

DAFTAR PUSTAKA

- [1] M. Aziz, Y. Marcellino, I. Agnita Rizki, S. Anwar Ikhwanuddin, and J. Welman Simatupang, “STUDI ANALISIS PERKEMBANGAN TEKNOLOGI DAN DUKUNGAN PEMERINTAH INDONESIA TERKAIT MOBIL LISTRIK.”
- [2] B. P. Resosudarmo *et al.*, “‘Greenhouse Gas Emission in Indonesia: The Significance of Fossil Fuel Combustion’, in GREENHOUSE GAS EMISSION IN INDONESIA: THE SIGNIFICANCE OF FOSSIL FUEL COMBUSTION,” 2009.
- [3] F. Un-Noor, S. Padmanaban, L. Mihet-Popa, M. N. Mollah, and E. Hossain, “A comprehensive study of key electric vehicle (EV) components, technologies, challenges, impacts, and future direction of development,” *Energies*, vol. 10, no. 8. MDPI AG, 2017. doi: 10.3390/en10081217.
- [4] Y. Widiyanto, A. Budijanto, and B. Widjanarko, “KENDALI MOBILE ROBOT DENGAN SUARA MENGGUNAKAN ANDROID SMART PHONE,” *Seminar Nasional Sistem Informasi*, vol. 9, 2018, [Online]. Available: <http://www.martyncurrey.com>
- [5] A. Yusika, A. Rofiq, and A. T. Ramadhani, “PERANCANGAN MOBILE REMOTE CONTROL MENGGUNAKAN KONTROL BLUETOOTH ARDUINO UNO”.
- [6] M. F. Wicaksono and M. D. Rahmatya, “Implementasi Arduino dan ESP32 CAM untuk Smart Home,” *Jurnal Teknologi dan Informasi*, doi: 10.34010/jati.v10i1.
- [7] Surawijaya Surahman, “Aplikasi MobileDriver Online Berbasis Android Untuk Perusahaan Rental Kendaraan,” *Ultima Info Sys*, vol. 8, no. 1, Jul. 2017.
- [8] Andi Setiawan and A. Irma Purnamasari, “Pengembangan Smart Home Dengan Microcontrollers ESP32 Dan MC-38 Door Magnetic Switch Sensor Berbasis Internet of Things (IoT) Untuk Meningkatkan Deteksi Dini Keamanan Perumahan,” *Jurnal Resti*, vol. 3, no. 3, pp. 451–457, 2017.
- [9] A. Ramschie *et al.*, “Pemanfaatan ESP32 Pada Sistem Keamanan Rumah Tinggal Berbasis IoT,” 2021.
- [10] F. Shoufika Hilyana, H. Mirza, and B. Gunawan, “APLIKASI SISTEM KENDALI PERGERAKAN KAMERA BERBASIS ANDROID,” *Jurnal SIMETRIS*, vol. 11, no. 1, 2020.
- [11] P. Rahmiati, G. Firdaus, and N. Fathorrahman, “Implementasi Sistem Bluetooth menggunakan Android dan Arduino untuk Kendali Peralatan Elektronik,” 2014.
- [12] S. Oleh and Y. Fauzan, “KOTAK PENERIMA PAKET BERBASIS IoT MENGGUNAKAN MODUL ESP32-CAM.”

- [13] P. Arri Ape Pane Basabilik Prodi Fisika, J. Fisika, and U. Tanjungpura, "RANCANG BANGUN SISTEM PEMANTAU KEDATANGAN TAMU BERBASIS INTERNET OF THINGS (IOT)," vol. 9, no. 2, pp. 110–116, 2021.
- [14] D. Setiawan, H. Jaya, S. Nurarif, T. Syahputra, M. Syahril Syafnur, and S. Triguna Dharma, "IMPLEMENTASI ESP32-CAM DAN BLYNK PADA WIFI DOOR LOCK SYSTEM MENGGUNAKAN TEKNIK DUPLEX," 2022. [Online]. Available: <http://jurnal.goretanpena.com/index.php/JSSR>
- [15] B. Yanto, B. Basorudin, S. Anwar, A. Lubis, and K. Karmi, "Smart Home Monitoring Pintu Rumah Dengan Identifikasi Wajah Menerapkan Camera ESP32 Berbasis IoT," *Jurnal Sisfokom (Sistem Informasi dan Komputer)*, vol. 11, no. 1, pp. 53–59, Mar. 2022, doi: 10.32736/sisfokom.v11i1.1180.
- [16] T. Suwanto, T. Imbang, A. Samadhi, K. Kunci, : Sepeda, and A. Uno, "APLIKASI KONTROL SEPEDA MOTOR MENGGUNAKAN BLUETOOTH PADA SMARTPHONE ANDROID," *Jurnal Ilmiah Widya Teknik*, vol. 15, p. 2016.
- [17] Supriyadi, "PEMANFAATAN BLUETOOTH ANDROID SEBAGAI PENGENDALI PENGAMAN TAMBAHAN PADA SISTEM KEAMANAN SEPEDA MOTOR."
- [18] HEN NANDA PERDANA KRISTYABUDI, "SISTEM KENDALI REMOTE CONTROL MINI- BLIMP MENGGUNAKAN ANDROID SMARTPHONE DENGAN KOMUNIKASI BLUETOOTH BERBASIS MIKROKONTROLER," Yogyakarta, 2016.
- [19] D. Setiawan, J. Yos Sudarso Km, K. Kunci, and A. Uno, "SISTEM KONTROL MOTOR DC MENGGUNAKAN PWM ARDUINO BERBASIS ANDROID SYSTEM," *Jurnal Sains, Teknologi dan Industri*, vol. 15, no. 1, pp. 7–14, 2017.
- [20] Y. S. Handayani and Y. Mardiana, "KENDALI ROBOT BLUETOOTH DENGAN SMARTPHONE ANDROID BERBASIS ARDUINO UNO," *ILKOM Jurnal Ilmiah*, vol. 10, no. 3, pp. 331–337, Dec. 2018, doi: 10.33096/ilkom.v10i3.363.331-337.
- [21] S. Kurniadi, "PERANCANGAN ALAT PENGONTROL KECEPATAN MOTOR MENGGUNAKAN ARDUINO MEGA 2560 BERBASIS ANDROID MELALUI BLUETOOTH," Subang.
- [22] S. Muliarni and I. Husnaini, "Aplikasi Pengontrolan Lampu Dan Motor DC Berbasis Arduino," *JURNAL TEKNIK ELEKTRO DAN VOKASIONAL*, vol. 6, no. 2, 2020, [Online]. Available: <http://ejournal.unp.ac.id/index.php/jtev/index>
- [23] Faqih Muhammad, "RANCANG BANGUN APLIKASI MOBIL REMOTE CONTROL PEMANTAU BERBASIS ANDROID PADA MIKROKONTROLER ARDUINO," *Universitas Tanjungpura*, vol. 1, no. 1, 2017.

- [24] “Handson Technology User Guide L298N Dual H-Bridge Motor Driver.” [Online]. Available: www.handsontec.com
- [25] “Everything You Need To Know About DC Motors | RS.” <https://ie.rs-online.com/web/generalDisplay.html?id=ideas-and-advice/dc-motors-guide> (accessed Jan. 26, 2022).
- [26] G. S. Miratunnisa and A. H. S. Budi, “Traffic Light Monitoring System based on NodeMCU using Internet of Things,” *IOP Conf Ser Mater Sci Eng*, vol. 384, p. 012024, Jul. 2018, doi: 10.1088/1757-899X/384/1/012024.
- [27] A. Imran and M. Rasul, “PENGEMBANGAN TEMPAT SAMPAH PINTAR MENGGUNAKAN ESP32,” 2020.
- [28] D. Atmajaya, N. Kurniati, Y. Salim, W. Astuti, and F. I. Komputer, “Sistem Kontrol Timbangan Sampah Non Organik Berbasis Load Cell dan ESP32,” 2018.
- [29] G. S. Miratunnisa and A. H. S. Budi, “Traffic Light Monitoring System based on NodeMCU using Internet of Things,” *IOP Conf Ser Mater Sci Eng*, vol. 384, no. 1, 2018, doi: 10.1088/1757-899X/384/1/012024.
- [30] D. Aziz, “Webserver Based Smart Monitoring System Using ESP8266 Node MCU Module,” *Int J Sci Eng Res*, vol. 9, p. 801, Jun. 2018.
- [31] “Advanced Information Preliminary Datasheet OV2640 Color CMOS UXGA (2.0 MegaPixel) CAMERACHIP TM Sensor O mni ision® with OmniPixel2 TM TechnologyGeneral Description,” 2007.
- [32] “PEMBANGUNAN APLIKASI KONVERSI DATABASE KE XML BERBASIS WEB - PDF Download Gratis”.
- [33] K. D. Lee and S. Hubbard, “Undergraduate Topics in Computer Science Data Structures and Algorithms with Python.” [Online]. Available: <http://www.springer.com/series/7592>

LAMPIRAN

Lampiran 1 : Listing Program Main_Activity.java

```
package com.example.blue3;

import androidx.annotation.NonNull; import androidx.annotation.Nullable;
import androidx.appcompat.app.AppCompatActivity;import android.Manifest;
import android.bluetooth.BluetoothAdapter; import android.bluetooth.BluetoothDevice;
import android.content.Intent;
import android.os.Build; import android.os.Bundle; import android.util.Log; import
android.view.View; import android.view.ViewGroup;
import android.webkit.WebChromeClient;import android.webkit.WebView;
import android.webkit.WebViewClient;import android.widget.AdapterView; import
android.widget.Button;
import android.widget.ListView; import android.widget.ProgressBar;import
android.widget.TextView;

import java.util.ArrayList;import java.util.List; import java.util.Set; import
java.util.UUID;

public class MainActivity extends AppCompatActivity { private static final String
TAG = "MainActivityTAG";
    private static final UUID uuid = UUID.fromString("00001101-0000-1000- 8000-
00805F9B34FB");
    private BluetoothAdapter bluetoothAdapter;private BluetoothDevice selectedDevices;
    public static final String DEV_UUID = "com.example.blue3.DEV_UUID";public static
final String DEV_DEVI = "com.example.blue3.DEV_DEVI";ProgressBar progressBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState); setContentView(R.layout.activity_main);

        bluetoothAdapter = BluetoothAdapter.getDefaultAdapter(); if(bluetoothAdapter ==
null || !bluetoothAdapter.isEnabled()) {
            assert bluetoothAdapter != null;bluetoothAdapter.enable();
        }
        checkBTPermissions();
        Button search = findViewById(R.id.search);ListView lvl = findViewById(R.id.lv1);
```

```

        search.setOnClickListener(v-> {
            Set<BluetoothDevice> pairedDevices =
            bluetoothAdapter.getBondedDevices();
            List<BluetoothDevice> listDevices = new
            ArrayList<>(pairedDevices);

            ArrayAdapter<BluetoothDevice> arrayAdapter = new
            ArrayAdapter<BluetoothDevice>(this, android.R.layout.simple_list_item_2,
            android.R.id.text1, listDevices) {
                @NonNull
                @Override
                public View getView(int position, @Nullable View
            convertView, @NonNull ViewGroup parent) {
                    View view = super.getView(position, convertView,
            parent);
                    TextView text1 =
            view.findViewById(android.R.id.text1);
                    TextView text2 =
            view.findViewById(android.R.id.text2);

                    text1.setText(listDevices.get(position).getName());
            text2.setText(listDevices.get(position).getAddress());
                    return view;
                }
            };

            lvl.setAdapter(arrayAdapter);
            lvl.setOnItemClickListener((parent, view, position, id) -> {
                selectedDevices = (BluetoothDevice)
            lvl.getItemAtPosition(position);
                Intent intent = new Intent(this, Controller.class);
                intent.putExtra(DEV_UUID, uuid.toString());
                intent.putExtra(DEV_DEVI, selectedDevices.getAddress());
                startActivity(intent);
            });
        });

        private void checkBTPermissions() {
            if (Build.VERSION.SDK_INT > Build.VERSION_CODES.LOLLIPOP) {
                int permissionCheck;
                if (android.os.Build.VERSION.SDK_INT >=
            android.os.Build.VERSION_CODES.M) {
                    permissionCheck =
            this.checkSelfPermission("Manifest.permission.ACCESS_FINE_LOCATION");
                    permissionCheck +=
            this.checkSelfPermission("Manifest.permission.ACCESS_COARSE_LOCATION");
                    if (permissionCheck != 0) {
                        this.requestPermissions(new
            String[]{Manifest.permission.ACCESS_FINE_LOCATION,
            Manifest.permission.ACCESS_COARSE_LOCATION}, 1001);
                    }
                } else {
                    Log.d(TAG, "checkBTPermissions: No need to check permissions.
            SDK version < LOLLIPOP.");
                }
            }
        }
    }
}

```

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();
  if (get_percentage() <= 30.0) {
    Serial.println("Sangat LOW");
    relay_state(true);
    send_again = false;
  } else if (get_percentage() >= 100.00) {
    relay_state(false);
    send_again = true;
  }
  if (send_again) {
    float amps = round2(ACS.mA_DC() * 1000.0);
    total = round2(total - amps);
    float volt = round2(get_voltage());
    float get_state = get_percentage();
    String to_send = String(amps) + "/" + String(volt) + "/" +
String(get_state) + "/" + String(total);
    snprintf(msg, BUFF_SIZE, "%s", to_send.c_str());
    client.publish("Rafli/BMS/1318", msg);
    unsigned long wait = millis();
    while ((unsigned long) millis() - wait < 1000);
  }
}
```

Lampiran 2 : Listing Program ESP 32-CAM

```
#include "esp_camera.h"

#include <WiFi.h>

#define CAMERA_MODEL_WROVER_KIT // Has PSRAM

#include "camera_pins.h"

const char* ssid = "404";

const char* password = "Sukaria2b";

void startCameraServer();

void setup() {

  Serial.begin(115200);

  Serial.setDebugOutput(true);

  Serial.println();

  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;
if(psramFound()){
    config.frame_size = FRAMESIZE_UXGA;
    config.jpeg_quality = 10;
    config.fb_count = 2;
} else {
    config.frame_size = FRAMESIZE_SVGA;
    config.jpeg_quality = 12;
    config.fb_count = 1;
}
#ifdef CAMERA_MODEL_ESP_EYE
pinMode(13, INPUT_PULLUP);
pinMode(14, INPUT_PULLUP);
#endif
esp_err_t err = esp_camera_init(&config);
if (err != ESP_OK) {
    Serial.printf("Camera init failed with error 0x%x", err);
    return;
}
sensor_t * s = esp_camera_sensor_get();
// initial sensors are flipped vertically and colors are a bit saturated
if (s->id.PID == OV3660_PID) {
    s->set_vflip(s, 1); // flip it back
    s->set_brightness(s, 1); // up the brightness just a bit
    s->set_saturation(s, -2); // lower the saturation
}
```



```
s->set_framesize(s, FRAMESIZE_QVGA);

#if defined(CAMERA_MODEL_M5STACK_WIDE) || defined(CAMERA_MODEL_M5STACK_ESP32CAM) s-
>set_vflip(s, 1);
s->set_hmirror(s, 1);
#endif

WiFi.begin(ssid, password);

while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
}
Serial.println("");
Serial.println("WiFi connected");

startCameraServer();

Serial.print("Camera Ready! Use 'http://");
Serial.print(WiFi.localIP());
Serial.println("' to connect");
}

void loop() {
  // put your main code here, to run repeatedly:
  delay(10000);
}
```

Lampiran 3 : Listing Program Kontrol Motor Bluetooth

```
#define dira 2
#define dirb 4
#define pwma 3
#define pwmb 5
#define vccbt 9
#define gndbt 10
#define txbt 11
#define rxbt 12

String data;

#include <SoftwareSerial.h>

SoftwareSerial Serialbt (txbt, rxbt);

void motor(int a, int b) {
  if (a >= 0) {
    digitalWrite (dira, 0);
    analogWrite (pwma, a);
  }
  else if (a < 0) {
    digitalWrite (dira, 1);
    analogWrite (pwma, a + 255);
  }
  if (b >= 0) {
    digitalWrite (dirb, 0);
    analogWrite (pwmb, b);
  }
}
```

```
else if (b < 0) {  
    digitalWrite (dirb, 1);  
    analogWrite (pwmb, b + 255);  
}  
}  
void setup() {  
    // put your setup code here, to run once:  
    Serial.begin(9600);  
    Serialbt.begin(9600);  
    pinMode (dira, OUTPUT);  
    pinMode (dirb, OUTPUT);  
    pinMode (vccbt, OUTPUT);  
    pinMode (gndbt, OUTPUT);  
  
    digitalWrite (vccbt, 1);  
    digitalWrite (gndbt, 0);  
}  
  
void loop() {  
    // put your main code here, to run repeatedly:  
    while (Serialbt.available() > 0) {  
        delay(10);  
        char c = Serialbt.read();  
        data += c;  
    }  
    if (data.length() > 0) {  
        Serial.println(data);  
        if (data == "S") {  
            motor(0, 0);  
        }  
    }  
}
```

```
else if (data == "F") {
    motor(255, 255);
}
else if (data == "G") {
    motor(100, 255);
}
else if (data == "I") {
    motor(255, 100);
}
else if (data == "L") {
    motor(0, 255);
}
else if (data == "R") {
    motor(255, 0);
}
else if (data == "B") {
    motor(-255, -255);
}
else if (data == "H") {
    motor(-100, -255);
}
else if (data == "J") {
    motor(-255, -100);
}
    else if (data == "H") {
    motor(-100, -255);
}
else if (data == "J") {
    motor(-255, -100);
}
data = "";
```