

***PROTOTYPE KENDALI MOTOR DC DAN ESP32-CAM  
BERBASIS WEB SERVER LOCAL SEBAGAI  
MOBIL PENGANTAR***

**TUGAS AKHIR**

Diajukan Sebagai Salah Satu Syarat  
Untuk Kelulusan Program Diploma  
Program Studi Teknik Komputer



**Oleh :**

**Nama : Bagus Ilham Santoso**

**NPM : 43A87005190050**

**SEKOLAH TINGGI MANAJEMEN INFORMATIKA DAN KOMPUTER  
BANI SALEH  
BEKASI  
2022**

## LEMBAR PERSETUJUAN TUGAS AKHIR

Telah diperiksa dan disetujui oleh pembimbing untuk disidangkan pada Sidang Tugas Akhir Program Diploma Tiga (D3), Program Studi Teknik Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Bani Saleh Tugas Akhir dengan judul:

### ***PROTOTYPE KENDALI MOTOR DC DAN ESP32-CAM BERBASIS WEB SERVER LOCAL SEBAGAI MOBIL PENGANTAR***

Bekasi, 31 Oktober 2022

Pembimbing Utama,

Peembimbing Pendamping,

**Riesa Syaiful Akbar, S.T., M.T**

**NIDN. 0413049302**

**Rahmadi, S.Kom., M.Kom.**

**NIDN. 0431017502**

Mengetahui:

Ketua Program Studi Teknik Komputer,

**Rahmadi, S.Kom., M.Kom.**

**NIDN. 0431017502**

## LEMBAR PENGESAHAN TUGAS AKHIR

Telah disidangkan dan dinyatakan Lulus Sidang Tugas Akhir Program Diploma Tiga (D3), Program Studi Teknik Komputer Sekolah Tinggi Manajemen Informatika dan Komputer Bani Saleh pada **03 November 2022** Tugas Akhir dengan judul:

### ***PROTOTYPE KENDALI MOTOR DC DAN ESP32-CAM BERBASIS WEB SERVER LOCAL SEBAGAI MOBIL PENGANTAR***

Nama Penguji

Tanda Tangan

**Rahmadi, S.Kom., M.Kom.**

NIDN. 0431017502

\_\_\_\_\_

**Agustian Budiawana, Drs., M.Kom.**

NIDN. 0412086003

\_\_\_\_\_

Mengetahui :

Ketua Program Studi Teknik Komputer,

**Rahmadi, S.Kom., M.Kom.**

**NIDN. 0431017502**

## LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR

Nama : Bagus Ilham Santoso

NPM : 43A87005190050

Program Studi : Teknik Komputer

Judul Skripsi : PROTOTYPE KENDALI MOTOR DC DAN ESP32-CAM

BERBASIS WEB SERVER LOCAL SEBAGAI

MOBIL PENGANTAR

Dengan ini saya menyatakan bahwa dalam Tugas Akhir ini tidak terdapat karya yang pernah diajukan untuk memperoleh gelar Diploma Tiga (D3) di suatu Perguruan Tinggi, dan sepanjang pengetahuan saya juga tidak terdapat karya atau pendapat yang pernah ditulis atau diterbitkan oleh orang lain, kecuali yang secara tertulis dirujuk dalam naskah ini dan disebutkan dalam daftar pustaka.

Apabila dikemudian hari saya terbukti memberikan pernyataan yang tidak benar, saya bersedia menerima sanksi berupa pencabutan gelar Diploma Tiga (D3) saya.

Bekasi, 31 Oktober 2022

(Bagus Ilham Santoso )

## KATA PENGANTAR

*Assalaamu'alaikum Warrahmatullaahi Wabarakaatuh.*

Puji syukur kehadirat Allah SWT atas limpahan rahmat dan karunia-Nya hingga saya dapat menyelesaikan Laporan Tugas Akhir ini yang berjudul “PROTOTYPE KENDALI MOTOR DC DAN ESP32-CAM BERBASIS WEB SERVER LOCAL SEBAGAI MOBIL PENGANTAR” dengan sebaik - baiknya. Tidak lupa saya juga ingin mengucapkan terimakasih kepada kedua orang tua dan juga keluarga yang selalu mendukung saya secara materi maupun moral. Penyusunan laporan ini disusun sebagai salah satu syarat Kelulusan Program Diploma III bagi mahasiswa Program Diploma III Teknik Komputer.

Saya menyadari berkat banyaknya bantuan dan bimbingan dari pihak lain dalam proses pembuatan proposal ini, saya akhirnya dapat menyelesaikan Laporan Tugas Akhir ini dengan sebaik - baiknya. Bersama dengan laporan ini ucapan terima kasih diberikan kepada:

1. Bapak Kikim Mukiman, S.Kom., M.Kom selaku Ketua STMIK Bani Saleh yang telah memberikan kesempatan belajar bagi penulis untuk dapat menyelesaikan program diploma III di kampus ini.
2. Bapak Rahmadi, S.Kom., M.Kom. selaku Ketua Program Studi Teknik Komputer STMIK Bani Saleh Bekasi, yang banyak membantu penulis dalam mengarah penulisan tugas akhir ini.
3. Bapak Riesa Syaiful Akbar, S.T., M.T selaku Pembimbing Pertama Tugas Akhir dan dosen yang dengan sabar dan tekun memberikan arahan perbaikan yang berarti bagi penulis.
4. Bapak Rahmadi, S.Kom., M.Kom., selaku Pembimbing Kedua Tugas Akhir dan dosen yang telah membimbing dengan sabar, selalu memberikan solusi ketika penulis kesulitan tidak hanya itu serta yang telah membagi ilmu pengetahuan dan pengalaman materi yang sangat berarti bagi penulis.

5. Special *thanks for* Feggy Ferlyanti A , manusia yang bisa menyenangkan dan menyebalkan dalam satu waktu, yang selalu berusaha untuk membantu, memberikan saran serta menjadi salah satu motivasi penulis dalam menyelesaikan laporan ini.
6. Dan tak lupa seluruh teman-teman seangkatan, khususnya kelas D3 Teknik Komputer angkatan 2019 yang telah saling membantu dan memberikan semangat dalam menyelesaikan laporan ini.
7. Serta pihak-pihak lain yang tidak dapat penulis sebutkan satu persatu, terimakasih atas dukungan dan perhatiannya.

Penulis menyadari bahwa laporan ini masih jauh dari kata sempurna, dengan segala kerendahan hati penulis menerima kritik dan saran yang membangun demi kesempurnaan laporan ini. Semoga laporan ini dapat bermanfaat untuk semua pihak yang berkepentingan, khususnya mahasiswa program Diploma III Teknik Komputer

*Wassalaamu'alaikum Warrahmatullaahi Wabarakaatuh.*

Bekasi, 31 Oktober 2022

Bagus Ilham Santoso

## DAFTAR ISI

LEMBAR PERSETUJUAN TUGAS AKHIR.....	i
LEMBAR PENGESAHAN TUGAS AKHIR .....	ii
LEMBAR PERNYATAAN KEASLIAN TUGAS AKHIR .....	iii
KATA PENGANTAR .....	iv
DAFTAR ISI.....	vi
DAFTAR GAMBAR .....	ix
DAFTAR TABEL .....	x
BAB I PENDAHULUAN .....	1
1.1    Latar Belakang .....	1
1.2    Permasalahan Penulisan .....	1
1.2.1    Identifikasi Masalah .....	1
1.2.2    Batasan Masalah.....	2
1.2.3    Rumusan Masalah .....	2
1.3    Tujuan dan Manfaat Penulisan .....	2
1.3.1    Tujuan Penulisan.....	2
1.3.2    Manfaat Penulisan.....	3
1.4    Sistematika Penulisan.....	3
BAB II LANDASAN TEORI .....	5
2.1    ESP32-CAM.....	5
2.1.1    Datasheet ESP32-CAM.....	6
2.1.2    Motor <i>Driver</i> L298N.....	7
2.1.3    Datasheet L298N.....	7
2.2    Motor DC <i>Gearbox</i> .....	7

2.3	Baterai 18650 Ultra Fire .....	8
2.4	Web Server .....	9
2.4.1	Fungsi Web Server .....	9
2.4.2	Cara Kerja Web Server .....	10
2.5	Web Server Lokal .....	11
2.5.1	Jenis Aplikasi <i>Web Server</i> Lokal .....	11
2.5.2	Fungsi Web Server Lokal .....	12
2.6	Prototype .....	13
2.6.1	Tujuan Prototype .....	14
2.6.2	Manfaat Prototype .....	14
2.6.3	Contoh Prototype .....	15
2.7	Bahasa Pemrograman C .....	16
2.8	HTML (Hypertext Markup Language).....	18
2.9	CSS (Cascading Style Sheets) .....	19
2.10	JavaScript .....	19
2.11	Arduino IDE .....	20
<b>BAB III RANCANG BANGUN HARDWERE .....</b>		<b>21</b>
3.1	Diagram Blok Perangkat Keras .....	21
3.2	Flowchart Alat .....	23
3.3	Cara Kerja Alat .....	24
3.4	Schematic Alat .....	25
3.5	Source Code .....	26
<b>BAB IV HASIL DAN PEMBAHASAN .....</b>		<b>35</b>
4.1	Tegangan Pada Mobil.....	35
4.1.1	Arus Pada Mobil Pengantar .....	35



4.1.2	Tegangan Output Mobil .....	36
4.2	Kecepatan dan Beban .....	36
4.3	User Interface Web .....	37
4.4	Intesitas Cahaya Lampu dan Kualitas Kamera.....	38
4.4.1	Intesitas Cahaya 0% .....	38
4.4.2	Intesitas Cahaya 25% .....	39
4.4.3	Intesitas Cahaya 50% .....	40
4.4.4	Intesitas Cahaya 75% .....	41
4.4.5	Intesitas Cahaya 100% .....	42
4.4.6	Jarak Kontrol Mobil .....	43
4.4.7	Kualitas Kamera.....	44
BAB V KESIMPULAN DAN SARAN.....		45
5.1	Kesimpulan.....	45
5.2	Saran .....	45
DAFTAR PUSTAKA .....		46
LAMPIRAN LAMPIRAN .....		48

## DAFTAR GAMBAR

<b>Gambar 2. 1</b> ESP32-CAM.....	5
<b>Gambar 2.1. 1</b> Datasheet ESP32-CAM.....	6
<b>Gambar 2.1. 1</b> Datasheet ESP32-CAM.....	7
<b>Gambar 2. 2</b> Motor DC Gearbox .....	7
<b>Gambar 2. 3</b> Baterai 18650 Ultra Fire .....	8
<b>Gambar 2. 4</b> Web Server .....	9
<b>Gambar 2. 5</b> Web Server Local .....	11
<b>Gambar 2. 6</b> Prototype.....	13
<b>Gambar 2. 7</b> Structure of C Program.....	16
<b>Gambar 2. 8</b> HTML (Hypertext Markup Language).....	18
<b>Gambar 2. 9</b> CSS (Cascading Style Sheets) .....	19
<b>Gambar 2. 10</b> JavaScript.....	19
<b>Gambar 2. 11</b> Arduino IDE .....	20
<b>Gambar 2. 11</b> Arduino IDE Interface .....	21
<b>Gambar 3. 1</b> Diagram Blok Perangkat Keras .....	22
<b>Gambar 3. 2</b> Flowchart Alat .....	23
<b>Gambar 3. 3</b> Cara Kerja Alat.....	24
<b>Gambar 3. 4</b> Schematic Alat.....	25
<b>Gambar 4.4</b> User Interface Web .....	37
<b>Gambar 4.3. 1</b> Intensitas Cahaya 0% .....	38
<b>Gambar 4.3. 2</b> Intensitas Cahaya 25% .....	39
<b>Gambar 4.3. 3</b> Intensitas Cahaya 50% .....	40
<b>Gambar 4.3. 4</b> Intensitas Cahaya 75% .....	41
<b>Gambar 4.3. 5</b> Intensitas Cahaya 100% .....	42

## DAFTAR TABEL

<b>Tabel 3. 4</b> Schematic Alat ESP32-CAM .....	23
<b>Tabel 4. 1</b> Tegangan Pada Mobil.....	35
<b>Tabel 4.1. 1</b> Arus Pada Mobil Pengantar.....	36
<b>Tabel 4.1. 2</b> Tegangan Output Mobil .....	36
<b>Tabel 4. 2</b> Kecepatan dan Beban .....	36
<b>Tabel 4.4. 6</b> Jarak Kontrol Mobil .....	43
<b>Tabel 4.4. 7</b> Kualitas Kamera .....	44

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Dalam masa industri saat ini banyak perusahaan ingin meningkatkan efisiensi pekerjaan yang dilakukan sumber daya manusia dengan cepat dan tepat. Semakin cepat pekerjaan yang dilakukan karyawan maka semakin banyak produktifitas pekerjaan yang yang disanggupi oleh karyawan.

Saat pekerjaan divisi sudah menyelesaikan produksi suatu barang, maka barang yang sudah selesai dikerjakan maka akan dikirimkan ke divisi lain untuk diproses lebih lanjut lagi. Ketika pengiriman barang tersebut maka dibutuhkan satu orang untuk mengirimkan barang yang sudah jadi ke divisi lainnya. Maka, dalam divisi produksi harus menggantikan seseorang yang mengirimkan barang ke divisi lain atau mengurangi produktifitas sementara untuk menunggu orang tersebut selesai mengirimkan barang yang akan dikirimkan ke divisi lainnya.

Kecepatan dan Ketepatan dalam produksi sangat diperhitungkan lebih, untuk memaksimalkan hasil dan jumlah yang banyak dalam proses produksi. Jika hasil produksi yang di hasilkan sangat banyak maka perusahaan menguntungkan penjualan yang lebih. Maka dari itu, *Prototype* Mobil Pengantar ini dibuat untuk menggantikan tenaga manusia dalam pengiriman antar divisi di dalam satu industri, sehingga dapat menghemat waktu dan memaksimalkan pekerjaan di dalam divisi tersebut.

### **1.2 Permasalahan Penulisan**

#### **1.2.1 Identifikasi Masalah**

Berdasarkan penjelasan pada latar belakang diatas, maka dapat diambil permasalahan sebagai berikut :

- a. Waktu yang kurang efisien saat mengantarkan barang.
- b. Menunggu antrian untuk bergantian pengiriman barang.

### **1.2.2 Batasan Masalah**

Adapun batasan masalah yang penulis ambil untuk menghindari penulisan yang lebih luas adalah :

- a. Mikrokontroler yang digunakan adalah ESP32-CAM.
- b. Bahasa Pemrograman yang digunakan pada Mikrokontroler adalah Bahasa C.
- c. Bahasa Pemrograman untuk tampilan menggunakan bahasa HTML, CSS dan Javascript.
- d. ESP32-CAM digunakan sebagai *Access Point* dan *Web Server Local* untuk penghubung ke kontroler perangkat.
- e. *Prototype* ini hanya berfungsi untuk mendukung pengiriman barang berbasis *Web Server Local*.

### **1.2.3 Rumusan Masalah**

Berdasarkan penjelasan pada latar belakang, maka rumusan masalah yang didapat adalah :

- a. Bagaimana mobil pengantar berbasis ESP32-CAM bekerja mengantarkan barang ?
- b. Bagaimana merancang program mobil pengantar dengan kontroler melalui *web server local* ?

## **1.3 Tujuan dan Manfaat Penulisan**

### **1.3.1 Tujuan Penulisan**

Tujuan dari penulisan ini adalah :

- a. Untuk membuat rancangan sebuah alat transportasi yang dapat digunakan sebagai alat pengantar barang antar divisi yang menggunakan kontrol jarak jauh menggunakan aplikasi android.
- b. Untuk memenuhi Tugas Akhir jurusan D3 Teknik Komputer STMIK Bani Saleh.

### **1.3.2 Manfaat Penulisan**

Manfaat dari penulisan ini adalah :

- a. Sebagai alat bantu untuk pengiriman barang/komponen antar divisi dalam perusahaan.
- b. Memberikan kemudahan dan kecepatan untuk karyawan industri mengirimkan komponen/bahan produksi antar divisi
- c. Mempraktikan dan menerapkan ilmu pengetahuan IoT yang didapat selama mengikuti perkuliahan.
- d. Mengembangkan kreatifitas bagi penulis dibidang Teknik Komputer sebagai bidang yang dipelajari.

### **1.4 Sistematika Penulisan**

Sistematika penulisan dibagi menjadi beberapa bab dengan pokok pembahasan yang bertujuan untuk memudahkan pembaca agar memahami isi dari tugas akhir ini. Adapun sistematika penulisan Tugas Akhir ini adalah diantaranya :

#### **BAB I                      PENDAHULUAN**

Bab pendahuluan ini mendeskripsikan latar belakang permasalahan. Selain itu dalam bab ini juga dipaparkan tentang identifikasi masalah, batasan masalah, rumusan masalah, tujuan dan manfaat penelitian serta sistematika penulisan.

#### **BAB II                    LANDASAN TEORI**

Penjelasan tentang pengertian mikrokontroler serta tentang teori pengertian spesifikasi alat – alat yang digunakan pada penelitian.

### **BAB III RANCANG BANGUN *HARDWARE***

Bab ini membahas perancangan alat dan metode serta sistematis alat *Prototype* Kendali Motor DC dan ESP32-CAM berbasis *Web Server Local* sebagai Mobil Pengantar

### **BAB IV HASIL DAN PEMBAHASAN**

Bab ini membahas mengenai hasil beserta pembahasannya dari penelitian yang telah dilakukan.

### **BAB V KESIMPULAN DAN SARAN**

Kesimpulan yang diperoleh dari penelitian ini beserta beberapa saran untuk pengembangan alat pada laporan penulisan ini.

### **DAFTAR PUSTAKA**

### **LAMPIRAN - LAMPIRAN**

## **BAB II**

### **LANDASAN TEORI**

Perancangan mobil pengantar dengan menggunakan Mikrokontroler ESP32-CAM perlu beberapa pengetahuan dan teori dari komponen yang digunakan, komunikasi data, aktuator serta media yang digunakan untuk dapat mengakses ke dalam hardware.

*Carrier car* merupakan cara pengiriman mobil menggunakan truk yang sudah dimodifikasi sedemikian rupa hingga bisa membawa mobil kecil berjumlah sedikitnya 6 mobil dan maksimal 8. Secara kemampuan angkut, *carrier car* berbeda dengan *towing car*. Biaya yang dikeluarkan juga tidak mahal untuk menggunakan *carrier car*. Namun untuk kebutuhan yang tiba-tiba atau mendesak, jasa pengiriman *carrier car* tidak bisa menjadi pilihan.

Dari kedua jenis pilihan cara pengiriman mobil di atas, semua bisa digunakan sesuai kemampuan dan tingkat urgensi. Jika mobil dibutuhkan dalam waktu dekat, *towing car* bisa jadi pilihan. Namun, biaya yang dikeluarkan juga cukup mahal. Dan sebaliknya, untuk menekan biaya dan mobil tidak dibutuhkan secara cepat, *carrier car* bisa menjadi pilihan terbaik. (NDE CARGO 2021)

#### **2.1 ESP32-CAM**



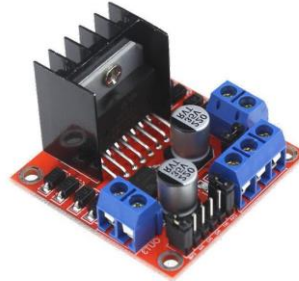
**Gambar 2. 1 ESP32-CAM**

ESP32-CAM merupakan salah satu mikrokontroler yang cocok untuk digunakan *Home Smart Devices*, *Industrial wireless control*, *Wireless monitoring* dan aplikasi IoT lainnya. Mikrokontroler ini lebih sedikit pin I/O dibandingkan dengan modul ESP32 Wroom / ESP32 Dev Kit, hal ini dikarenakan sudah banyak pin yang digunakan secara internal untuk fungsi kamera dan fungsi slot kartu *microSD* selain itu, ESP32-CAM tidak





### 2.1.2 Motor *Driver* L298N



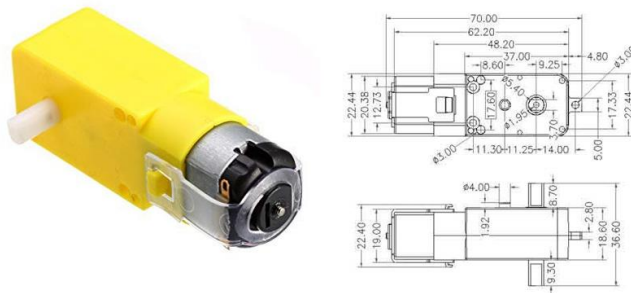
**Gambar 2.1. 2** *Motor Driver L298N*

*Motor Driver* L298N adalah *H-Bridge Dual Motor Controller* 2A yang memungkinkan kita untuk mengatur putaran motor maupun kecepatan dari motor DC. IC L298 mampu mengendalikan beban-beban induktif, motor DC dan motor *stepper*. (nyebarilmu.com 2017)

### 2.1.3 Datasheet L298N

- Menggunakan IC L298N (*Double H bridge Drive Chip*)
- Tegangan minimal untuk masukan power antara 5V-35V
- Tegangan operasional : 5V
- Arus untuk masukan antara 0-36mA
- Arus maksimal untuk keluaran per *Output A* maupun B yaitu 2A
- Daya maksimal yaitu 25W
- Dimensi modul yaitu 43 x 43 x 26mm
- Berat : 26g

## 2.2 Motor DC Gearbox



**Gambar 2. 2 Motor DC Gearbox**

Datasheet Motor DC Gearbox :

- a) *Rated Voltage: 3~6V*
- b) *Continuous No-Load Current: 150mA +/- 10%*
- c) *Min. Operating Speed (3V): 90+/- 10% RPM*
- d) *Min. Operating Speed (6V): 200+/- 10% RPM*
- e) *Torque: 0.15Nm ~0.60Nm*
- f) *Stall Torque (6V): 0.8kg.cm*
- g) *Gear Ratio: 1:48*
- h) *Body Dimensions: 70 x 22 x 18mm*
- i) *Wires Length: 200mm & 28 AWG*
- j) *Weight: 30.6g*

(Adafruit n.d.)

### 2.3 Baterai 18650 Ultra Fire

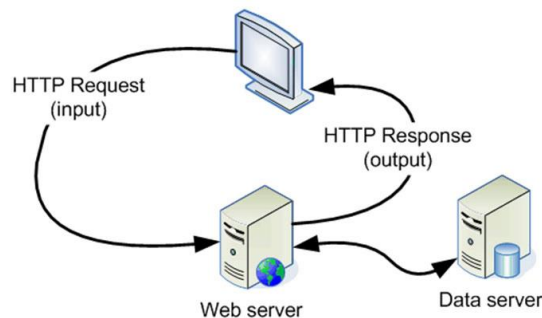


**Gambar 2. 3** Baterai 18650 *Ultra Fire*

<i>Type</i>	<i>B8630</i>
<i>Capacity</i>	<i>3000mAh</i>
<i>Rated Voltage</i>	<i>3.7V</i>
<i>Charge Termination Voltage</i>	<i>4.2V</i>

<b><i>Discharge Termination Voltage</i></b>	2.5V
<b><i>Maximum Charge Current</i></b>	3A
<b><i>Maximum Discharge Current</i></b>	9A
<b><i>Material</i></b>	Lithium
<b><i>Weight</i></b>	46g
<b><i>Dimensions</i></b>	6.7*1.8*1.8(CM)
<b><i>Packing List</i></b>	UltraFire 18650 3.7V 3000mAh Rechargeable Lithium Battery * 2
<b><i>Certification</i></b>	CE FCC MSDS UN38.8
(Ultrafire n.d.)	

## 2.4 Web Server



**Gambar 2. 4** Web Server

Web server adalah sebuah software (perangkat lunak) yang memberikan layanan berupa data. Berfungsi untuk menerima permintaan HTTP atau HTTPS dari klien atau kita kenal dengan *web browser* (*Chrome, Firefox*). Selanjutnya ia akan mengirimkan respon atas permintaan tersebut kepada client dalam bentuk halaman web. (Dicoding 2021)

### 2.4.1 Fungsi Web Server

Jika mau dibedah lebih dalam lagi, fungsi utama web server adalah untuk mentransfer berkas yang diminta pengguna. Biasanya, sebuah laman web terdiri dari berbagai macam informasi dalam berbagai

bentuk, seperti teks, video, gambar, audio, atau dokumen. Berkas-berkas inilah yang nantinya akan dikirimkan ke browser-mu sesuai permintaan.

Pada praktiknya langsung, web server akan merespon permintaan yang kamu tulis di address bar dan informasi yang kamu minta tadi akan ditampilkan ke browser-mu. Jika permintaanmu tidak bisa dicari, maka web server akan melakukan pengiriman balik berupa penolakan dengan cara memberikan informasi yang biasa dikenal dengan kode '404', artinya kata kunci yang kamu cari tidak bisa ditemukan. (Dewaweb 2022)

#### **2.4.2 Cara Kerja Web Server**

- a.** Konsumen memasukkan permintaan tertentu lewat *web browser*. Entah pergi ke sebuah *website*, mengirim email, mencari konten pada suatu blog, dsb.
- b.** Browser meminta data yang dibutuhkan melalui HTTP/HTTPS. HTTP/HTTPS adalah protokol untuk membantu proses pemindahan informasi seperti teks, gambar, video, dan komponen lainnya.
- c.** *Web server* mencari apa yang browser minta di dalam hosting. Hosting adalah tempat untuk menyimpan file-file yang dibutuhkan *website*.
- d.** Setelah datanya ketemu, *web server* mengirimkan seluruh data ini lewat HTTP/HTTPS. Namun jika tidak, *web server* akan mengirimkan halaman *error*.
- e.** Browser menampilkan informasi kepada konsumen. Baik menampilkan halaman *website*, rekomendasi konten, dll. (Benefita 2022)

## 2.5 Web Server Local



**Gambar 2. 5** *Web Server Local*

Localhost adalah server lokal yang bertugas untuk menampung segala kebutuhan data untuk program yang akan dijalankan. Perangkat komputer yang sedang kamu gunakan saat ini, bisa dibilang sebagai localhost. Setiap perangkat komputer pada dasarnya memiliki server lokal. Kamu bisa memanfaatkannya untuk membangun website secara offline, sebelum pada akhirnya website tersebut dapat diakses secara online oleh pengguna di internet.

Kamu bisa mengakses server lokal di perangkatmu dengan mengetikkan `http://localhost` di aplikasi browser milikmu. Localhost sendiri merupakan nama domain dengan IP address 127.0.0.1 atau yang dikenal juga dengan loopback address.

### 2.5.1 Jenis Aplikasi Web Server Lokal

#### a. XAMPP

Salah satu aplikasi pendukung *localhost* adalah XAMPP. Aplikasi *open source* gratis dan *cross platform* ini, dikembangkan oleh *Apache and Friends*. Melalui aplikasi ini, kamu dapat menggunakan *Apache HTTP Server*, *MariaDB* sebagai database-nya, dan PHP serta *Pearl* sebagai bahasa pemrogramannya. Aplikasi XAMPP dapat diunduh pada perangkat yang menggunakan sistem operasi Windows, Linux, dan MacOS.

#### **b. AMPPS**

AMPPS merupakan aplikasi yang dikembangkan oleh *Softaculous* dan bersifat *open source*. Di dalamnya, terdapat kumpulan perangkat lunak yang dapat kamu gunakan seperti *Apache*, *MySQL*, *MongoDB*, *PHP*, *Perl*, *Python*, serta penginstalan otomatis *Softaculous* untuk *desktop* dan *server* kantor. Aplikasi pendukung *localhost* ini dapat diinstall pada perangkat dengan sistem operasi Windows, Linux, dan MacOS.

#### **c. WAMPSEVER**

Aplikasi terakhir yang dapat mendukung kinerja *localhost* adalah *WampServer*. Aplikasi ini dikembangkan oleh *Romain Bourdon*, dan khusus digunakan untuk perangkat yang menggunakan sistem operasi Windows.

### **2.5.2 Fungsi Web Server Local**

#### **f. Menjadi Server Offline**

*Localhost* adalah *server* lokal yang bekerja di satu perangkat komputer saja dan tidak bisa diakses secara bersamaan seperti web hosting. Untuk itu, kamu dapat memanfaatkan server ini sebagai *server offline*.

Memiliki *server offline* adalah hal yang penting untuk *developer website*. Kamu akan lebih leluasa untuk membuat website menggunakan *server offline*. Sehingga, nantinya akan dapat meminimalisir kesalahan sebelum *website* yang kamu buat dapat diakses secara *online*.

#### **g. Menguji Website Secara Offline**

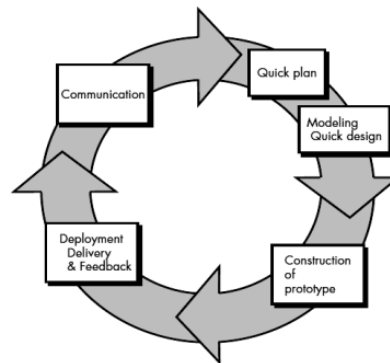
Bagi seorang *developer website*, pengecekan *website* pada saat proses pembuatan merupakan hal yang tidak boleh dilewatkan. Kamu bisa melakukan pengecekan, apakah website yang kamu buat sudah berjalan dengan baik atau belum. Hingga akhirnya *website* benar-benar siap dan dapat ditayangkan secara online.

Selain itu, pengecekan melalui server lokal tentunya tidak membutuhkan biaya apapun. Kamu juga tidak perlu memikirkan tentang batasan *bandwidth server*.

#### **h. Mengecek Koneksi internet**

*Localhost* juga dapat digunakan untuk memeriksa koneksi internet Anda. Untuk melakukannya, Anda hanya perlu membuka *Command Prompt* (Windows) atau *Terminal* (Mac OS) dan mengetikkan “ping *localhost*” atau “ping 127.0.0.1”. Setelah itu, tekan tombol enter dan lihat hasilnya. (Napizahni 2022)

## **2.6 Prototype**



**Gambar 2. 6** *Prototype*

*Prototype* diciptakan untuk melakukan beberapa uji coba sekaligus. Tujuannya untuk mengetahui apakah konsep yang sudah diperkenalkan tadi dapat diimplementasikan atau hanya sekadar menguji selera pasar. Pembuatan *prototype* adalah langkah yang dilakukan setelah menemukan ide untuk pembuatan produk baru. *Prototype* hanya bersifat uji coba, bukan final. Jadi, produk *prototype* tidak akan diedarkan atau dipublikasikan langsung kepada *end user* nantinya. Dalam dunia programming, *prototype* dibuat hanya untuk memenuhi kebutuhan di awal *software development*. Ini dilakukan guna mengetahui bagaimana fungsi dan fitur pada program bekerja. Apakah sesuai dengan yang diperkirakan atau tidak. Apabila ada kesalahan atau *error* akan



lebih mudah dideteksi jika menggunakan *prototype* terlebih dahulu. Sehingga produk yang nantinya dipublikasikan kepada end user sudah sempurna. (Academy 2022)

### **2.6.1 Tujuan Prototype**

Tujuan utama dari *prototype* adalah mengembangkan model atau rancangan produk menjadi produk final yang dapat memenuhi permintaan pengguna. Dalam proses pengembangan produk, pengguna dapat ikut andil dalam proses pengembangan produk dengan cara mengevaluasi dan memberikan umpan balik. Umpan balik yang diberikan dapat digunakan sebagai acuan dalam pengembangan produk. Selain itu, penggunaan prototipe dapat memunculkan ide-ide baru yang bisa dikembangkan menjadi sebuah fitur untuk melengkapi produk.

### **2.6.2 Manfaat Prototype**

Ada banyak manfaat dan keuntungan yang bisa kamu dapatkan saat menggunakan sistem prototyping ini. Berikut adalah manfaatnya.

#### **a. Dapat menghemat waktu dan biaya pengembangan produk**

Yang pertama adalah kamu dapat menekan biaya dan menghemat waktu dalam proses pengembangan produk. Dengan begitu, sumber daya yang tersisa dapat dialokasikan untuk kebutuhan yang lain.

#### **b. Dapat mengetahui kebutuhan pengguna terlebih dahulu**

Manfaat yang kedua yaitu dengan memanfaatkan sistem prototyping kamu dapat mengetahui kebutuhan pengguna terlebih dahulu. Sehingga kamu dan tim-mu dapat mengetahui apa saja prioritas dan kebutuhan pengguna. Dengan begitu proses pengembangan produk akan berlangsung lebih cepat.

#### **c. Menjadi acuan untuk mengembangkan produk**

Selanjutnya, model *prototype* dapat menjadi acuan atau patokan untuk kamu dalam mengembangkan sebuah produk. Kamu juga

dapat menggunakan *prototype* untuk menemukan kekurangan dan mencari solusi untuk membuat produk kamu menjadi semakin baik lagi.

**d. Dapat menjadi bahan presentasi**

Terakhir, kamu dapat menggunakan *prototype* untuk mempresentasikan produk yang akan kamu luncurkan. Dengan adanya *prototype* akan memudahkan audiens mendapatkan gambaran tentang produk yang akan diluncurkan. (Setiawan 2021)

### 2.6.3 Contoh Prototype

Berikutnya adalah mengenal apa saja contoh prototype. Selain mengetahui jenis dan manfaatnya, Anda juga perlu untuk mengenal contoh-contoh prototipe yang ada. berikut adalah tiga contoh prototipe yang perlu untuk Anda ketahui:

**A. HTML *Prototype***

Sesuai dengan namanya, jenis pembuatan prototipe ini memiliki tingkat kesulitan nomor satu. Sebagai praktisi UI/UX yang akan membuat prototipe contoh ini wajib menguasai kemampuan coding yang cakap. Prototipe yang terbuat dengan metode ini merupakan hasil dari perancangan kode oleh seorang desainer sistem. Dengan memberikan rancangan secara sesuai maka akan membantu perkembangan rencana sebuah proyek.

Membuat *prototype* dengan cara HTML memang membutuhkan desainer ahli coding, namun sisi positifnya adalah hemat biaya. Hal ini karena tidak terlalu banyak pengeluaran dalam perancangan prototipe. Tetapi perlu Anda ketahui kembali bahwa desainer yang melakukan haruslah benar-benar seseorang yang mampu untuk melakukannya.

**B. *Low-Fidelity***

Metode paling awal dalam membuat *prototype* adalah menggunakan media kertas. Seiring berkembangnya zaman, nyatanya cara ini juga

masih ada yang menerapkannya. Dengan menggoreskan pensil dan membuat sketsa dua dimensi menjadikan contoh ini juga disebut sebagai *Paper Prototyping*.

Contoh prototipe dengan metode ini pun menumbuhkan rasa kerjasama tim yang optimal dan menggunakan media yang sederhana. Setelah sketsa selesai maka dapat melakukan trial berdasarkan prototipe. Proses revisi pun dapat langsung Anda atau desainer lakukan karena tidak perlu proses yang rumit.

### **C. High-Fidelity**

Contoh prototipe yang ketiga adalah *High-Fidelity Prototype* dengan contoh ini adalah memberikan hasil berupa bentuk digital. Karena memberikan hasil digital maka desainer membuat prototipe ini melalui alat yang dapat berfungsi untuk merancanganya. *Software* yang umum untuk membuatnya adalah *Keynote* atau *Microsoft Power Point*.

Jenis rancangan yang dapat Anda tampilkan dalam contoh ini adalah menghadirkan layout bervariasi yang lebih berkarakter. Anda dapat memberikan permainan warna atau tata letak yang menggambarkan karakter produk secara UI agar nantinya dapat menghasilkan *feedback* pengguna yang sesuai dengan ekspektasi.

## **2.7 Bahasa Pemrograman C**

C adalah bahasa pemrograman prosedural. Ini awalnya dikembangkan oleh Dennis Ritchie pada tahun 1972. Ini terutama dikembangkan sebagai bahasa pemrograman sistem untuk menulis sistem operasi. Fitur utama dari bahasa C termasuk akses memori tingkat rendah, satu set kata kunci yang sederhana, dan gaya yang bersih, fitur-fitur ini membuat bahasa C cocok untuk pemrograman sistem seperti sistem operasi atau pengembangan kompiler.

Banyak bahasa kemudian meminjam sintaks/fitur secara langsung atau tidak langsung dari bahasa C. Seperti sintaks Java, PHP, JavaScript, dan banyak bahasa lainnya terutama didasarkan pada bahasa C. C++ hampir merupakan *superset* dari bahasa C (Beberapa program dapat dikompilasi dalam C, tetapi tidak dalam C++). Komponen pertama dan terpenting adalah penyertaan file *Header* dalam program C.

Structure of C Program	
Header	#include <stdio.h>
main()	int main() {
Variable declaration	int a = 10;
Body	printf( "%d ", a );
Return	return 0; }

**Gambar 2. 7** *Structure of C Program*

*File header* adalah *file* dengan ekstensi *.h* yang berisi deklarasi fungsi C dan definisi makro untuk dibagikan di antara beberapa file sumber.

Beberapa *file C Header*:

- **stddef.h** – Mendefinisikan beberapa tipe dan makro yang berguna.
- **stdint.h** – Mendefinisikan tipe integer lebar yang tepat.
- **stdio.h** – Mendefinisikan fungsi input dan output inti
- **stdlib.h** – Mendefinisikan fungsi konversi numerik, generator jaringan pseudo-acak, alokasi memori
- **string.h** – Mendefinisikan fungsi penanganan string
- **math.h** – Mendefinisikan fungsi matematika umum

Bagian selanjutnya dari program C adalah deklarasi variabel. Ini mengacu pada variabel yang akan digunakan dalam fungsi. Harap dicatat bahwa dalam program C, tidak ada variabel yang dapat digunakan tanpa dideklarasikan. Juga dalam program C, variabel harus dideklarasikan sebelum operasi apa pun dalam fungsi. *body* fungsi dalam program C, mengacu pada operasi yang dilakukan dalam fungsi. Itu bisa apa saja seperti manipulasi, pencarian, penyortiran, pencetakan, dll. Bagian terakhir dari program C adalah pernyataan pengembalian. Pernyataan kembali mengacu pada pengembalian nilai dari suatu fungsi. Pernyataan pengembalian dan nilai pengembalian ini bergantung pada tipe pengembalian fungsi. Misalnya, jika tipe pengembalian batal, maka tidak akan ada pernyataan pengembalian. Dalam kasus lain, akan ada pernyataan pengembalian dan nilai pengembalian akan menjadi tipe dari tipe pengembalian yang ditentukan. (geeksforgeeks.org 2022)

Pemrogram sebenarnya mempunyai keleluasaan dalam menuliskan bentuk atau gaya kode di sketsa. Namun hendaknya ditulis dalam bentuk yang mudah dibaca oleh siapa saja, bukan dalam bentuk yang paling pampat

## 2.8 HTML (Hypertext Markup Language)



**Gambar 2. 8** HTML (*Hypertext Markup Language*)

HTML adalah adalah singkatan dari *Hypertext Markup Language*. HTML memungkinkan seorang pengguna dapat membuat dan menyusun bagian heading, paragraf, link atau tautan, dan blockquote untuk halaman sebuah website. HTML sebenarnya bukanlah bahasa pemrograman, artinya HTML tidak punya kemampuan untuk membuat fungsionalitas yang dinamis. (Dicoding 2020)

## 2.9 CSS (Cascading Style Sheets)



**Gambar 2. 9** CSS (*Cascading Style Sheets*)

CSS adalah singkatan dari *cascading style sheets*, yaitu bahasa yang digunakan untuk menentukan tampilan dan format halaman website. Dengan CSS, Anda bisa mengatur jenis font, warna tulisan, dan latar belakang halaman. CSS digunakan bersama dengan bahasa markup, seperti HTML dan XML untuk membangun sebuah *website* yang menarik dan memiliki fungsi yang berjalan baik. CSS juga berguna untuk mengatasi keterbatasan HTML dalam mengatur format halaman website. Kenapa demikian? Apabila hanya menggunakan HTML ketika membangun website dengan beberapa halaman, Anda harus menulis tag untuk sebuah elemen HTML di semua halaman tersebut. Dengan adanya CSS, Anda cukup menulis kode satu kali untuk sebuah elemen HTML untuk diterapkan ke semua halaman. Nantinya, ketika akan melakukan perubahan, Anda juga cukup melakukan perubahan pada satu kode tadi. (Nayoan 2022)

## 2.10 JavaScript



**Gambar 2. 10** JavaScript

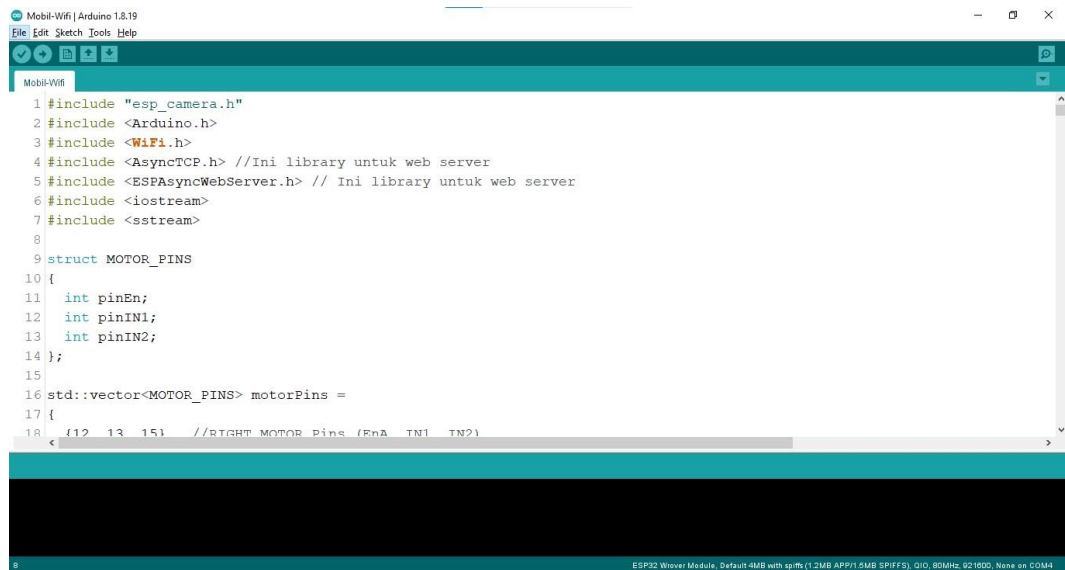
*JavaScript* adalah bahasa pemrograman berbasis teks yang digunakan baik di sisi klien dan sisi *server* yang memungkinkan seseorang membuat halaman *web* menjadi interaktif. Jika HTML dan CSS adalah bahasa yang memberikan struktur dan gaya ke halaman web, *JavaScript* adalah yang memberikan elemen interaktif halaman web yang melibatkan pengguna. Bicara teknis, *JavaScript* adalah bahasa pemrograman jenis *interpreter*, sehingga kamu tidak memerlukan compiler untuk menjalankannya. *JavaScript* memiliki fitur-fitur seperti berorientasi objek, *client-side*, *high-level programming*, dan *loosely typed*. Memasukkan *JavaScript* meningkatkan pengalaman pengguna halaman *web* dengan mengubahnya dari halaman statis menjadi halaman interaktif. Penggunaan kode *JavaScript* pada sebuah *website* bersifat opsional, artinya tidak harus selalu ada. Namun, mayoritas *website* dan blog modern saat ini menggunakan kode *JavaScript*. Contoh umum *JavaScript* yang mungkin kamu gunakan setiap hari termasuk kotak pencarian di *e-commerce*, video rekap berita yang disematkan di media online, atau *refresh timeline* Twitter. Selain itu, semenjak adanya *Node.js*, *JavaScript* dapat digunakan di luar dari *browser*. Misalnya, mengembangkan *back-end* (*server*), *console*, program *desktop*, *mobile*, IoT, dan game. (Rheny 2022)

## 2.11 Arduino IDE



**Gambar 2. 11** Arduino IDE

Perangkat Lunak Arduino (IDE) open-source memudahkan penulisan kode dan mengunggahnya ke papan. Perangkat lunak ini dapat digunakan dengan papan Arduino apa pun. (arduino.cc n.d.)



**Gambar 2. 11** *Arduino IDE Interface*

Arduino IDE adalah software yang digunakan untuk membuat sketch pemrograman atau dengan kata lain arduino IDE sebagai media untuk pemrograman pada board yang ingin diprogram. Arduino IDE ini berguna untuk mengedit, membuat, meng-upload ke board yang ditentukan, dan meng-coding program tertentu. Arduino IDE dibuat dari bahasa pemrograman JAVA, yang dilengkapi dengan library C/C++(wiring), yang membuat operasi input/output lebih mudah. (allgoblog.com 2017)

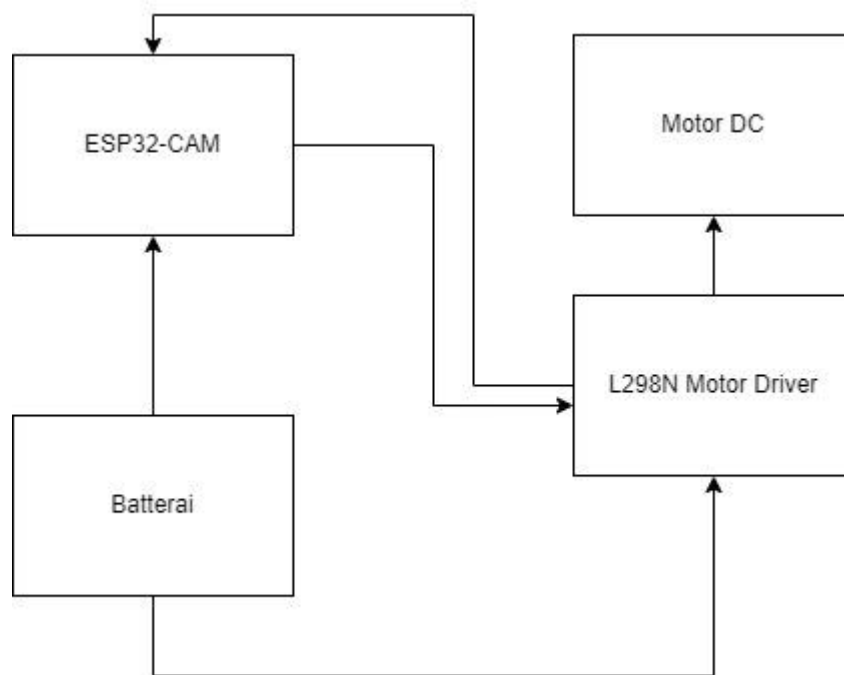


### **BAB III**

#### **RANCANG BANGUN HARDWARE**

Pada berikut ini adalah tahapan pembuatan dan rancangan logika pembuatan alat Prototype Kendali Motor DC dan ESP32-CAM berbasis Web Server Local sebagai Mobil Pengantar.

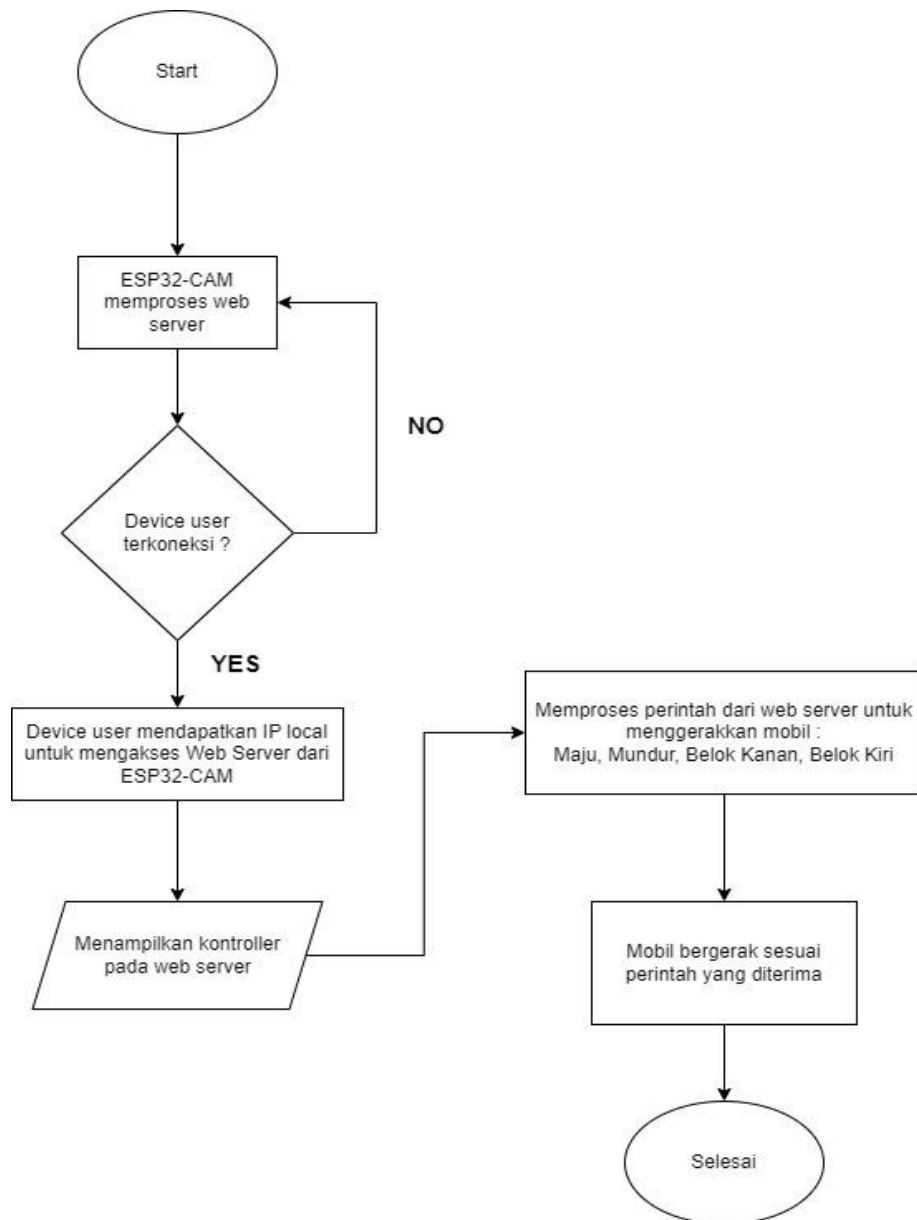
##### **3.1 Diagram Blok Perangkat Keras**



**Gambar 3. 1** Diagram Blok Perangkat Keras

Penjelasan dalam blok diagram ini memberitahukan kerja alat dari baterai memberikan suplai daya kepada Mikrokontoler ESP32-CAM dan L298N Motor Driver. ESP32-CAM akan memberikan perintah kepada L298N untuk menggerakkan Motor DC sesuai dengan yang diperintahkan dan juga L298N akan memberikan respon kepada ESP32-CAM untuk memberikan reaction dari perintah yang diterima L298N.

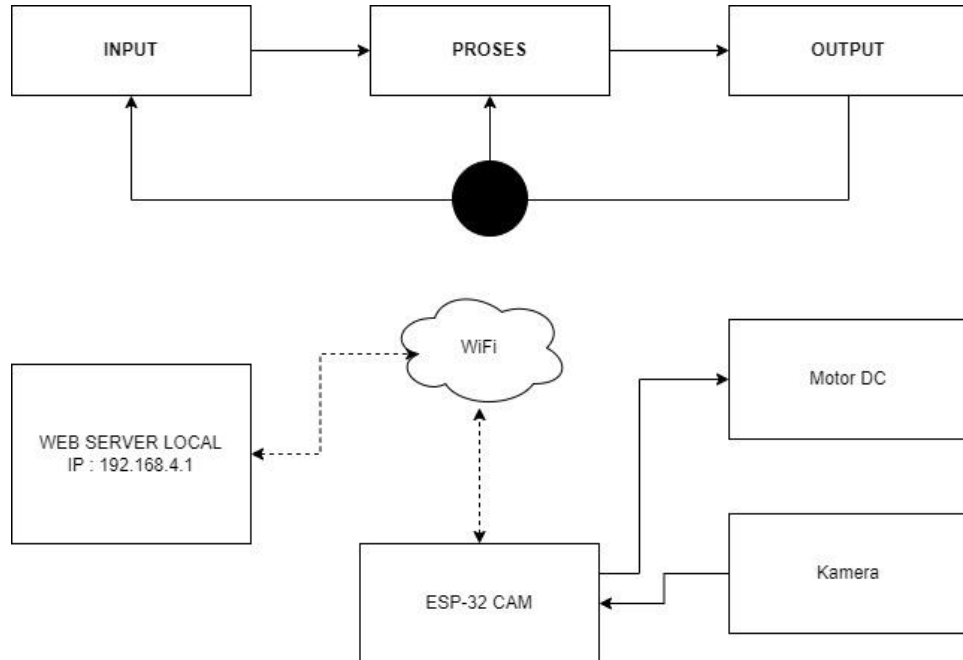
### 3.2 Flowchart Alat



**Gambar 3. 2** *Flowchart Alat*

ESP32-CAM sebagai piranti Access Point sehingga user harus mengkoneksikan ke dalam Wifi ESP32-CAM untuk mendapatkan IP Local dari ESP32-CAM untuk mengakses web server kontroler mobil, lalu user dapat memberikan perintah maju, mundur, belok kiri, belok kanan serta juga dapat mengendalikan lampu pada ESP32-CAM pada web server kontroler.

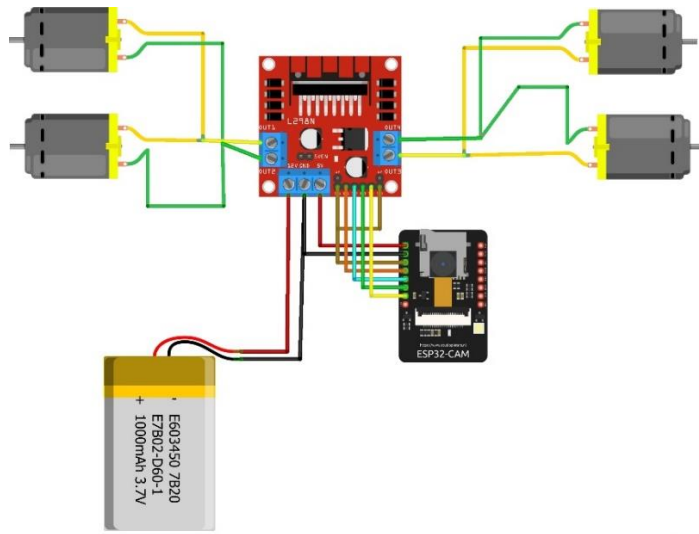
### 3.3 Cara Kerja Alat



**Gambar 3. 3** Cara Kerja Alat

Ketika input dari Kamera menampilkan gambar yang tertangkap dan Web Server Local sudah menerima perintah dari user maka akan dikirimkan ke ESP32-CAM untuk di proses menuju Motor Driver L298N untuk mengeluarkan output Motor DC untuk mengatur pergerakan maju, mundur, belok kiri dan belok kanan.

### 3.4 Schematic Alat



**Gambar 3. 4** Schematic Alat

Ini adalah schematic alat yang menggunakan ESP32-CAM yang di desain menggunakan aplikasi *Fritzing*. Dalam rancangan ini menggunakan 4 DC motor melalui motor driver L298N dengan tegangan 5V sampai dengan 35V dengan besar maksimal daya yang ditampung yaitu 25 watt. ESP32-CAM di koneksikan dengan seperti dalam tabel berikut ini :

**Tabel 3. 4** Schematic Alat ESP32-CAM

ESP-32 CAM	L298N
GPIO 13	IN1
GPIO 15	IN2
GPIO 14	IN3
GPIO 2	IN4
GPIO 12	ENA, ENB
5V	+5V
GND	GND

### 3.5 Source Code

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

Pada bagian ini untuk mengambil library yang dibutuhkan menggerakan mobil dan dapat menyalakan web server pada device ESP32-CAM, library yang paling penting di masukkan AsyncTCP dan ESPAsyncWebServer untuk dapat memberikan Two-Way Handshake pada device yang terkoneksi pada Wifi mobil

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

std::vector<MOTOR_PINS> motorPins =
{
    {12, 13, 15}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
    {12, 14, 2}, //LEFT_MOTOR Pins (EnB, IN3, IN4)
};
```

Disini kita akan mendefinisikan pin yang akan digunakan pada motor driver L298N sehingga dapat memproses perintah dari ESP32-CAM.

```
#define LIGHT_PIN 4
```

Ini mendefinisikan secara template pada modul ESP32-CAM dan dibutuhkan untuk mengatur lampu pada ESP32-CAM.

```
const char* ssid = "MyWiFiCar";  
const char* password = "12345678";  
AsyncWebServer server(80);  
AsyncWebSocket wsCamera("/Camera");  
AsyncWebSocket wsCarInput("/CarInput");  
uint32_t cameraClientId = 0;
```

Di bagian ini kita akan mengatur nama Wifi dan password untuk device yang akan terkoneksi disini kita mengubah sesuai kebutuhan dan syarat yang digunakan sebagai password adalah dengan metode keamanan wireless WPA-PSK (Password Personal Key) sehingga membutuhkan 8 karakter saja tanpa harus ada kombinasi.

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

Disini kita mendeklarasikan Web Server akan dijadikan di port 80 (HTTP) sehingga ketika mengakses web server tersebut URL yang digunakan secara default oleh web browser.

```

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

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

window.onload = initWebSocket;

document.getElementById("mainTable").addEventListener("touchend", function(event){
    event.preventDefault()
});
</script>
</body>
</html>
)HTMLHOMEPAGE";

```

Kode diatas adalah mengatur tampilan *homepage web server* menggunakan HTML dan CSS. Mengkoneksikan parameter dan variabel pada bahasa C arduino yaitu menggunakan bahasa pemrograman *JavaScript* memasukkan element yang didapat dari bahasa C.

```

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;
}
}
```

Disini kita mengatur element UP, DOWN, RIGHT, LEFT supaya dapat digerakkan melewati motor driver.

```

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.prin break;
case WS_EVT_DISCONNECT:
    Serial.printf("WebSocket client #%u disconnected\n", client->id());
    moveCar(0);
    tf("WebSocket client #%u connected from %s\n", client->id(), client-
    >remoteIP().toString().c_str());
    ledcWrite(PWMLightChannel, 0);
    break;
    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);
}
}
break;

```

Disini kita mengatur konfigurasi homepage dan konfigurasi penerimaan data dari web server beserta mengatur key value ketika menerima perintah untuk lampu, kecepatan dan variabel MoveCar untuk menggerakkan mobil.

```

void setupCamera(){
    camera_config_t config;
    config.ledc_channel = LEDC_CHANNEL_0;
    config.ledc_timer = LEDC_TIMER_0;
    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;}
}

```

Disini kita mengatur konfigurasi camera ESP32-CAM dan mengirimkan data gambar dan dikirimkan pada web server local.

## **BAB IV**

### **HASIL DAN PEMBAHASAN**

Pengujian sistem merupakan proses pengeksekusian sistem perangkat keras dan lunak untuk menentukan apakah sudah sesuai dengan yang diinginkan penulis. Pengujian ini dilakukan dengan melakukan percobaan untuk melihat kesalahan yang terjadi dari setiap proses.

Dalam melakukan pengujian tahapan-tahapan yang dilakukan pertama kali adalah melakukan pengujian terhadap perangkat-perangkat inputan kemudian pengujian secara keseluruhan sistem dan perangkat keras pada alat dan halaman web kontroler dari segi spesifikasi fungsional tanpa menguji desain dan kode pada mobil, untuk mengetahui apakah fungsi-fungsi dan output sudah berjalan semestinya. Adapun tahap pengukuran dalam pengujian perangkat dan sistem alat ini sebagai berikut.

#### **4.1 Tegangan Pada Mobil**

Pengukuran tegangan pada komponen L298N dan ESP32-CAM menggunakan Avometer dengan posisi keadaan alat menyala. Berikut tabel tegangan pada mobil :

**Tabel 4. 1** Tegangan Pada Mobil

<b>Komponen</b>	<b>Tegangan</b>
L298N	15.31 V
ESP32-CAM	4.96V

##### **4.1.1 Arus Pada Mobil Pengantar**

Arus yang terukur pada tabel, terukur ketika mobil sedang keadaan menyala dan diukur menggunakan alat Avometer. Berikut Arus pada mobil pengantar :

**Tabel 4.1. 1** Arus Pada Mobil Pengantar

<b>Komponen</b>	<b>Arus</b>
IN1	2.3 A
IN2	2.3 A
IN3	2.3 A
IN4	2.3 A

#### 4.1.2 Tegangan Output Mobil

Tegangan output mobil dilakukan ketika mobil menyala dan diukur ketika mobil menjalankan beberapa perintah.

**Tabel 4.1. 2** Tegangan Output Mobil

	<b>OUT 1</b>	<b>OUT 2</b>	<b>OUT 3</b>	<b>OUT 4</b>
<b>Maju</b>	11.5 V	0.9 V	11 V	09.V
<b>Mundur</b>	0.9 V	11.5 V	0.9 V	11.5 V
<b>Belok Kanan</b>	11.74 V	0.9 V	0.9 V	11.54 V
<b>Belok Kiri</b>	0.9 V	11.6 V	11.8 V	1 V

#### 4.2 Kecepatan dan Beban

**Tabel 4. 2** Kecepatan dan Beban

<b>Kecepatan</b>	<b>Beban</b>
Lulus	Tidak Ada Beban
Lulus	Beban 150 Gram
Lulus	Beban $\frac{1}{4}$ Kg
Lulus	Beban $\frac{1}{2}$ Kg
Lulus	Beban 1 Kg

Pada tahap testing ini menggunakan batu timbangan 150 gram,  $\frac{1}{4}$  Kg,  $\frac{1}{2}$  Kg. dan 1 Kg. Tabel diatas merupakan testing ketika mobil pengantar mengangkut beban dengan beberapa batu timbangan.

#### 4.3 User Interface Web



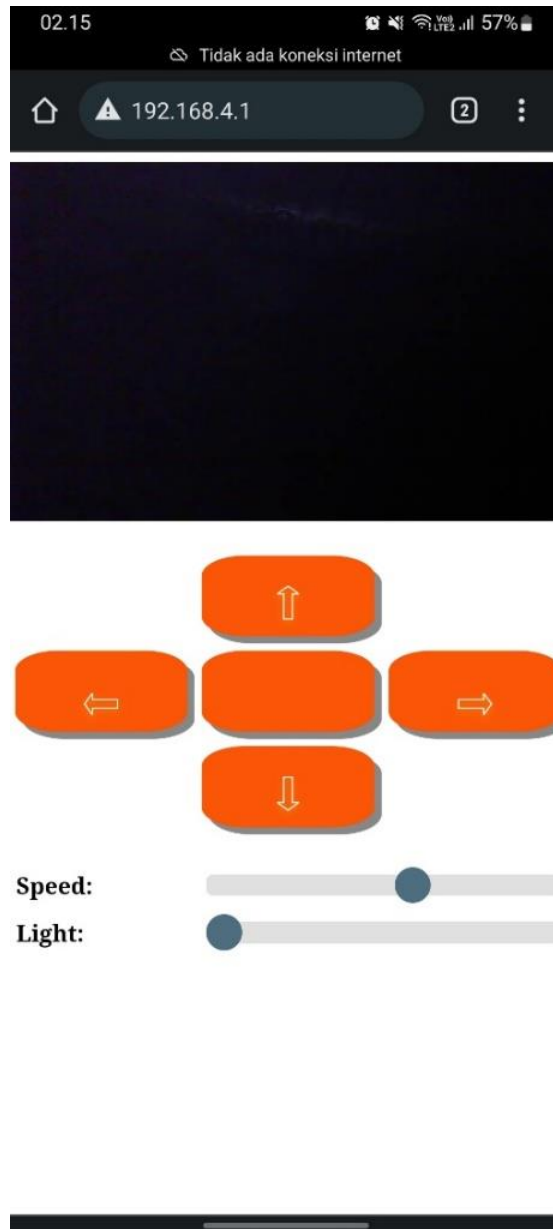
Gambar 4. 3 User Interface Web

1. Maju
2. Belok Kanan
3. Mundur
4. Belok Kiri
5. Pengatur Kecepatan
6. Pengatur Intesitas Cahaya Lampu



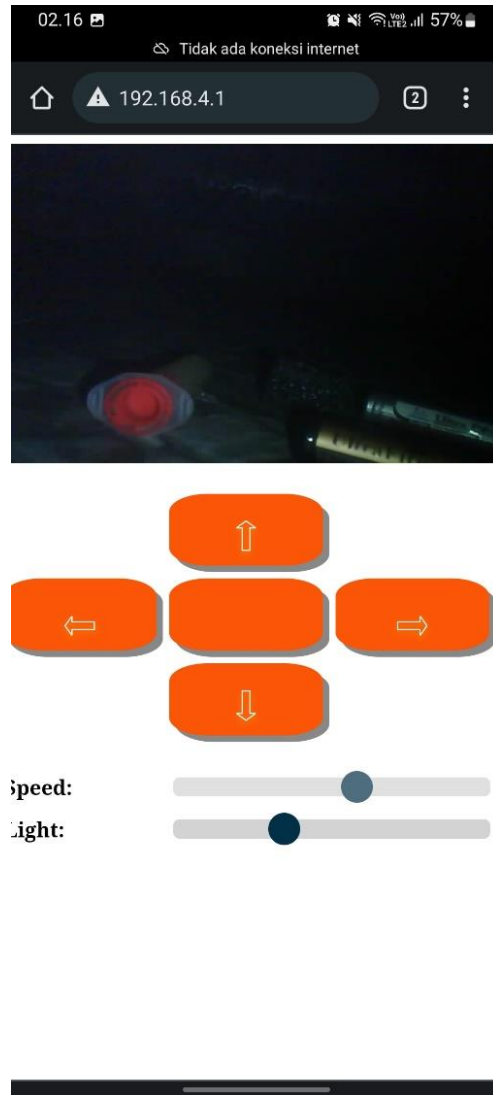
## 4.4 Intensitas Cahaya Lampu dan Kualitas Kamera

### 4.4.1 Intesitas Cahaya 0%



**Gambar 4.4. 1** Intensitas Cahaya 0%

#### 4.4.2 Intesitas Cahaya 25%



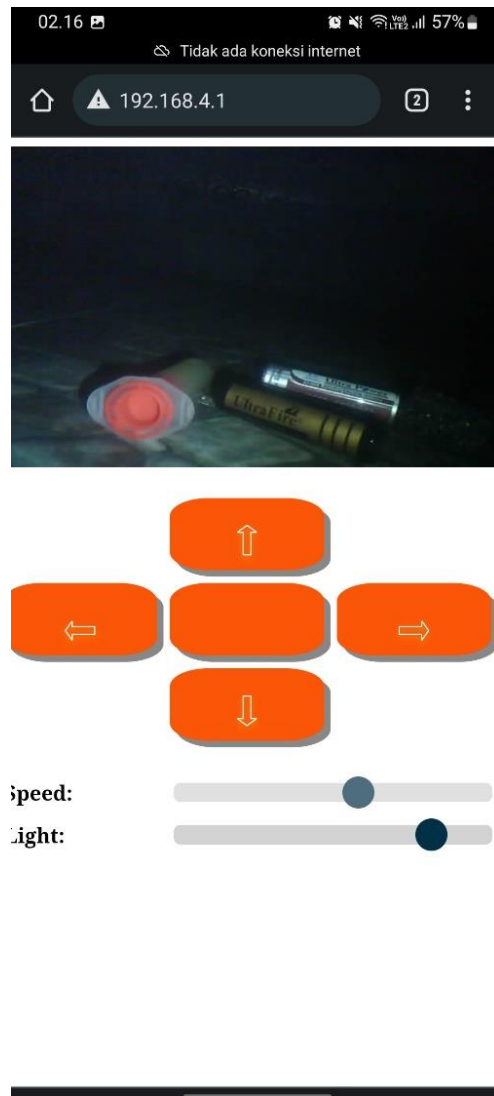
**Gambar 4.4. 2** Intensitas Cahaya 25%

#### 4.4.3 Intesitas Cahaya 50%



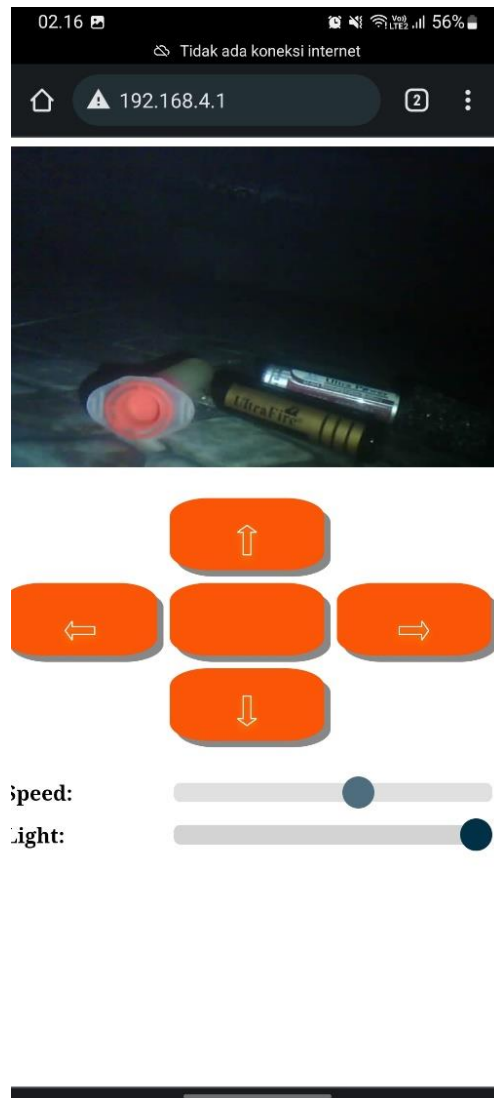
**Gambar 4.4. 3** Intensitas Cahaya 50%

#### 4.4.4 Intesitas Cahaya 75%



**Gambar 4.4. 4** Intesitas Cahaya 75%

#### 4.4.5 Intesitas Cahaya 100%



**Gambar 4.4. 5** Intensitas Cahaya 100%

#### 4.4.6 Jarak Kontrol Mobil

**Tabel 4.4.6** Jarak Kontrol Mobil

Jarak	Keterangan
1 M	Lulus
2 M	Lulus
3 M	Lulus
4 M	Lulus
5 M	Lulus
10 M	Lulus
20 M	Lulus
25 M	Lulus
30 M	Lulus

Dari beberapa percobaan kontrol jarak jauh. Mobil pengantar masih dapat di kontrol melalui web server dan di gerakkan dengan tingkat akurasi 100%.

#### 4.4.7 Kualitas Kamera

Dilakukan uji coba terhadap kualitas kamera yang terdapat perbedaan jarak antara mobil pengantar dan user yang memonitoring melalui web server.

**Tabel 4.4. 7** Kualitas Kamera

<b>Jarak</b>	<b>Delay</b>
1 M	>1 ms
2 M	>1 ms
3 M	>1 ms
4 M	1 ms
5 M	1 ms
10 M	1 detik
20 M	2 detik
25 M	2,5 detik
30 M	5 detk

Fungsi dari pengujian ini untuk mendapatkan kualitas kamera dari mobil pengantar di luar ruangan. Hasil dari uji coba ini di tampilkan dalam tabel diatas, dari jarak 1 Meter sampai dengan 30 Meter kualitas yang di terima terdapat delay > 1 Miliseconds sampai dengan 5 detik dengan jarak yang berbeda.

## **BAB V**

### **KESIMPULAN DAN SARAN**

#### **5.1 Kesimpulan**

Berdasarkan hasil penulisan yang telah dilakukan ada beberapa hal yang diperoleh kesimpulan sebagai berikut ini :

1. Alat dirancang dan dibuat menggunakan Mikrokontroler ESP32-CAM sebagai komponen utama, dan dilengkapi dengan beberapa komponen pendukung lainnya seperti Motor DC sebagai penggerak mobil, Motor Driver L298N sebagai pengatur output Motor DC dan Web Server sebagai input user interface antara perangkat dan user.
2. Hasil pergerakan mobil dan penangkapan gambar dari ESP32-CAM yang digunakan memiliki nilai masing-masing dan akan bergerak sesuai dengan perintah pada tombol halaman web server lokal. Pengujian system robot secara keseluruhan menunjukkan bahwa alat dapat berjalan sesuai dengan tampilan kontroler.
3. Hasil tes kecepatan mobil yang dilakukan untuk mencapai jarak tes adalah 30 m per menit dengan kecepatan 100 rpm atau kecepatan maksimal.

#### **5.2 Saran**

1. Untuk hasil yang lebih maksimal dapat menggunakan ESP32-CAM dengan versi Wroom32-Module yang kualitasnya lebih tinggi, sehingga dapat beroperasi lebih akurat dan tampilan kamera tidak delay.
2. Sebaiknya robot menggunakan Motor DC yang mempunyai kekuatan lebih besar sehingga mobil dapat bergerak dengan lebih cepat ketika menggerakkan mobil.



## DAFTAR PUSTAKA

- Academy, Binar. 2022. *Prototype : Pengertian, Tujuan, dan Manfaatnya*. Agustus 19. Accessed Oktober 31, 2022. <https://www.binaracademy.com/blog/pengertian-prototype-dan-tujuannya>.
- Adafruit. n.d. *DC Gearbox Motor - "TT Motor" - 200RPM - 3 to 6VDC*. Accessed Oktober 24, 2022. <https://www.adafruit.com/product/3777>.
- Benefita. 2022. *Yakin Anda Sudah Menggunakan Web Server Terbaik ?* Mei 31. Accessed Oktober 31, 2022. [https://www.niagahoster.co.id/blog/web-server-adalah/#Cara\\_Kerja\\_Web\\_Server](https://www.niagahoster.co.id/blog/web-server-adalah/#Cara_Kerja_Web_Server).
- Dewaweb. 2022. *Pengertian Web Server, Fungsi, Contoh dan Cara Kerjanya*. September 28. Accessed Oktober 22, 2022. <https://www.dewaweb.com/blog/apa-itu-web-server/>.
- Dicoding. 2021. *Apa itu Web Server dan Fungsinya ?* Januari 27. Accessed Oktober 31, 2022. <https://www.dicoding.com/blog/apa-itu-web-server-dan-fungsinya/>.
- . 2020. *Contoh Coding HTML Website Dalam 15 Menit*. Desember 9. Accessed Oktober 31, 2022. <https://www.dicoding.com/blog/contoh-coding-html-website-dalam-15-menit/>.
- geeksforgeeks.org. 2022. *C Language Introduction*. May 31. Accessed Oktober 31, 2022. <https://www.geeksforgeeks.org/c-language-set-1-introduction/>.
- Indobot. 2022. *Mengenal ESP32-Cam dan bagaimana cara menggunakannya*. Januari 14. Accessed Oktober 24, 2022. <https://indobot.co.id/blog/mengenal-esp32-cam-dan-bagaimana-cara-menggunakannya/>.
- Kadir, Abdul. 2017. *From Zero to a Pro Arduino Panduan mempelajari aneka proyek berbasis Mikrokontroler*. Yogyakarta: Penerbit Andi.

- Napizahni, Mike. 2022. *Apa Itu Localhost? Pengertian, Jenis, dan Kegunaannya*. Februari 20. Accessed Oktober 31, 2022. <https://www.dewaweb.com/blog/apa-itu-localhost/>.
- Nayoan, Aldwin. 2022. *Apa Itu CSS? Pengertian, Fungsi, dan Contohnya*. Juni 10. Accessed Oktober 31, 2022. <https://www.niagahoster.co.id/blog/pengertian-css/>.
- NDE CARGO. 2021. *Mengenal Istilah Towing Car dan Carrier Car*. Februari 19. Accessed Oktober 31, 2022. <https://ndecargo.co.id/mengenal-istilah-towing-car-dan-carrier-car-dalam-pengiriman-mobil/>.
- nyebarilmu.com. 2017. *Tutorial Arduino mengakses driver motor L298N*. Agustus 27. Accessed Oktober 24, 2022. <https://www.nyebarilmu.com/tutorial-arduino-mengakses-driver-motor-l298n/>.
- Rheny, Sylvia. 2022. *Mengenal apa itu JavaScript mulai dari pengertian hingga sejarahnya*. September 29. Accessed Oktober 31, 2022. <https://www.ekrut.com/media/sejarah-javascript>.
- Setiawan, Rony. 2021. *Apa Itu Prototype? Kenapa Itu Penting?* Agustus 11. Accessed Oktober 31, 2022. <https://www.dicoding.com/blog/apa-itu-prototype-kenapa-itu-penting/>.
- Ultrafire. n.d. *UltraFire 3000mAh 3.7V 18650 Rechargeable Lithium Batteries Without Protection Board(2PCS)*. Accessed Oktober 31, 2022. <https://www.ultrafire.com/products/ultrafire-18650-3-7v-3000mah-rechargeable-lithium-batteries-without-protection-2pcs>.

## LAMPIRAN - LAMPIRAN

### 1. Source Code Hardware

```
#include "esp_camera.h"
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h> //Ini library untuk web server
#include <ESPAsyncWebServer.h> // Ini library untuk web server
#include <iostream>
#include <sstream>

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

std::vector<MOTOR_PINS> motorPins =
{
    { 12, 13, 15}, //RIGHT_MOTOR Pins (EnA, IN1, IN2)
    { 12, 14, 2}, //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 = "bagus";
const char* password = "tugasakhir";

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

// Code WEB SERVER
const char* htmlHomePage PROGMEM = R"HTMLHOMEPAGE(

<!DOCTYPE html>
)HTMLHOMEPAGE";
// AKHIR CODE WEB SERVER

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);
            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);
    }
    }
    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_0;
    config.ledc_timer = LEDC_TIMER_0;
    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()
{
    //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());
}
```

## 2. Source Code pada Web server

```
<!DOCTYPE html>
<html>
<head>
<meta name="viewport" content="width=device-width, initial-scale=1,
maximum-scale=1, user-scalable=no">
<style>
.arrows {
font-size:40px;
color:#fff;
text-shadow: 0px 2px 5px #eea60c;
}
td.button {
background-color:#fb5607;
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: #023047;
  cursor: pointer;
}

.slider::-moz-range-thumb {
  width: 25px;
  height: 25px;
  border-radius: 50%;
  background: #023047;
  cursor: pointer;
}

</style>

</head>
<body class="noselect" align="center" style="background-color:white">

  <!--h2 style="color: teal;text-align:center;">Wi-Fi Camera &#128663;
Control</h2-->

  <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>
</table>

```

```

<script>
    var websocketCameraUrl = "ws://\V" + window.location.hostname +
"/Camera";
    var websocketCarInputUrl = "ws://\V" + 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)
        {
            var speedButton = document.getElementById("Speed");
            sendButtonInput("Speed", speedButton.value);
            var lightButton = document.getElementById("Light");
            sendButtonInput("Light", lightButton.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("touchend",
function(event){
    event.preventDefault()
  });
</script>
</body>
</html>
```