proses implementasi program serta *Hardware* yang digunakan sesuai dengan kebutuhan dan memastikan beberapa modul terintegrasi dengan benar dan tidak mengalami *Human Error*.

4. Integrasi dan Pengujian

Pada proses ini merupakan penggabungan antara beberapa modul pada tahap sebelumnya, dan dilakukan pengujian sesuai dengan judul penelitian, dan pada saat pengujian berjalan dengan baik sesuai dengan keinginan penulis.

5. Pemeliharaan

Pada tahap ini merupakan pengoperasian sistem secara berkala yang telah selesai dan melakukan sedikit nya revisi perbaikan komponen serta pemograman apabila ditemukan kinerja yang anomali.

3.4 Lokasi Penelitian

Dalam Penelitian ini, penulis menargetkan lokasi penelitian sesuai dengan kriteria perancangan sistem dan pembuatan laporan di Laboratorium Elektronika Institut Teknologi Batam, Fakultas Teknologi Informasi.

3.5 Alat dan Bahan

Berikut ini merupakan alat dan bahan yang digunakan penulis untuk menunjang kebutuhan penelitian dan merealisasikan rancangan sesuai dengan judul penelitian, adapun alat dan bahan terbagi atas 2 bagian yaitu perangkat keras (*Hardware*) dan perangkat lunak(*Software*). Berikut komponen-komponen yang digunakan pada penelitian :

Tabel 3.2 Perangkat Keras

Nama	Spesifikasi
<i>Personal Computer Lenovo T470s</i>	<ul style="list-style-type: none">- Sistem operasi : <i>Windows 11 Pro</i>- <i>Processor: Intel Core 5 Gen 7</i>- <i>RAM : 8 GB</i>- <i>SSD: 256 GB</i>

Lanjutan Tabel 3.2 Perangkat Keras

Nama	Spesifikasi
<i>Handphone Pocophone M4 PRO</i>	<ul style="list-style-type: none"> - Penyimpanan Internal : 128 GB - Chipset : <i>Mediatek Helio G69</i> - RAM : 6 + Ext 2 GB - Sistem Operasi : Android 12
<i>ESP-32 CAM</i>	<ul style="list-style-type: none"> - CPU: <i>Dual-core Xtensa 32-bit LX6</i> dengan clock hingga 240 MHz - Memori: 520 KB SRAM, 4 MB Flash - Sensor: OV2640 - Resolusi: 2 Megapixel (1600x1200 piksel)
<i>Motor DC</i>	<ul style="list-style-type: none"> - Kecepatan Putar : 1000 RPM - Berat : 200 gram - Tegangan : 12 Volt
<i>Driver Motor</i>	<ul style="list-style-type: none"> - Tegangan masukan: 4.5 V - 40 V - Tegangan keluaran: 1.25 V - 37 V - Dimensi: Sekitar 43 mm x 20 mm x 14 mm
<i>Motor Servo SG-90</i>	<ul style="list-style-type: none"> - Tegangan operasi: 4.8 V - 6 V - Kecepatan tanpa beban: 0.12 s/60° - Sudut rotasi: 180° - Tegangan sinyal kontrol : 5 V
Regulator Converter	<ul style="list-style-type: none"> - Tegangan Input: 4.5 - 40V Arus Input: Max 4A - Tegangan Output: 1.25 - 37V Arus Output: Max 4A
Baterai <i>Lithium Ion</i> 18650	<ul style="list-style-type: none"> - Tegangan : 3.7V - Kapasitas : 3000mAh / 3Ah
Kabel Jumper	<ul style="list-style-type: none"> - Tegangan Input: 3 - 32V Arus Input: Max 4A - Tegangan Output: 5 - 35V Arus Output: Max 4A

Tabel 3.3 Perangkat Lunak

Nama Perangkat Lunak	Spesifikasi
Windows 11 Pro	Sistem Operasi <i>Windows</i>
Arduino IDE Versi 1.8.0	<i>Software</i> Teks Editor yang digunakan untuk <i>Sketch Arduino</i>
<i>Google Chrome</i>	<i>Software Web Browser</i>
<i>Visual Studio Code</i>	<i>Software Teks Editor</i>

BAB IV

PERANCANGAN DAN ALAT

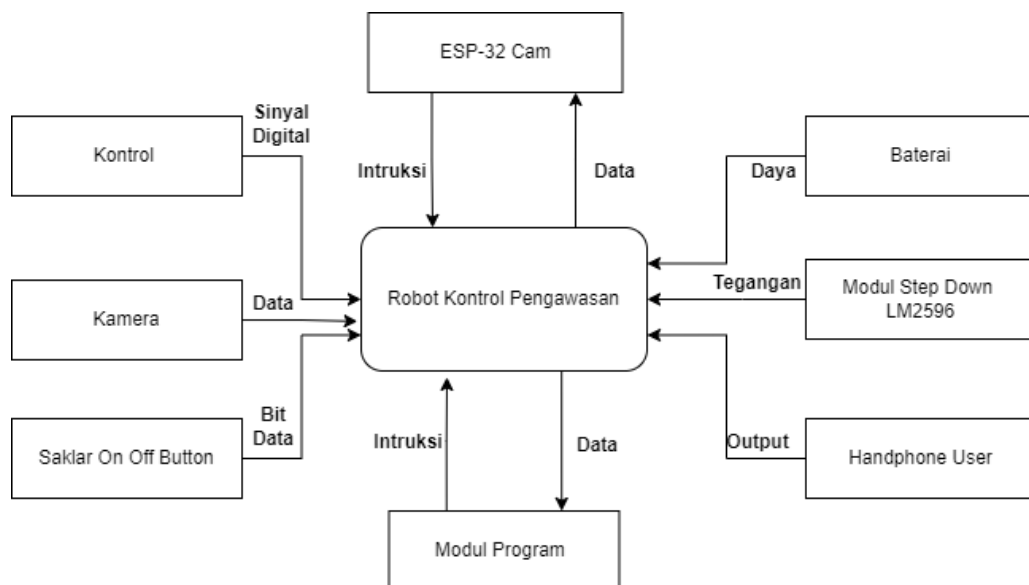
4.1 Analisis Sistem

Meningkatnya Kebutuhan akan gaya hidup dan mendorong banyaknya kejahatan yang terjadi diperumahan tentu saja membuat kewaspadaan semakin meningkat, salah satu modus operasi yang sering digunakan adalah mengelabui keamanan dan melakukan kejahatan yang terekam melalui *CCTV* statis. Hal ini memerlukan adanya solusi untuk sistem keamanan yang lebih baik, dan juga dapat di control dengan menggunakan genggaman tangan.

Pada penelitian ini penulis menerapkan sistem keamanan berjalan dengan menggunakan ESP-32 Cam, Modul Converter, Pant Tilt, Servo, dan Motor DC, pada penelitian ini ESP-32 Cam diprogram agar bisa mengontrol arah pergerakan robot sesuai dengan control yang ada di *Handphone User*/pengguna.

4.1.1 *Contex Diagram*

Contex Diagram merupakan diagram yang menggambarkan system yang akan dirancang secara menyeluruh, Tujuan digunakannya *Contex Diagram* adalah untuk mempermudah dalam memahami sistem secara keseluruhan. Berikut ini merupakan gambar *Contex Diagram* ditunjukkan pada Gambar 4.1.



Gambar 4.1 *Contex Diagram* Sistem Robot Pengawasan

1. Kontrol

Kontrol Merupakan Sinyal yang di input atau dikirimkan agar dapat mengarahakan robot sesuai dengan keinginan *user*.

2. Kamera

Kamera merupakan proses yang dilakukan oleh modul ESP-32 untuk memberikan data berupa Gambar kepada *user*.

3. *Switch On/Off Button*

Berfungsi untuk menghidupkan dan mematikan *system* pada robot pengawas.

4. Modul Program

Tempat penginputan *Source Code* yang berfungsi untuk menjalankan sistem.

5. *ESP-32 Cam*

Modul yang digunakan untuk mengontrol penuh pengendalian sistem dan melakukan proses agar robot dapat berjalan sesuai dengan keinginan *user*.

6. Baterai

Berfungsi sebagai sumber daya atau energi agar sistem dapat digunakan tanpa terhubung ke listrik.

7. Modul *Step Down*

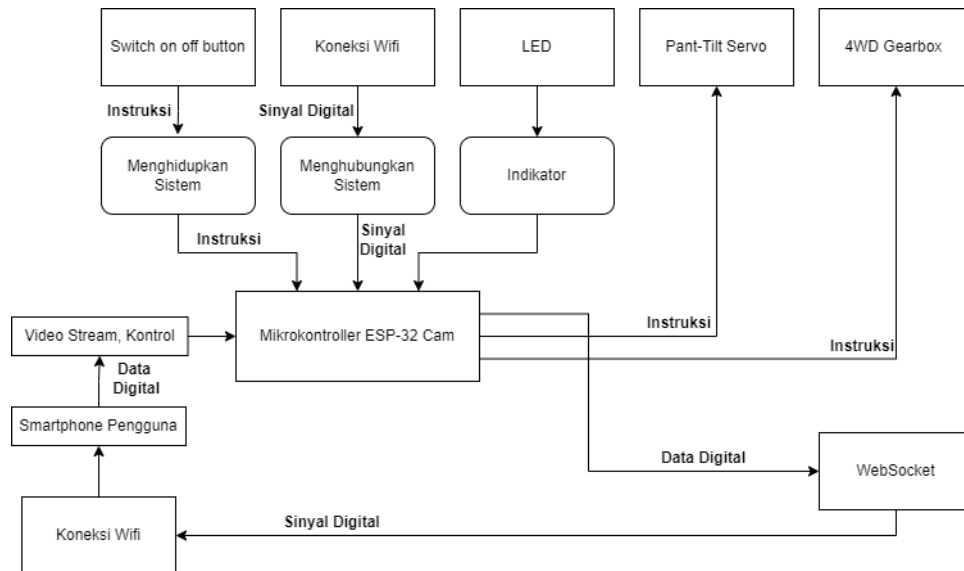
Berfungsi untuk menurunkan tegangan pada sistem agar sistem tidak mengalami galat.

8. *Handphone/Smartphone*

Perangkat digunakan untuk menampilkan output berupa *Websocket* yang berisikan kontrol dan kamera *stream*.

4.1.2 Data Flow Diagram

Data Flow Diagram merupakan penjabaran dari *Context Diagram*. DFD pada sistem dapat dilihat pada gambar 4.2 dibawah ini.



Gambar 4.2 Data Flow Diagram Sistem Robot Pengawas

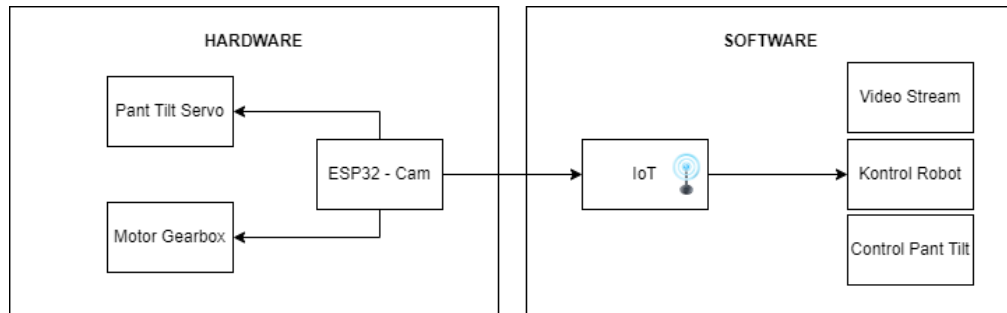
Pada Gambar 4.2 di atas dapat dijelaskan sebagai berikut:

1. Pada data *flow diagram* diatas dapat dilihat pada gambar diatas adalah untuk proses pertama yang dilakukan adalah menghidupkan saklar *on off* pada sistem.
2. Setelah sistem hidup dan sebelum melakukan pergerakan robot, dipastikan koneksi wifi sudah terhubung dengan Handphone user.
3. Untuk menandakan bahwa sistem sudah hidup, penulis meletakan indikator pada power yang ada pada robot.
4. Selanjutnya akan muncul *Web Socket* yang berisi sistem kendali melalui *Browser*.
5. Selanjutnya *Mikrokontroller* melalui user memberikan sinyal kepada robot agar dapat menggerakan Pant-Tilt Servo.
6. Setelah itu *Mikrokontroller* juga memberikan instruksi agar robot dapat berjalan dengan diberikan sinyal dari *Smartphone User* kepada 4 WD *Gearbox*.

7. Selanjutnya *Video Stream* adalah dikirimkan akan tampil dilayar pengguna.

4.1.3 Blok Diagram

Berikut ini merupakan blok diagram pada penelitian ini yang ditunjukkan pada gambar 4.3 dibawah.



Gambar 4.3 Block Diagram Sistem

Dari Blok Diagram Sistem diatas, dapat diuraikan sebagai berikut:

1. Mikrokontroler ESP-32 Cam

ESP32 Cam akan memproses atau sebagai *controller* pada sistem menjadikan logika pemograman untuk memberikan *command processing*.

2. Pant Tilt Servo

Diberikan instruksi kepada Servo agar dapat bergerak sesuai dengan Instruksi *User*.

3. Motor Gearbox

Diberikan instruksi kepada Motor Gearbox agar dapat bergerak sesuai dengan Instruksi *User* dan Motor Driver memberikan arah sesuai dengan instruksi *controller*.

4. Internet Of Things(IoT)

Internet Of Things adalah teknologi yang digunakan untuk menghubungkan server dan robot agar dapat di jalankan secara jarak jauh / dalam jaringan nirkabel.

5. Kontrol Robot

Berfungsi untuk mengontrol jalan robot dan pergerakan pant-tilt sesuai dengan instruksi *user*.

4.2 Perancangan Sistem

Analisa sistem diatas penulis melakukan dua tahapan perancangan sistem yang pertama adalah perancangan perangkat keras(*Hardware*) meliputi rangkaian yang menggunakan komponen-komponen elektronik yang digunakan dan perancangan perangkat lunak (*Software*) meliputi pembuatan program Kontrol menggunakan *Web based*.

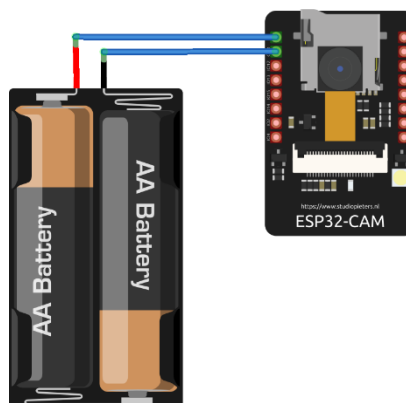
4.2.1 Perancangan Perangkat Keras (*Hardware*)

Dalam Perancangan *Hardware*, dilakukan beberapa proses diantaranya perancangan masing-masing rangkaian, komponen, dan pengkabelan. Perangkat keras yang digunakan pada penelitian ini adalah *ESP32-Cam* sebagai *Microcontroller*, Modul Driver sebagai penggerak, Motor *DC*, dan Konverter Arus 12V ke 5V.

4.2.2 Perancangan *Hardware* pada ESP32cam

Komponen Baterai atau sebagai sumber tegangan untuk mengaktifkan komponen lainnya. Dalam hal ini menggunakan *12V* namun menggunakan Konverter sebagai penurun tegangan sehingga komponen *ESP32Cam* dapat digunakan. Dapat dilihat pada Gambar 4.4.

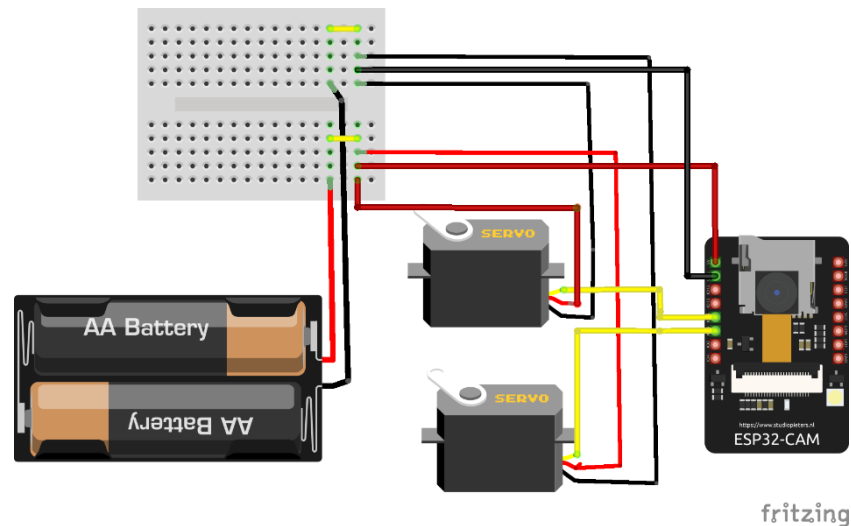
Berikut merupakan uraian gambar diatas pin dari *ESP32-Cam* yang digunakan untuk terhubung pada pin power *VCC(5 Volt)* dan *GROUND(GND)* dari *ESP32-Cam*.



Gambar 4.4 Schematic ESP-32 Cam

4.2.3 Perancangan Pant Tilt Servo

Pada Gambar 4.5 dibawah ini merupakan *Shcematic* yang meliputi *ESP-32-Cam* yang pin 12 dan 13 nya dihubungkan ke pin data yang ada pada 2 servo sebagai penggerak ke atas – bawah dan kiri-kanan. Dengan menggunakan input power sebesar 5*Volt* untuk mengalir ke komponen *ESP32-Cam* dan 2 Servo.

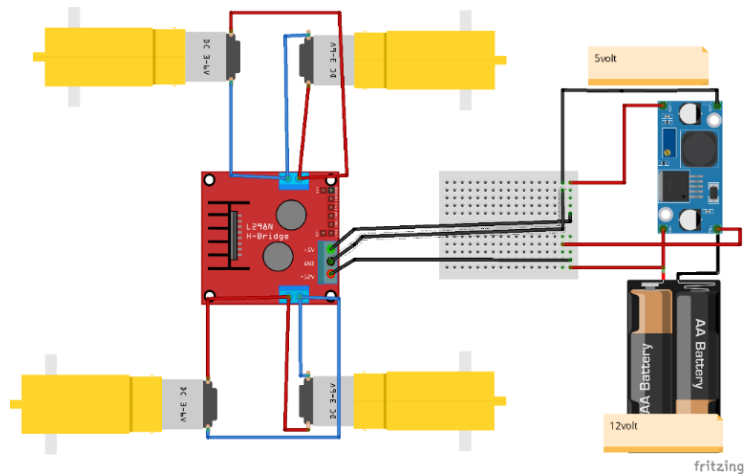


Gambar 4.5 Schematic Pant Tilt

4.2.4 Perancangan Motor DC Gearbox

Pada gambar 4.6 dibawah ini merupakan Perancangan Komponen Motor DC Gearbox yang digunakan agar robot dapat berjalan dan menyesuaikan arah.

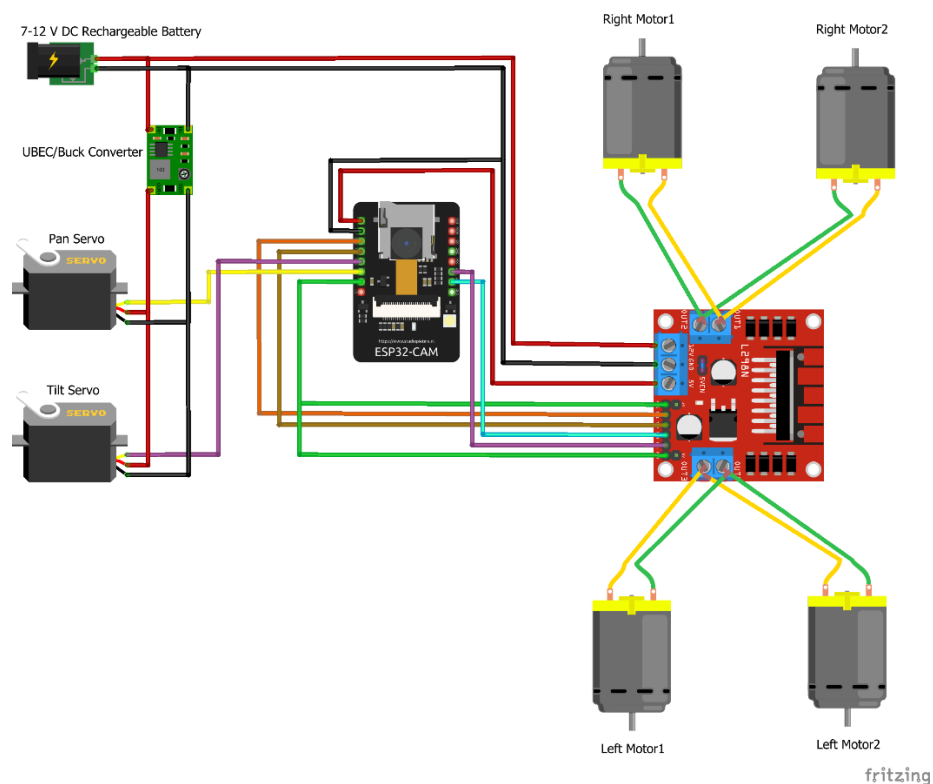
Diatas merupakan gambar *Schematic* Motor DC, pada gambar diatas masuk input sebesar 12 *Volt* yang di convert menjadi 5 *Volt* yang berguna untuk menghidupkan modul driver dan memberikan sinyal bahwa Motor driver dalam Kondisi Hidup.



Gambar 4.6 Schematic Motor DC Gearbox

4.2.5 Perancangan Keseluruhan *Hardware*

Untuk perancangan keseluruhan *Hardware* bertujuan untuk mengetahui hasil daripada penelitian yang telah dibangun. Perancangan ini dibangun melalui *Schematic* untuk mengetahui gambaran tentang bagaimana suatu penelitian dapat dicapai. Dapat dilihat pada gambar 4.7 dibawah merupakan *Schematic* Perancangan *Hardware*.

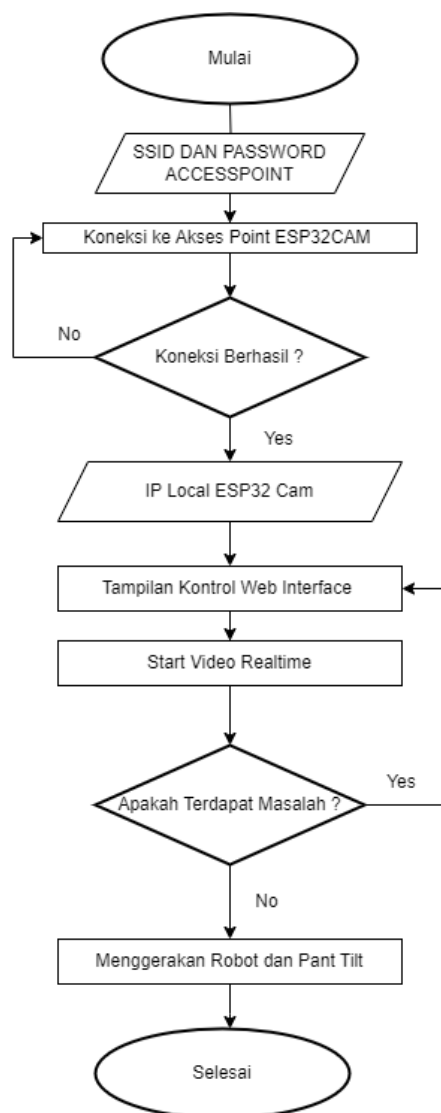


Gambar 4.7 Perancangan *Schematic* Keseluruhan

Perancangan *Schematic* tersebut menggunakan aplikasi *Fritzing* untuk desain dan membuat simulasi sebelum mengimplementasikan ke hasil penelitian seperti Modul ESP32-Cam, Motor *Servo*, Motor *DC*, Motor *Driver* dan Catu daya. Sehingga penulis dapat menggambarkan simulasi ke *realcase* yang ingin dibangun.

4.3 Perancangan Perangkat Lunak (*Software*)

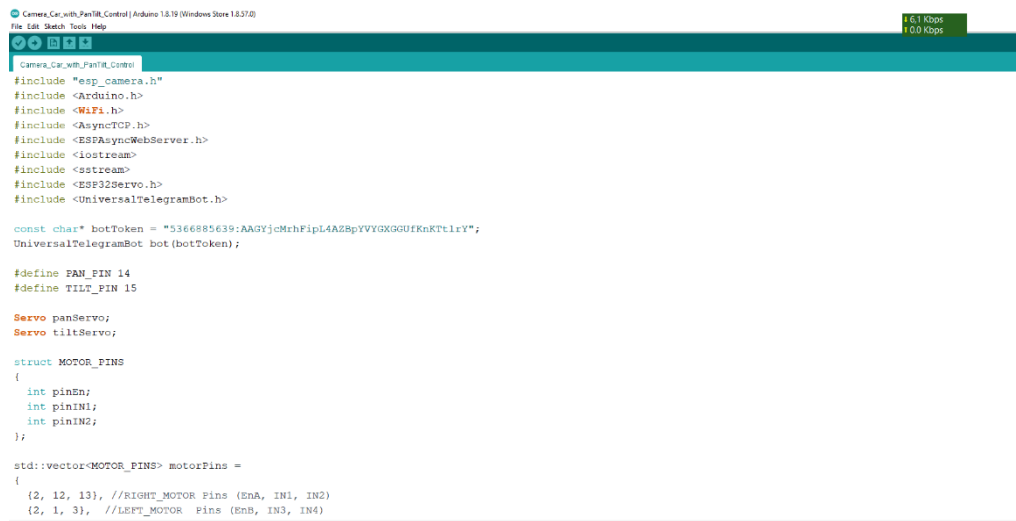
Setelah proses perancangan perangkat keras(*Hardware*) selesai, langkah selanjutnya adalah perancangan perangkat lunak yang bertujuan untuk membuat sebuah pengaturan sistem akan dibangun. Program ini akan di *upload* ke dalam modul *ESP32-Cam* sebagai *Mikrokontroler*. Berikut ini tampilan *Flowchart* Sistem Kerja yang akan dirancang oleh penulis pada Gambar 4.8.



Gambar 4.8 Flowchart Keluruhan Sistem

4.3.1 Implementasi *Arduino IDE*

Pada tahapan implementasi suatu tahapan dalam perancangan terhadap perangkat lunak yang akan digunakan dalam hal ini adalah menggunakan *Local Web Based*. Tahapan implementasi yang digunakan setelah proses Analisa dan perancangan telah selesai dilakukan. selanjutnya implemnetasi dihubungkan secara langsung dengan perangkat yang digunakan dalam penelitian dalam hal ini menggunakan Modul *ESP-32Cam*. Pada gambar 4.7 disajikan contoh keseluruhan Program yang digunakan dalam penelitian ini.



```
Camera_Car_with_PanTilt_Control (Arduino 1.8.19 (Windows Store 1.8.17.0))
File Edit Sketch Tools Help
Camera_Car_with_PanTilt_Control
1.61 Kbps
1.00 Kbps

Camera_Car_with_PanTilt_Control
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <iostream>
#include <sstream>
#include <ESP32Servo.h>
#include <UniversalTelegramBot.h>

const char* botToken = "5366885639:AAGYjcMrhFipL4AE8pYVYGXGGUfKnRTlrY";
UniversalTelegramBot bot(botToken);

#define PAN_PIN 14
#define TILT_PIN 15

Servo panServo;
Servo tiltServo;

struct MOTOR_PINS
{
    int pinEn;
    int pinIN1;
    int pinIN2;
};

std::vector<MOTOR_PINS> motorPins =
{
    {2, 12, 13}, //RIGHT_MOTOR Pins (ENa, IN1, IN2)
    {2, 1, 3}, //LEFT_MOTOR Pins (ENb, IN3, IN4)
}
```

Gambar 4.9 Contoh Keseluruhan program

Pada gambar diatas adalah contoh keseluruhan program yang dapat diuraikan sebagai berikut. Adapun tahapan yang pertama adalah memasukan *Library ESP32-Cam* fungsinya untuk mengaktifkan dan menghubungkan *ESP32-Cam* Modul ke *Wifi*. Proses kedua adalah memasukan kodingan *esp camera* yang berfungsi untuk mengkonfigurasi *pin board* agar bisa digunakan. Selain itu ada beberapa *library* pendukung, dan diprogram didalam *Arduino IDE* untuk menampilkan kontrol yang ada pada *Web Browser*.

4.3.2 Implementasi Kontrol Robot berbasis *Websockets*

WebSockets merupakan salah satu teknologi yang digunakan untuk mengkoneksikan antara klien dan server dalam hal ini browser dan server yang memungkinkan untuk komunikasi secara *real-time*. Dapat dilihat pada gambar 4.8 Kontrol Web yang ada pada *Handphone User*.



Gambar 4.10 Kontrol Web User

Untuk memuat *Websocket* digunakan *Source Code* seperti dibawah ini.

```
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-scale=1, maximum-scale=1,
user-scalable=no">
    <style>
      .arrows {
        font-size:30px;
        color:red;
      }
      td.button {
        background-color:black;
```

```

border-radius:25%;
box-shadow: 5px 5px #888888;
}
td.button:active {
transform: translate(5px,5px);
box-shadow: none;
}

.noselect {
-Webkit-touch-callout: none; /* iOS Safari */
-Webkit-user-select: none; /* Safari */
-khtml-user-select: none; /* Konqueror HTML */
-moz-user-select: none; /* Firefox */
-ms-user-select: none; /* Internet Explorer/Edge */
user-select: none; /* Non-prefixed version, currently
supported by Chrome and Opera */
}

.slidecontainer {
width: 100%;
}

.slider {
-Webkit-appearance: none;
width: 100%;
height: 15px;
border-radius: 5px;
background: #d3d3d3;
outline: none;
opacity: 0.7;
-Webkit-transition: .2s;
transition: opacity .2s;
}

.slider:hover {
opacity: 1;
}

```

```

.slider::-Webkit-slider-thumb {
  -Webkit-appearance: none;
  appearance: none;
  width: 25px;
  height: 25px;
  border-radius: 50%;
  background: red;
  cursor: pointer;
}

.slider::-moz-range-thumb {
  width: 25px;
  height: 25px;
  border-radius: 50%;
  background: red;
  cursor: pointer;
}

</style>

</head>
<body class="noselect" align="center" style="background-color:white">
  <table id="mainTable" style="width:400px;margin:auto;table-layout:fixed"
  CELSPACING=10>
    <tr>
      <img id="cameraImage" src="" style="width:400px;height:250px"></td>
    </tr>
    <tr>
      <td></td>
      <td class="button" ontouchstart='sendButtonInput("MoveCar","1")'
      ontouchend='sendButtonInput("MoveCar","0")><span class="arrows" >#8679;</span></td>
      <td></td>
    </tr>
    <tr>
      <td class="button" ontouchstart='sendButtonInput("MoveCar","3")'
      ontouchend='sendButtonInput("MoveCar","0")><span class="arrows" >#8678;</span></td>
      <td class="button"></td>
    </tr>

```

```

        <td class="button" ontouchstart='sendButtonInput("MoveCar","4")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8680;</span></td>
    </tr>
    <tr>
        <td></td>
        <td class="button" ontouchstart='sendButtonInput("MoveCar","2")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows" >&#8681;</span></td>
    </tr>
</tr><tr/>
<tr>
    <td style="text-align:left"><b>Speed:</b></td>
    <td colspan=2>
        <div class="slidecontainer">
            <input type="range" min="0" max="255" value="150" class="slider" id="Speed"
oninput='sendButtonInput("Speed",value)'>
        </div>
    </td>
</tr>
<tr>
    <td style="text-align:left"><b>Light:</b></td>
    <td colspan=2>
        <div class="slidecontainer">
            <input type="range" min="0" max="255" value="0" class="slider" id="Light"
oninput='sendButtonInput("Light",value)'>
        </div>
    </td>
</tr>
<tr>
    <td style="text-align:left"><b>Pan:</b></td>
    <td colspan=2>
        <div class="slidecontainer">
            <input type="range" min="0" max="180" value="90" class="slider" id="Pan"
oninput='sendButtonInput("Pan",value)'>
        </div>
    </td>
</tr>
<tr>

```

```
<td style="text-align:left"><b>Tilt:</b></td>
<td colspan=2>
  <div class="slidecontainer">
    <input type="range" min="0" max="180" value="90" class="slider" id="Tilt"
oninput='sendButtonInput("Tilt",value)'>
  </div>
</td>
</tr>
</table>
```

BAB V

HASIL DAN PEMBAHASAN

5.1 Implementasi

Setelah proses perancangan Sistem baik perangkat keras maupun perangkat lunak yang sudah di selesaikan. Maka dilanjutkan ketahap implementasi sistem. Proses ini meliputi rangkaian perakitan mekanikal terkait robot pengawasan, pemasangan *wiring*, serta pemograman *mikrokontroller*. Sistem yang telah terbentuk dan dapat di integrasikan satu sama lain akan dilanjutkan ke tahap pengujian. Adapun beberapa tahapan pengujian yaitu pengujian perkiraan jarak, pengujian Integrasi Jaringan, dan Pengujian Sistem Kontrol.

5.2 Hasil Pengujian

Tahapan pengujian meliputi beberapa pengujian yaitu pengujian perkiraan jarak tempuh, pengujian integrasi jaringan dan pengujian sistem kontrol.

5.2.1 Pengujian Jarak

Pengujian Jarak dilakukan untuk mengetahui berapa jauh jarak dan tingkat keberhasilan dari sistem yang sudah di jalankan. Adapun peran yang penting dalam pengujian jarak adalah untuk mengetahui berapa jarak tempuh yang bisa dilakukan oleh Robot Pengawasan. dan indikator utama dalam Pengujian jarak adalah beberapa halang rintang seperti Tembok/ Penghalang, Objek, dan lain-lain yang membuat robot tidak dapat berjalan dengan baik.

5.2.1.1 Langkah Pengujian

Adapun langkah-langkah dalam pengujian robot pengawasan sebagai berikut:




1. Siapkan Robot Pengawasan yang akan diuji.
2. Lakukan *Pairing* ke Handphone.
3. Buka *IP Address* untuk terkoneksi ke *Websocket*.
4. Lakukan Pengujian terhadap jarak yang ditempuh oleh robot pengawasan

5. Simpulkan Kemampuan robot berdasarkan hasil dari tabel pengujian.

5.2.1.2 Hasil Pengujian Jarak Tempuh

Data hasil pengujian Jarak tempuh menggunakan aplikasi *inssider* pada robot pengawasan didalam ruangan dengan dapat dilihat pada Tabel 5.1 dibawah ini :

Tabel 5.1 Hasil Pengujian Jarak Tempuh

No	Jarak	Gambar	Berhasil/Gagal
1.	8m		Berhasil
2.	12m		Berhasil
3	16m		Berhasil

Dari beberapa pengujian diatas merupakan beberapa pengujian dibeberapa *Range* yang diukur dengan satuan Meter, terlihat bahwa gambar diatas, pada saat pengujian robot dan dijalankan, memiliki beberapa jangkauan yang cukup jauh.

5.2.2 Pengujian Integrasi Jaringan

Proses Pengujian Integrasi Jaringan berguna untuk mengetahui seberapa besar kekuatan sinyal dari jarak yang ditempuh robot pengawas, dan indikator keberhasilan robot dilihat dari kekuatan sinyal yang dapat terkoneksi dengan stabil sehingga robot dapat dijalankan sesuai dengan instruksi yang diberikan, Robot Pengawas tentu memiliki jarak ideal agar dapat di komunikasikan dengan baik antar *smartphone* yang digunakan dan kontrol robot yang dijalankan.

5.2.2.1 Langkah Pengujian Integrasi Jaringan


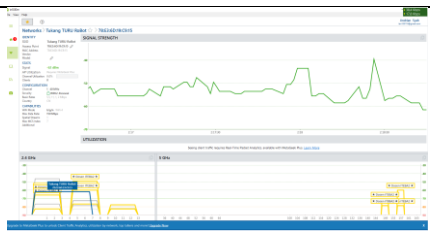
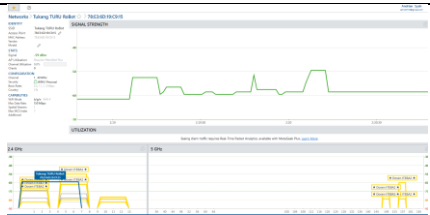
Adapun langkah-langkah pengujian integrasi jaringan adalah sebagai berikut :

1. Siapkan Robot Pengawas yang akan diuji.
2. *Pairing* Handphone dan Wifi yang digunakan pada Robot Pengawas.
3. Siapkan *Software Inssider* yang berfungsi untuk melihat berapa kekuatan sinyal Robot Pengawas.
4. Buka dan Lihat Frekuensi yang digunakan pada Robot pengawas.
5. Lakukan Kesimpulan dalam tabel pengujian integrasi jaringan.

5.2.2.2 Hasil Pengujian Integrasi Jaringan

Berikut ini merupakan Tabel 5.2 Hasil dari pengujian integrasi jaringan menggunakan aplikasi *inssider* , dengan 3 fase percobaan dengan jarak 8m, 12m, dan 16m.

Tabel 5.2 Pengujian Integrasi Jaringan

No	Kekuatan Sinyal	Jarak	Gambar	Berhasil/Gagal /Delay
1.	-56 dbm	8m		Berhasil
2.	-62 dbm	12m		Berhasil
3.	-70 dbm	16m		Delay

Hasil dari Analisa diatas adalah didapatkan dari kekuatan sinyal yang ada pada *Wifi* Robot Tukang Turu yang dimana hasil yang didapatkan adalah semakin jauh jarak yang ditempuh maka kekuatan sinyal yang dipancarkan akan semakin berkurang.

5.2.3 Pengujian Sistem Kontrol

Pengujian Sistem Kontrol dilakukan untuk menguji Hasil dari Pengontrolan robot yang ada pada *Smartphone User*. Dengan beberapa indikator keberhasilan yaitu menampilkan hasil objek yang di pantau melalui *Microcontroller ESP32-Cam*.

5.2.3.1 Langkah Pengujian Sistem Kontrol

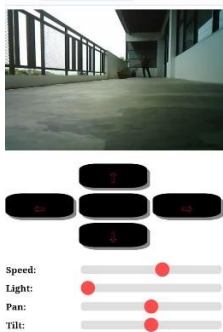
Berikut ini adalah langkah-langkah pengujian Sistem Kontrol yang ada pada robot pengawasan :

1. Siapkan Robot Pengawas yang akan diuji.
2. *Pairing* Handphone dan Wifi yang digunakan pada Robot Pengawas.
3. Masukkan *IP Address* untuk mengakses kontrol robot.
4. Lakukan Pengujian Kontrol Robot dengan beberapa objek.
5. Lakukan Pengujian untuk menggerakkan *Servo Pant-Tilt* sesuai dengan derajat ketentuan *servo*.
6. Buat Hasil Kesimpulan untuk uji sistem kontrol pada robot pengawas.


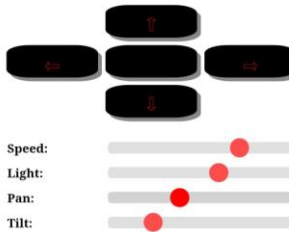

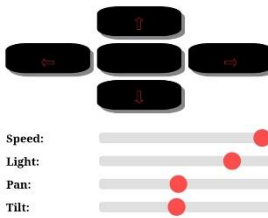
5.2.3.2 Hasil Pengujian Sistem Kontrol

Berikut Merupakan Hasil dari pengujian sistem kontrol yang digunakan pada saat sistem di jalankan.

Tabel 5.3 Tabel Uji Sistem Kontrol

No	Objek	Gambar	Berhasil/Gagal /Delay
1.	Andrian Syah		Berhasil

Lanjutan Tabel 5.4 Tabel Uji Sistem Kontrol

2.	Siswa SMA Negeri 4 Batam	 	Berhasil
3.	Maranti	 	Delay

Dari Hasil uji diatas dari beberapa objek beberapa indikator dinyatakan berhasil karena robot dapat berjalan dengan baik dan untuk 1 indikator yang delay adalah kamera, karena pada saat pengujian alat sedang tidak berjalan dengan sehingga menyebabkan hasil yang didapatkan delay.

BAB VI

KESIMPULAN DAN SARAN

6.1 Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan yang berjudul rancang bangun robot *mobile* pengawasan berbasis *internet of things* menggunakan Kamera ESP 32, dapat disimpulkan sebagai berikut:

1. Perancangan dan Pembuatan Robot Pengawasan dengan Menggunakan *Mikrokontroller* Kamera ESP-32 dapat di implementasikan di kehidupan sehari-hari. dengan adanya penambahan kontrol mobile maka robot dapat berjalan sesuai dengan instruksi pengguna/*user*.
2. Robot dapat dikendalikan menggunakan *Websocket* yang terhubung melalui *Wi-Fi* dari robot ke *Handphone* pengguna tanpa menggunakan kabel dan kontrol yang digunakan memudahkan pengguna di dalam menggerakkan bagian-bagian robot seperti *Motor DC* dan *Pant Tilt Servo*.

6.2 Saran

Adapun beberapa saran untuk penelitian yang telah dibuat dengan harapan penelitian ini dapat terus berkembang, maka penulis menyarankan sebagai berikut :

1. Diharapkan pada pengembang selanjutnya agar dapat menambahkan beberapa komponen pada robot pengawasan seperti sensor tracking dan *GPS* agar robot dapat di track/dipantau melalui aplikasi yang bertujuan untuk memberikan informasi kepada pengguna.
2. Diharapkan penelitian ini dibuat dan dikembangkan oleh mahasiswa Institut Teknologi Batam bagi setiap Angkatan yang mengambil tugas akhir.

DAFTAR PUSTAKA

- Adikara, F., Ramadhan, M. I., & Mulia, U. B. (2021). *PENYULUHAN MENGENAI IoT FOR SMART BUILDING (STUDI KASUS KAMPUS GADING SERPONG UEU)*. 7.
- Agustina, S. (2023). *PROTOTYPE SISTEM KEAMANAN DAN AUTOMASI RUMAH PINTAR BERBASIS INTERNET OF THINGS (IOT)*. 3.
- Alsumady, M. O., Alturk, Y. K., Dagamseh, A., & Tantawi, M. (2021). Controlling of DC-DC Buck Converters Using Microcontrollers. *International Journal of Circuits, Systems and Signal Processing*, 15, 197–202. <https://doi.org/10.46300/9106.2021.15.22>
- Arby, W., Hendrik, B., & Awal, H. (2022). *PERANCANGAN DAN IMPLEMENTASI ROBOT KESEIMBANGAN BERODA DUA BERBASIS MIKROKONTROLER*. 2(1).
- Caniago, D. P. (2022). The Internet of Things application on Student Assignee Smart Box using ESP32-Cam. *Jurnal CoSciTech (Computer Science and Information Technology)*, 3(3), 479–486. <https://doi.org/10.37859/coscitech.v3i3.4347>
- Eulaerts, O., & Joanny, G. (2022). *Weak signals in border management and surveillance technologies*. <https://doi.org/10.2760/784388>
- H, K., Subrata, R., H., & Gozali, F. (2019). Sistem Keamanan Ruang Berbasis Internet Of Things Dengan Menggunakan Aplikasi Android. *TESLA: Jurnal Teknik Elektro*, 20(2), 127. <https://doi.org/10.24912/tesla.v20i2.2989>
- Hergika, G., Siswanto, & S, S. (2021). PERANCANGAN INTERNET OF THINGS (IOT) SEBAGAI KONTROL INFRASTRUKTUR DAN PERALATAN TOLL PADA PT. ASTRA INFRATOLL ROAD. *PROSISKO: Jurnal Pengembangan Riset dan Observasi Sistem Komputer*, 8(2), 86–98. <https://doi.org/10.30656/prosisko.v8i2.3862>
- Hidayanto, F. (2015). *MEMANFAATKAN PERKEMBANGAN TEKNOLOGI INTERNET DAN SITUS WEB UNTUK KEPENTINGAN WARGA SEKALIGUS SEBAGAI SARANA PROMOSI POTENSI DESA GERBOSARI*. 4(1).

- Hidayat, R., Marlina, S., & Utami, L. D. (2017). *Perancangan Sistem Informasi Penjualan Barang Handmade Berbasis Website Dengan Metode Waterfall*.
- Ilham, F. C., & Sulasmoro, A. H. (n.d.). *RANCANG BANGUN SISTEM BEL OTOMATIS PADA KANTOR MENGGUNAKAN SENSOR SUHU TUBUH BERBASIS ARDUINO*.
- Isrofi, A., Utama, S. N., & Putra, O. V. (2021). RANCANG BANGUN ROBOT PEMOTONG RUMPUT OTOMATIS MENGGUNAKAN WIRELESS KONTROLER MODUL ESP32-CAM BERBASIS INTERNET of THINGS (IoT). *Jurnal Teknoinfo*, 15(1), 45. <https://doi.org/10.33365/jti.v15i1.675>
- Latifa, U., & Slamet Saputro, J. (2018). PERANCANGAN ROBOT ARM GRIPPER BERBASIS ARDUINO UNO MENGGUNAKAN ANTARMUKA LABVIEW. *Barometer*, 3(2), 138–141. <https://doi.org/10.35261/barometer.v3i2.1395>
- Nurmalina, R. (n.d.). *Perencanaan dan Pengembangan Aplikasi Absensi Mahasiswa Menggunakan Smart Card Guna Pengembangan Kampus Cerdas (Studi Kasus Politeknik Negeri Tanah Laut)*.
- Parihar, Y. S. (2019). *Internet of Things and Nodemcu*. 6(6).
- Peerzada, P., Larik, W. H., & Mahar, A. A. (2021). DC Motor Speed Control Through Arduino and L298N Motor Driver Using PID Controller. *International Journal of Electrical Engineering*, 04(02).
- Rusimamto, P. W., Endryansyah, E., Anifah, L., Harimurti, R., & Anistyasari, Y. (2021). Implementation of arduino pro mini and ESP32 cam for temperature monitoring on automatic thermogun IoT-based. *Indonesian Journal of Electrical Engineering and Computer Science*, 23(3), 1366. <https://doi.org/10.11591/ijeecs.v23.i3.pp1366-1375>
- Salim, A. I., Saragih, Y., & Hidayat, R. (2020). Implementasi Motor Servo SG 90 Sebagai Penggerak Mekanik Pada E. I. Helper (ELECTRONICS INTEGRATION HELMET WIPER). *Electro Luceat*, 6(2), 236–244. <https://doi.org/10.32531/jelekn.v6i2.256>

- Setiawan, A. (2022, Desember). *Kriminalitas di Tanjungpinang meningkat selama 2022*. <https://www.antaranews.com/berita/3330480/kriminalitas-di-tanjungpinang-meningkat-selama-2022>
- Setiawan, F. B., Wibowo, Y. Y. C., Pratomo, L. H., & Riyadi, S. (2022). Perancangan Automated Guided Vehicle Menggunakan Penggerak Motor DC dan Motor Servo Berbasis Raspberry Pi 4. *Jurnal Rekayasa Elektrika*, 18(2). <https://doi.org/10.17529/jre.v18i2.25863>
- Ulinnuha, M. (2020). *Pengertian dan Sejarah Robotika*. 8 September. <https://mitech-ndt.co.id/pengertian-dan-sejarah-robotika/>
- Wahid, A. A. (2020). *Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*.
- Wardoyo, P., & Liana Putri, I. (2023). Sistem Informasi Akademik E-Rapor Pada SMAN 2 KS Cilegon. *Jurnal Insan Unggul*, 11(2), 145–176. <https://doi.org/10.47926/jiu.2023.11.2.145-176>
- Zhang, C., Wang, Q., Zhan, Q., He, T., & An, Y. (2019). Autonomous system design for dam surveillance robots. *Proceedings - 2019 IEEE SmartWorld, Ubiquitous Intelligence and Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Internet of People and Smart City Innovation, SmartWorld/UIC/ATC/SCALCOM/IOP/SCI 2019*, 158–161. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00069>
- Efendi, Y. (2018). INTERNET OF THINGS (IOT) SISTEM PENGENDALIAN LAMPU MENGGUNAKAN RASPBERRY PI BERBASIS MOBILE. *Jurnal Ilmiah Ilmu Komputer*, 4(1). <http://ejournal.fikom-unasman.ac.id>
- Sanfreni, Adikara, F., & Ikshan Ramadhan, M. (2021). PENYULUHAN MENGENAI IoT FOR SMART BUILDING (STUDI KASUS KAMPUS GADING SERPONG UEU).
- Kristomson, H., Subrata, R. H., & Gozali, F. (2018). Sistem Keamanan Ruangan Berbasis Internet Of Things Dengan Menggunakan Aplikasi Android.
- Roma Doni, F. (2020). Akses Kamera CCTV Dari Jarak Jauh Untuk Monitoring Keamanan Dengan Penerapan PSS. *Jurnal Sains Dan Manajemen*, 8(1).

LAMPIRAN SOURCE CODE

```
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <iostream>
#include <sstream>
#include <ESP32Servo.h>

#define PAN_PIN 14
#define TILT_PIN 15

Servo panServo;
Servo tiltServo;

struct MOTOR_PINS
{
    int pinEn;
    int pinIN1;
    int pinIN2;
};

std::vector<MOTOR_PINS> motorPins =
{
    {2, 12, 13}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
    {2, 1, 3},  //LEFT_MOTOR Pins (EnB, IN3, IN4)
};

#define LIGHT_PIN 4

#define UP 1
#define DOWN 2
#define LEFT 3
#define RIGHT 4
#define STOP 0

#define RIGHT_MOTOR 0
#define LEFT_MOTOR 1

#define FORWARD 1
#define BACKWARD -1

const int PWMFreq = 1000; /* 1 KHz */
const int PWMResolution = 8;
const int PWMSpeedChannel = 2;
const int PWMLightChannel = 3;
```

```

//Camera related constants
#define PWDN_GPIO_NUM    32
#define RESET_GPIO_NUM   -1
#define XCLK_GPIO_NUM     0
#define SIOD_GPIO_NUM    26
#define SIOC_GPIO_NUM    27
#define Y9_GPIO_NUM      35
#define Y8_GPIO_NUM      34
#define Y7_GPIO_NUM      39
#define Y6_GPIO_NUM      36
#define Y5_GPIO_NUM      21
#define Y4_GPIO_NUM      19
#define Y3_GPIO_NUM      18
#define Y2_GPIO_NUM       5
#define VSYNC_GPIO_NUM    25
#define HREF_GPIO_NUM     23
#define PCLK_GPIO_NUM     22

const char* ssid        = "Tukang Turu Robot";
const char* password    = "12345678";

AsyncWebServer server(80);
AsyncWebsocket wsCamera("/Camera");
AsyncWebsocket wsCarInput("/CarInput");
uint32_t cameraClientId = 0;

const char* htmlHomePage PROGMEM = R"HTMLHOMEPAGE(
<!DOCTYPE html>
<html>
  <head>
    <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1, user-scalable=no">
    <style>
      .arrows {
        font-size:30px;
        color:red;
      }
      td.button {
        background-color:black;
        border-radius:25%;
        box-shadow: 5px 5px #888888;
      }
      td.button:active {
        transform: translate(5px,5px);
        box-shadow: none;
      }

```

```

.noselect {
    -Webkit-touch-callout: none; /* iOS Safari */
    -Webkit-user-select: none; /* Safari */
    -khtml-user-select: none; /* Konqueror HTML */
    -moz-user-select: none; /* Firefox */
    -ms-user-select: none; /* Internet Explorer/Edge */
    user-select: none; /* Non-prefixed version,
currently
                                supported by Chrome and Opera
*/
}

.slidecontainer {
    width: 100%;
}

.slider {
    -Webkit-appearance: none;
    width: 100%;
    height: 15px;
    border-radius: 5px;
    background: #d3d3d3;
    outline: none;
    opacity: 0.7;
    -Webkit-transition: .2s;
    transition: opacity .2s;
}

.slider:hover {
    opacity: 1;
}

.slider::-webkit-slider-thumb {
    -Webkit-appearance: none;
    appearance: none;
    width: 25px;
    height: 25px;
    border-radius: 50%;
    background: red;
    cursor: pointer;
}

.slider::-moz-range-thumb {
    width: 25px;
    height: 25px;
    border-radius: 50%;

```

```

        background: red;
        cursor: pointer;
    }

</style>

</head>
<body class="noselect" align="center" style="background-
color:white">
    <table id="mainTable" style="width:400px;margin:auto;table-
layout:fixed" CELLSPACING=10>
        <tr>
            <img id="cameraImage" src=""
style="width:400px;height:250px"></td>
        </tr>
        <tr>
            <td></td>
            <td class="button"
ontouchstart='sendButtonInput("MoveCar","1")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8679;</span></td>
            <td></td>
        </tr>
        <tr>
            <td class="button"
ontouchstart='sendButtonInput("MoveCar","3")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8678;</span></td>
            <td class="button"></td>
            <td class="button"
ontouchstart='sendButtonInput("MoveCar","4")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8680;</span></td>
        </tr>
        <tr>
            <td></td>
            <td class="button"
ontouchstart='sendButtonInput("MoveCar","2")'
ontouchend='sendButtonInput("MoveCar","0")'><span class="arrows"
>#8681;</span></td>
            <td></td>
        </tr>
        <tr/><tr/>
        <tr>
            <td style="text-align:left"><b>Speed:</b></td>
            <td colspan=2>
                <div class="slidecontainer">

```

```

        <input type="range" min="0" max="255" value="150"
class="slider" id="Speed" oninput='sendButtonInput("Speed",value)'+>
    </div>
</td>
</tr>
<tr>
    <td style="text-align:left"><b>Light:</b></td>
    <td colspan=2>
        <div class="slidecontainer">
            <input type="range" min="0" max="255" value="0"
class="slider" id="Light" oninput='sendButtonInput("Light",value)'+>
        </div>
    </td>
</tr>
<tr>
    <td style="text-align:left"><b>Pan:</b></td>
    <td colspan=2>
        <div class="slidecontainer">
            <input type="range" min="0" max="180" value="90"
class="slider" id="Pan" oninput='sendButtonInput("Pan",value)'+>
        </div>
    </td>
</tr>
<tr>
    <td style="text-align:left"><b>Tilt:</b></td>
    <td colspan=2>
        <div class="slidecontainer">
            <input type="range" min="0" max="180" value="90"
class="slider" id="Tilt" oninput='sendButtonInput("Tilt",value)'+>
        </div>
    </td>
</tr>
</table>

```

```

<script>
    var WebSocketCameraUrl = "ws:///" + window.location.hostname
+ "/Camera";
    var WebSocketCarInputUrl = "ws:///" +
window.location.hostname + "/CarInput";
    var WebSocketCamera;
    var WebSocketCarInput;

    function initCameraWebSocket()
    {
        WebSocketCamera = new WebSocket(WebSocketCameraUrl);
        WebSocketCamera.binaryType = 'blob';
        WebSocketCamera.onopen    = function(event){};
    }

```

```

        WebsocketCamera.onclose    =
function(event){setTimeout(initCameraWebSocket, 2000)};
        WebsocketCamera.onmessage = function(event)
        {
            var imageId = document.getElementById("cameraImage");
            imageId.src = URL.createObjectURL(event.data);
        };
    }

function initCarInputWebSocket()
{
    WebsocketCarInput = new WebSocket(WebSocketCarInputUrl);
    WebsocketCarInput.onopen    = function(event)
    {
        sendButtonInput("Speed",
document.getElementById("Speed").value);
        sendButtonInput("Light",
document.getElementById("Light").value);
        sendButtonInput("Pan",
document.getElementById("Pan").value);
        sendButtonInput("Tilt",
document.getElementById("Tilt").value);
    };
    WebsocketCarInput.onclose    =
function(event){setTimeout(initCarInputWebSocket, 2000)};
    WebsocketCarInput.onmessage = function(event){};
}

function initWebSocket()
{
    initCameraWebSocket ();
    initCarInputWebSocket();
}

function sendButtonInput(key, value)
{
    var data = key + "," + value;
    WebsocketCarInput.send(data);
}

window.onload = initWebSocket;
document.getElementById("mainTable").addEventListener("touchen
d", function(event){
    event.preventDefault()
});
</script>
</body>

```

```
</html>
```

```
)HTMLHOMEPAGE";
```

```
void rotateMotor(int motorNumber, int motorDirection)
{
    if (motorDirection == FORWARD)
    {
        digitalWrite(motorPins[motorNumber].pinIN1, HIGH);
        digitalWrite(motorPins[motorNumber].pinIN2, LOW);
    }
    else if (motorDirection == BACKWARD)
    {
        digitalWrite(motorPins[motorNumber].pinIN1, LOW);
        digitalWrite(motorPins[motorNumber].pinIN2, HIGH);
    }
    else
    {
        digitalWrite(motorPins[motorNumber].pinIN1, LOW);
        digitalWrite(motorPins[motorNumber].pinIN2, LOW);
    }
}
```

```
void moveCar(int inputValue)
{
    Serial.printf("Got value as %d\n", inputValue);
    switch(inputValue)
    {

        case UP:
            rotateMotor(RIGHT_MOTOR, FORWARD);
            rotateMotor(LEFT_MOTOR, FORWARD);
            break;

        case DOWN:
            rotateMotor(RIGHT_MOTOR, BACKWARD);
            rotateMotor(LEFT_MOTOR, BACKWARD);
            break;

        case LEFT:
            rotateMotor(RIGHT_MOTOR, FORWARD);
            rotateMotor(LEFT_MOTOR, BACKWARD);
            break;

        case RIGHT:
            rotateMotor(RIGHT_MOTOR, BACKWARD);
            rotateMotor(LEFT_MOTOR, FORWARD);
```



```

        break;

    case STOP:
        rotateMotor(RIGHT_MOTOR, STOP);
        rotateMotor(LEFT_MOTOR, STOP);
        break;

    default:
        rotateMotor(RIGHT_MOTOR, STOP);
        rotateMotor(LEFT_MOTOR, STOP);
        break;
    }
}

void handleRoot(AsyncWebServerRequest *request)
{
    request->send_P(200, "text/html", htmlHomePage);
}

void handleNotFound(AsyncWebServerRequest *request)
{
    request->send(404, "text/plain", "File Not Found");
}

void onCarInputWebSocketEvent(AsyncWebSocket *server,
                              AsyncWebSocketClient *client,
                              AwsEventType type,
                              void *arg,
                              uint8_t *data,
                              size_t len)
{
    switch (type)
    {
        case WS_EVT_CONNECT:
            Serial.printf("WebSocket client #%u connected from %s\n",
client->id(), client->remoteIP().toString().c_str());
            break;
        case WS_EVT_DISCONNECT:
            Serial.printf("WebSocket client #%u disconnected\n", client-
>id());
            moveCar(0);
            ledcWrite(PWMLightChannel, 0);
            panServo.write(90);
            tiltServo.write(90);
            break;
        case WS_EVT_DATA:
            AwsFrameInfo *info;

```

```

        info = (AwsFrameInfo*)arg;
        if (info->final && info->index == 0 && info->len == len &&
info->opcode == WS_TEXT)
        {
            std::string myData = "";
            myData.assign((char *)data, len);
            std::istringstream ss(myData);
            std::string key, value;
            std::getline(ss, key, ',');
            std::getline(ss, value, ',');
            Serial.printf("Key [%s] Value[%s]\n", key.c_str(),
value.c_str());
            int valueInt = atoi(value.c_str());
            if (key == "MoveCar")
            {
                moveCar(valueInt);
            }
            else if (key == "Speed")
            {
                ledcWrite(PWMSpeedChannel, valueInt);
            }
            else if (key == "Light")
            {
                ledcWrite(PWMLightChannel, valueInt);
            }
            else if (key == "Pan")
            {
                panServo.write(valueInt);
            }
            else if (key == "Tilt")
            {
                tiltServo.write(valueInt);
            }
        }
        break;
    case WS_EVT_PONG:
    case WS_EVT_ERROR:
        break;
    default:
        break;
}
}

void onCameraWebSocketEvent(AsyncWebSocket *server,
                            AsyncWebSocketClient *client,
                            AwsEventType type,
                            void *arg,

```

```

        uint8_t *data,
        size_t len)
{
    switch (type)
    {
        case WS_EVT_CONNECT:
            Serial.printf("WebSocket client #%u connected from %s\n",
client->id(), client->remoteIP().toString().c_str());
            cameraClientId = client->id();
            break;
        case WS_EVT_DISCONNECT:
            Serial.printf("WebSocket client #%u disconnected\n", client-
>id());
            cameraClientId = 0;
            break;
        case WS_EVT_DATA:
            break;
        case WS_EVT_PONG:
        case WS_EVT_ERROR:
            break;
        default:
            break;
    }
}

```

```

void setupCamera()
{
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_4;
    config.ledc_timer = LEDC_TIMER_2;
    config.pin_d0 = Y2_GPIO_NUM;
    config.pin_d1 = Y3_GPIO_NUM;
    config.pin_d2 = Y4_GPIO_NUM;
    config.pin_d3 = Y5_GPIO_NUM;
    config.pin_d4 = Y6_GPIO_NUM;
    config.pin_d5 = Y7_GPIO_NUM;
    config.pin_d6 = Y8_GPIO_NUM;
    config.pin_d7 = Y9_GPIO_NUM;
    config.pin_xclk = XCLK_GPIO_NUM;
    config.pin_pclk = PCLK_GPIO_NUM;
    config.pin_vsync = VSYNC_GPIO_NUM;
    config.pin_href = HREF_GPIO_NUM;
    config.pin_sscb_sda = SIOD_GPIO_NUM;
    config.pin_sscb_scl = SIOC_GPIO_NUM;
    config.pin_pwdn = PWDN_GPIO_NUM;
    config.pin_reset = RESET_GPIO_NUM;
    config.xclk_freq_hz = 20000000;

```

```

config.pixel_format = PIXFORMAT_JPEG;

config.frame_size = FRAMESIZE_VGA;
config.jpeg_quality = 10;
config.fb_count = 1;

// camera init
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK)
{
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}

if (psramFound())
{
    heap_caps_malloc_extmem_enable(20000);
    Serial.printf("PSRAM initialized. malloc to take memory from
psram above this size");
}
}

void sendCameraPicture()
{
    if (cameraClientId == 0)
    {
        return;
    }
    unsigned long startTime1 = millis();
    //capture a frame
    camera_fb_t * fb = esp_camera_fb_get();
    if (!fb)
    {
        Serial.println("Frame buffer could not be acquired");
        return;
    }

    unsigned long startTime2 = millis();
    wsCamera.binary(cameraClientId, fb->buf, fb->len);
    esp_camera_fb_return(fb);

    //Wait for message to be delivered
    while (true)
    {
        AsyncWebSocketClient * clientPointer =
wsCamera.client(cameraClientId);
        if (!clientPointer || !(clientPointer->queueIsFull()))

```

```

    {
        break;
    }
    delay(1);
}

unsigned long startTime3 = millis();
Serial.printf("Time taken Total: %d|%d|%d\n", startTime3 -
startTime1, startTime2 - startTime1, startTime3 - startTime2 );
}

void setUpPinModes()
{
    panServo.attach(PAN_PIN);
    tiltServo.attach(TILT_PIN);

    //Set up PWM
    ledcSetup(PWMSpeedChannel, PWMFreq, PWMResolution);
    ledcSetup(PWMLightChannel, PWMFreq, PWMResolution);

    for (int i = 0; i < motorPins.size(); i++)
    {
        pinMode(motorPins[i].pinEn, OUTPUT);
        pinMode(motorPins[i].pinIN1, OUTPUT);
        pinMode(motorPins[i].pinIN2, OUTPUT);
        /* Attach the PWM Channel to the motor enb Pin */
        ledcAttachPin(motorPins[i].pinEn, PWMSpeedChannel);
    }
    moveCar(STOP);

    pinMode(LIGHT_PIN, OUTPUT);
    ledcAttachPin(LIGHT_PIN, PWMLightChannel);
}

void setup(void)
{
    setUpPinModes();
    //Serial.begin(115200);

    WiFi.softAP(ssid, password);
    IPAddress IP = WiFi.softAPIP();
    Serial.print("AP IP address: ");
    Serial.println(IP);

    server.on("/", HTTP_GET, handleRoot);
    server.onNotFound(handleNotFound);
}

```

```
wsCamera.onEvent(onCameraWebSocketEvent);
server.addHandler(&wsCamera);

wsCarInput.onEvent(onCarInputWebSocketEvent);
server.addHandler(&wsCarInput);

server.begin();
Serial.println("HTTP server started");

setupCamera();
}

void loop()
{
    wsCamera.cleanupClients();
    wsCarInput.cleanupClients();
    sendCameraPicture();
    Serial.printf("SPIRam Total heap %d, SPIRam Free Heap %d\n",
ESP.getPsramSize(), ESP.getFreePsram());
}
```

LAMPIRAN GAMBAR

