

LAMPIRAN

LISTING KODE PROGRAM

```
1. #include <Arduino.h>
2. #include <Wire.h>
3. #include <PCF8574.h>
4. #include <LiquidCrystal_I2C.h>
5. #include <WiFi.h>
6. #include <WiFiClientSecure.h>
7. #include <UniversalTelegramBot.h>
8. #include <HX711.h>
9. #include "soc/soc.h"
10. #include "soc/rtc_cntl_reg.h"
11. #include "esp_camera.h"
12.
13. // Pengaturan I2C
14. PCF8574 keypad(0x20); // Keypad
15. LiquidCrystal_I2C lcd(0x27, 16, 2); // LCD
16.
17. const byte ROWS = 4; // Four rows
18. const byte COLS = 4; // Four columns
19.
20. char keys[ROWS][COLS] = {
21.   { '1', '2', '3', 'A' },
22.   { '4', '5', '6', 'B' },
23.   { '7', '8', '9', 'C' },
24.   { '*', '0', '#', 'D' }
25. };
26.
27. byte colPins[COLS] = { 0, 1, 2, 3 }; // Pinout Kolom
28. byte rowPins[ROWS] = { 4, 5, 6, 7 }; // Pinout Baris
29.
30. // Kredensial Wifi
31. const char* ssid = "C15";
32. const char* password = "lsampai8";
33.
34. // Kredensial Telegram
35. const char* botToken =
    "7399521376:AAEYHOAHfCI71YiUwHc3Srfj5UqK2v1wO8s";
36. const String chat_id = "412434642";
37.
38. WiFiClientSecure client;
39. UniversalTelegramBot bot(botToken, client);
40. bool wifiConnectedMessageSent = false;
41.
42. // State machine states
43. enum State { MAIN_MENU,
44.             BAYAR_PAKET,
45.             WAIT_FOR_COMPLETION };
```

```

46. State currentState = MAIN_MENU;
47.
48. // Konfigurasi Relay pins
49. const int relayPintu1Pin = 0; // GPIO2
50. const int relayPintu2Pin = 2; // GPIO3
51.
52. // Konfigurasi Load Cell
53. #define DOUT_PIN 12
54. #define SCK_PIN 13
55. HX711 scale;
56. float calibration_factor = 125.27; // Faktor kalibrasi
57.
58. const int reedSwitchPin = 14; // Pin untuk reed switch
59.
60. // Definisi Variabel untuk Menyimpan Status dan Jumlah
61. int pintu1 = 0;
62. int pintu2 = 0;
63. int paketCount = 0;
64. String pintuNomor = ""; // Inisialisasi variabel pintuNomor
65.
66. // Menggunakan definisi kamera WROVER KIT
67. #define CAMERA_MODEL_WROVER_KIT
68. #include "camera_pins.h"
69.
70. // Variabel Pewaktuan
71. unsigned long lastLCDUpdate = 0;
72. const unsigned long lcdUpdateInterval = 1000; // Update LCD
    every 1 seconds
73. bool lcdMessageComplete = false;
74. unsigned long lastWeightCheck = 0;
75. unsigned long lastTelegramCheck = 0;
76. const unsigned long weightCheckInterval = 3000; // 3 detik
77. const unsigned long telegramCheckInterval = 5000; // 5 detik
78. String bayarInput = "";
79.
80. void setup() {
81.     WRITE_PERI_REG(RTC_CNTL_BROWN_OUT_REG, 0); // Disable brownout
    detector
82.     configInitCamera();
83.
84.     Serial.begin(115200);
85.     Wire.begin(32, 33); // Initialize I2C
86.     keypad.begin();
87.
88.     lcd.begin(16, 2);
89.     lcd.backlight();
90.     lcd.setCursor(0, 0);
91.
92.     for (int i = 0; i < ROWS; i++) {

```

```

93.     keypad.write(rowPins[i], HIGH);
94.   }
95.
96.   for (int i = 0; i < COLS; i++) {
97.     keypad.write(colPins[i], HIGH);
98.   }
99.
100.  pinMode(relayPintu1Pin, OUTPUT);
101.  pinMode(relayPintu2Pin, OUTPUT);
102.  digitalWrite(relayPintu1Pin, HIGH);
103.  digitalWrite(relayPintu2Pin, HIGH);
104.
105.  pinMode(reedSwitchPin, INPUT_PULLUP);
106.
107.  scale.begin(DOUT_PIN, SCK_PIN);
108.  scale.set_scale(calibration_factor);
109.  scale.tare(); // Mengatur berat awal menjadi 0
110.
111.  // Connect to WiFi
112.  client.setInsecure();
113.  WiFi.begin(ssid, password);
114.  while (WiFi.status() != WL_CONNECTED) {
115.    delay(500);
116.    Serial.print(".");
117.    lcd.setCursor(0, 0);
118.    lcd.print("MENGHUBUNGKAN ");
119.    lcd.setCursor(0, 1);
120.    lcd.print("WIFI ...");
121.  }
122.
123.  Serial.println("WIFI TERHUBUNG");
124.  lcd.clear();
125.  lcd.setCursor(0, 0);
126.  lcd.print("WIFI TERHUBUNG");
127.  delay(2000);
128.
129.  showMainMenu();
130. }
131.
132. void loop() {
133.   bool isDoorClosed = digitalRead(reedSwitchPin) == LOW;
134.   static bool wasDoorClosed = isDoorClosed;
135.
136.   if (isDoorClosed) {
137.     if (!wasDoorClosed) {
138.       lcd.clear();
139.       showMainMenu();
140.       lcdMessageComplete = false;
141.     }

```

```

142.
143.     static unsigned long lastKeyPressTime = 0;
144.     char key = getKeyFromKeypad();
145.     if (key && (millis() - lastKeyPressTime > 50)) {
146.         lastKeyPressTime = millis();
147.         handleKeypadInput(key);
148.     }
149. } else {
150.     if (wasDoorClosed) {
151.         lcd.clear();
152.         lcd.print("PINTU TERBUKA!");
153.         if (currentState == WAIT_FOR_COMPLETION) {
154.             // Jika pintu terbuka dalam state WAIT_FOR_COMPLETION
155.             bot.sendMessage(chat_id, "Pintu paket dibuka,
konfirmasi dibatalkan");
156.             currentState = MAIN_MENU; // Kembali ke menu utama
157.         }
158.     }
159. }
160.
161.     wasDoorClosed = isDoorClosed;
162.
163.     if (lcdMessageComplete && millis() - lastLCDUpdate >=
lcdUpdateInterval) {
164.         lcd.clear();
165.         showMainMenu();
166.         lcdMessageComplete = false;
167.     }
168.
169.     if (millis() - lastTelegramCheck >= telegramCheckInterval) {
170.         lastTelegramCheck = millis();
171.         checkTelegramMessages();
172.     }
173.
174.     if (WiFi.status() != WL_CONNECTED) {
175.         reconnectWiFi();
176.     }
177. }
178.
179. // Fungsi untuk mendapatkan input dari keypad
180. char getKeyFromKeypad() {
181.     for (int row = 0; row < ROWS; row++) {
182.         keypad.write(rowPins[row], LOW); // Pull the row pin low
183.         for (int col = 0; col < COLS; col++) {
184.             if (keypad.read(colPins[col]) == LOW) {
185.                 while (keypad.read(colPins[col]) == LOW)
186.                     ;
187.                 keypad.write(rowPins[row], HIGH);
188.                 return keys[row][col];

```

```

189.     }
190.     }
191.     keypad.write(rowPins[row], HIGH);
192.     }
193.     return 0; // No key pressed
194. }
195.
196. void reconnectWiFi() {
197.     Serial.println("WIFI TERPUTUS, MENGHUBUNGAN ULANG...");
198.     lcd.clear();
199.     lcd.setCursor(0, 0);
200.     lcd.print("MENGHUBUNGAN ");
201.     lcd.setCursor(0, 1);
202.     lcd.print("ULANG WIFI ...");
203.     WiFi.begin(ssid, password);
204.     while (WiFi.status() != WL_CONNECTED) {
205.         delay(500);
206.         Serial.print(".");
207.     }
208.     lcd.clear();
209.     lcd.setCursor(0, 0);
210.     lcd.print("WIFI TERHUBUNG");
211.     showMainMenu();
212. }
213.
214. void handleKeypadInput(char key) {
215.     Serial.print("Tombol Ditekan: ");
216.     Serial.println(key);
217.
218.     if (digitalRead(reedSwitchPin) != LOW) { // Jika pintu
        terbuka, abaikan input keypad
219.         return;
220.     }
221.
222.     switch (currentState) {
223.         case MAIN_MENU:
224.             lcd.clear();
225.             lcd.print("MENU UTAMA");
226.             lcd.setCursor(0, 1);
227.             lcd.print("TEKAN KEYPAD");
228.             if (key) {
229.                 currentState = BAYAR_PAKET;
230.                 bayarInput = ""; // Clear the input buffer
231.                 lcd.clear();
232.                 lcd.print("MASUKKAN BIAYA");
233.                 lcd.setCursor(0, 1);
234.                 lcd.print("PAKET:");
235.             }
236.             break;

```



```

237.
238.     case BAYAR_PAKET:
239.         if (key >= '0' && key <= '9') {
240.             bayarInput += key;
241.             lcd.setCursor(6, 1);
242.             lcd.print(bayarInput);
243.         } else if (key == 'C') { // Use 'C' to clear the input
244.             bayarInput = "";
245.             lcd.clear();
246.             lcd.print("MASUKKAN BIAYA");
247.             lcd.setCursor(0, 1);
248.             lcd.print("PAKET:");
249.         } else if (key == 'D') { // Use 'D' as the enter key
250.             int bayarValue = bayarInput.toInt();
251.             float currentWeight = read_load_cell(); // Read weight
252.             String message = "Ada Paket Baru dengan biaya: " +
bayarInput + " dan berat: " + String(currentWeight, 1) + " gram";
253.             if (bayarValue == 0) {
254.                 bot.sendMessage(chat_id, message);
255.                 lcd.clear();
256.                 lcd.print("PAKET DITERIMA");
257.                 lcd.setCursor(0, 1);
258.                 lcd.print("TERIMA KASIH");
259.                 paketCount++;
260.                 String jumlah = "Paket diterima!";
261.                 bot.sendMessage(chat_id, jumlah);
262.                 lcd.clear();
263.                 showMainMenu();
264.                 currentState = MAIN_MENU;
265.             } else {
266.                 String message = "Ada Paket Baru dengan biaya: " +
bayarInput + " dan berat: " + String(currentWeight, 0) + " gram";
267.                 bot.sendMessage(chat_id, message);
268.                 sendPhotoTelegram(); // Send photo before message
269.                 lcd.clear();
270.                 lcd.print("MENUNGGU RESPON");
271.                 lcd.setCursor(0, 1);
272.                 lcd.print("PENERIMA...");
273.                 String konfirmasi = "Pengantar Menagih Biaya Paket: " +
String(bayarValue) + ". /terima atau /tolak?";
274.                 bot.sendMessage(chat_id, konfirmasi);
275.                 currentState = WAIT_FOR_COMPLETION;
276.             }
277.         }
278.         break;
279.
280.     case WAIT_FOR_COMPLETION:
281.         // No keypad input expected in this state
282.         break;

```

```

283.     }
284. }
285.
286. void showMainMenu() {
287.     lcd.clear();
288.     lcd.setCursor(0, 0);
289.     lcd.print("MENU UTAMA");
290.     lcd.setCursor(0, 1);
291.     lcd.print("TEKAN KEYPAD");
292. }
293.
294. void handleNewMessages(int numNewMessages) {
295.     Serial.println("Handling new message");
296.     for (int i = 0; i < numNewMessages; i++) {
297.         String chat_id = String(bot.messages[i].chat_id);
298.         String text = bot.messages[i].text;
299.         String from_name = bot.messages[i].from_name;
300.
301.         if (text == "/start") {
302.             String welcome = "Gunakan perintah berikut untuk
mulai\n";
303.             welcome += "Gunakan /pintul [value] -- untuk mengatur
pintul\n";
304.             welcome += "Gunakan /pintu2 [value] -- untuk mengatur
pintu2\n";
305.             welcome += "Gunakan /status -- untuk melihat nilai
tersimpan\n";
306.             welcome += "Gunakan /reset -- untuk menghapus nilai
tersimpan\n";
307.             bot.sendMessage(chat_id, welcome, "");
308.         } else if (text.startsWith("/pintul ")) {
309.             pintul = text.substring(7).toInt();
310.             digitalWrite(relayPintulPin, LOW);
311.             delay(5000); // Membuka relay selama 2 detik
312.             digitalWrite(relayPintulPin, HIGH);
313.             bot.sendMessage(chat_id, "Nilai pintul diatur ke " +
String(pintul), "");
314.         } else if (text.startsWith("/pintu2")) {
315.             pintu2 = text.substring(7).toInt();
316.             digitalWrite(relayPintu2Pin, LOW);
317.             delay(5000); // Membuka relay selama 2 detik
318.             digitalWrite(relayPintu2Pin, HIGH);
319.             bot.sendMessage(chat_id, "Nilai pintu2 diatur ke " +
String(pintu2), "");
320.         } else if (text == "/status") {
321.             String status = "Nominal Pintu 1: " + String(pintul) +
"\n";
322.             status += "Nominal Pintu 2: " + String(pintu2) + "\n";
323.             status += "Jumlah Paket: " + String(paketCount);

```



```

324.     bot.sendMessage(chat_id, status, "");
325.     } else if (text == "/reset") {
326.         pintul = 0;
327.         pintu2 = 0;
328.         bot.sendMessage(chat_id, "Nominal Pintu diatur ulang",
            "");
329.     } else if (text == "/lcd") { //backup reset lcd
330.         lcd.begin(16, 2);         // Inisialisasi ulang LCD
            dengan ukuran 16x2
331.         lcd.backlight();         // Menyalakan kembali
            backlight LCD
332.         showMainMenu();           // Kembali ke menu utama
333.         currentState = MAIN_MENU;
334.     } else if (text == "/terima" || text == "/tolak") {
335.         handleKurirConfirmation(text);
336.     } else {
337.         String unknownCommand = "Perintah tidak dikenal. Gunakan
            /start untuk melihat daftar perintah.";
338.         bot.sendMessage(chat_id, unknownCommand, "");
339.     }
340. }
341. }
342.
343. void handleKurirConfirmation(String text) {
344.     if (currentState == WAIT_FOR_COMPLETION) {
345.         // Memastikan pintu tidak terbuka
346.         if (digitalRead(reedSwitchPin) != LOW) {
347.             lcd.clear();
348.             lcd.print("Pintu Terbuka!");
349.             delay(3000);
350.             showMainMenu();
351.             currentState = MAIN_MENU;
352.             return;
353.         }
354.
355.         // Jika paket diterima
356.         if (text == "/terima") {
357.             paketCount++; // Menambah hitungan paket
358.             String jumlah = "Paket diterima!";
359.             bot.sendMessage(chat_id, jumlah); // Mengirim pesan ke
            bot
360.
361.             // Membuka relay sesuai input bayar
362.             int bayarValue = bayarInput.toInt();
363.             if (bayarValue == pintul) {
364.                 pintuNomor = "1"; // Set nomor pintu terlebih dahulu
365.             } else if (bayarValue == pintu2) {
366.                 pintuNomor = "2"; // Set nomor pintu terlebih dahulu
367.             }

```

```

368.
369.     // Menampilkan pesan di LCD terlebih dahulu
370.     lcd.clear();
371.     lcd.print("PAKET DITERIMA");
372.     lcd.setCursor(0, 1);
373.     lcd.print("BUKA PINTU " + pintuNomor);
374.
375.     // Kemudian membuka relay setelah menampilkan di LCD
376.     if (bayarValue == pintu1) {
377.         digitalWrite(relayPintu1Pin, LOW);
378.         delay(5000); // Membuka relay selama 5 detik
379.         digitalWrite(relayPintu1Pin, HIGH);
380.         pintu1 = 0;
381.     } else if (bayarValue == pintu2) {
382.         digitalWrite(relayPintu2Pin, LOW);
383.         delay(2000); // Membuka relay selama 2 detik
384.         digitalWrite(relayPintu2Pin, HIGH);
385.         pintu2 = 0;
386.     }
387.
388.     // Jika paket ditolak
389. } else if (text == "/tolak") {
390.     lcd.clear();
391.     lcd.print("PAKET DITOLAK");
392.     lcd.setCursor(0, 1);
393.     lcd.print("MOHON ULANGI LAGI");
394.
395.     // Menghapus nilai nominal pintu yang diambil
396.     bayarInput = "";
397. }
398.
399.     delay(3000);
400.     showMainMenu(); // Kembali ke menu utama
401.     currentState = MAIN_MENU;
402. }
403. }
404.
405. // Fungsi untuk menginisialisasi kamera
406. void configInitCamera() {
407.     camera_config_t config;
408.     config.ledc_channel = LEDC_CHANNEL_0;
409.     config.ledc_timer = LEDC_TIMER_0;
410.     config.pin_d0 = Y2_GPIO_NUM;
411.     config.pin_d1 = Y3_GPIO_NUM;
412.     config.pin_d2 = Y4_GPIO_NUM;
413.     config.pin_d3 = Y5_GPIO_NUM;
414.     config.pin_d4 = Y6_GPIO_NUM;
415.     config.pin_d5 = Y7_GPIO_NUM;
416.     config.pin_d6 = Y8_GPIO_NUM;

```

```

417.     config.pin_d7 = Y9_GPIO_NUM;
418.     config.pin_xclk = XCLK_GPIO_NUM;
419.     config.pin_pclk = PCLK_GPIO_NUM;
420.     config.pin_vsync = VSYNC_GPIO_NUM;
421.     config.pin_href = HREF_GPIO_NUM;
422.     config.pin_sscb_sda = SIOD_GPIO_NUM;
423.     config.pin_sscb_scl = SIOC_GPIO_NUM;
424.     config.pin_pwdn = PWDN_GPIO_NUM;
425.     config.pin_reset = RESET_GPIO_NUM;
426.     config.xclk_freq_hz = 20000000;
427.     config.pixel_format = PIXFORMAT_JPEG;
428.     config.frame_size = FRAMESIZE_SVGA;
429.     config.jpeg_quality = 10;
430.     config.fb_count = 1;
431.
432.     // Initialize the camera
433.     esp_err_t err = esp_camera_init(&config);
434.     if (err != ESP_OK) {
435.         Serial.printf("Camera init failed with error 0x%x", err);
436.         return;
437.     }
438. }
439.
440. void sendPhotoTelegram() {
441.     const char* myDomain = "api.telegram.org";
442.     String getAll, getBody;
443.     int res = 0;
444.     camera_fb_t* fb = NULL;
445.
446.     // Mulai streaming
447.     fb = esp_camera_fb_get();
448.     if (!fb) {
449.         Serial.println("Camera capture failed");
450.         return;
451.     }
452.
453.     // Ambil frame terbaru dari stream
454.     esp_camera_fb_return(fb); // Kembalikan frame buffer yang
    lama
455.     fb = esp_camera_fb_get(); // Ambil frame terbaru
456.
457.     if (!fb) {
458.         Serial.println("Camera capture failed");
459.         return;
460.     }
461.
462.     // Proses pengiriman gambar selanjutnya
463.
464.

```

```

465.     Serial.println("Connect to " + String(myDomain));
466.
467.     if (client.connect(myDomain, 443)) {
468.         Serial.println("Connection successful");
469.         String head = "--boundary\r\nContent-Disposition: form-
data; name=\"chat_id\" \r\n\r\n" + chat_id + "\r\n--
boundary\r\nContent-Disposition: form-data; name=\"photo\";
filename=\"photo.jpg\"\r\nContent-Type: image/jpeg\r\n\r\n";
470.         String tail = "\r\n--boundary--\r\n";
471.
472.         uint16_t imageLen = fb->len;
473.         uint16_t extraLen = head.length() + tail.length();
474.         uint16_t totalLen = imageLen + extraLen;
475.
476.         client.println("POST /bot" + String(botToken) + "/sendPhoto
HTTP/1.1");
477.         client.println("Host: " + String(myDomain));
478.         client.println("Content-Length: " + String(totalLen));
479.         client.println("Content-Type: multipart/form-data;
boundary=boundary");
480.         client.println();
481.         client.print(head);
482.
483.         uint8_t* fbBuf = fb->buf;
484.         size_t fbLen = fb->len;
485.         for (size_t n = 0; n < fbLen; n = n + 1024) {
486.             if (n + 1024 < fbLen) {
487.                 client.write(fbBuf, 1024);
488.                 fbBuf += 1024;
489.             } else if (fbLen % 1024 > 0) {
490.                 size_t remainder = fbLen % 1024;
491.                 client.write(fbBuf, remainder);
492.             }
493.         }
494.         client.print(tail);
495.
496.         // Kembalikan buffer setelah selesai digunakan
497.         esp_camera_fb_return(fb);
498.
499.         int waitTime = 10000;
500.         long startWait = millis();
501.         bool state = false;
502.
503.         while ((startWait + waitTime) > millis()) {
504.             Serial.print(".");
505.             delay(100);
506.             while (client.available()) {
507.                 char c = client.read();
508.                 if (res == 1) getBody += String(c);

```

```

509.         if (c == '\n') {
510.             if (getAll.length() == 0) res = 1;
511.             getAll = "";
512.         } else if (c != '\r') getAll += String(c);
513.             startWait = millis();
514.         }
515.         if (getBody.length() > 0) break;
516.     }
517.     client.stop();
518.     Serial.println(getBody);
519. } else {
520.     getBody = "Connected to api.telegram.org failed.";
521.     Serial.println("Connection failed");
522. }
523. }
524.
525. // Function to check telegram messages
526. void checkTelegramMessages() {
527.     if (WiFi.status() == WL_CONNECTED) {
528.         if (!wifiConnectedMessageSent) {
529.             sendWifiConnectedMessage();
530.             wifiConnectedMessageSent = true;
531.         }
532.         Serial.println("Memeriksa Pesan Baru...");
533.         int numNewMessages =
            bot.getUpdates(bot.last_message_received + 1);
534.         Serial.print("Pesan Baru Diterima: ");
535.         Serial.println(numNewMessages);
536.         if (numNewMessages) {
537.             handleNewMessages(numNewMessages);
538.         }
539.     } else {
540.         wifiConnectedMessageSent = false;
541.     }
542. }
543.
544. void sendWifiConnectedMessage() {
545.     String message = "Penerima Paket terhubung dengan
internet\n";
546.     message += "Gunakan /start untuk melihat daftar perintah";
547.     bot.sendMessage(chat_id, message, "");
548. }
549.
550. float read_load_cell() {
551.     if (scale.is_ready()) {
552.         return scale.get_units(5);
553.     } else {
554.         Serial.println("Load cell tidak ditemukan.");
555.         return 0;

```


556. }

557. }

558.

559.

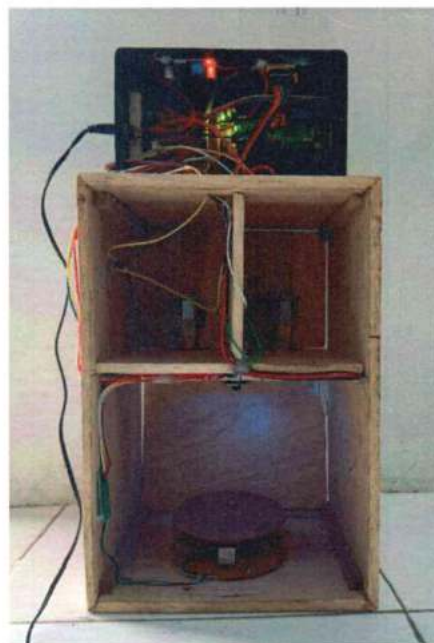
LISTING HASIL PROGRAM

1. Gambar Alat

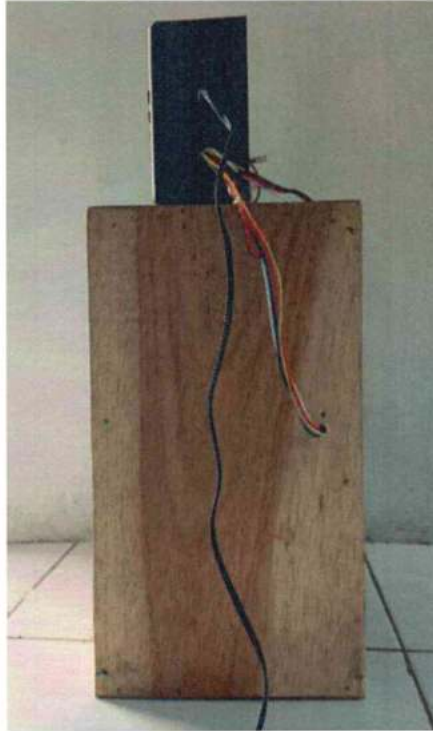
a. Tampak Depan



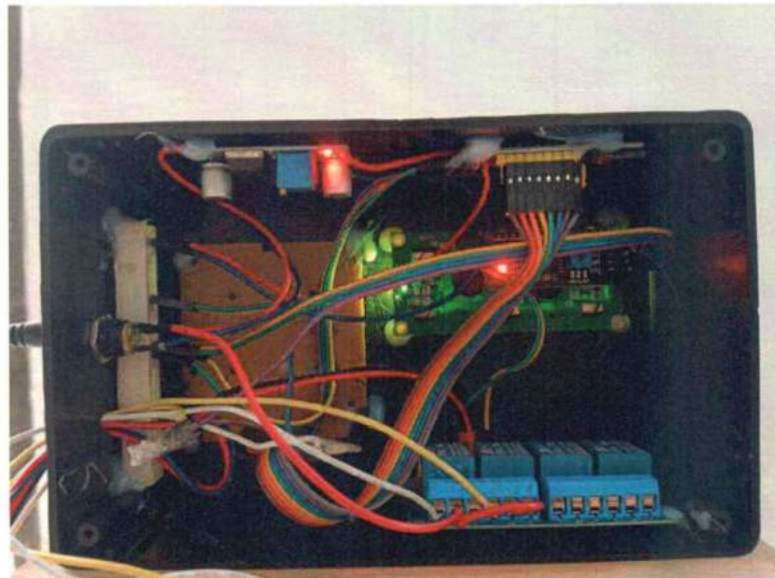
b. Tampak Belakang

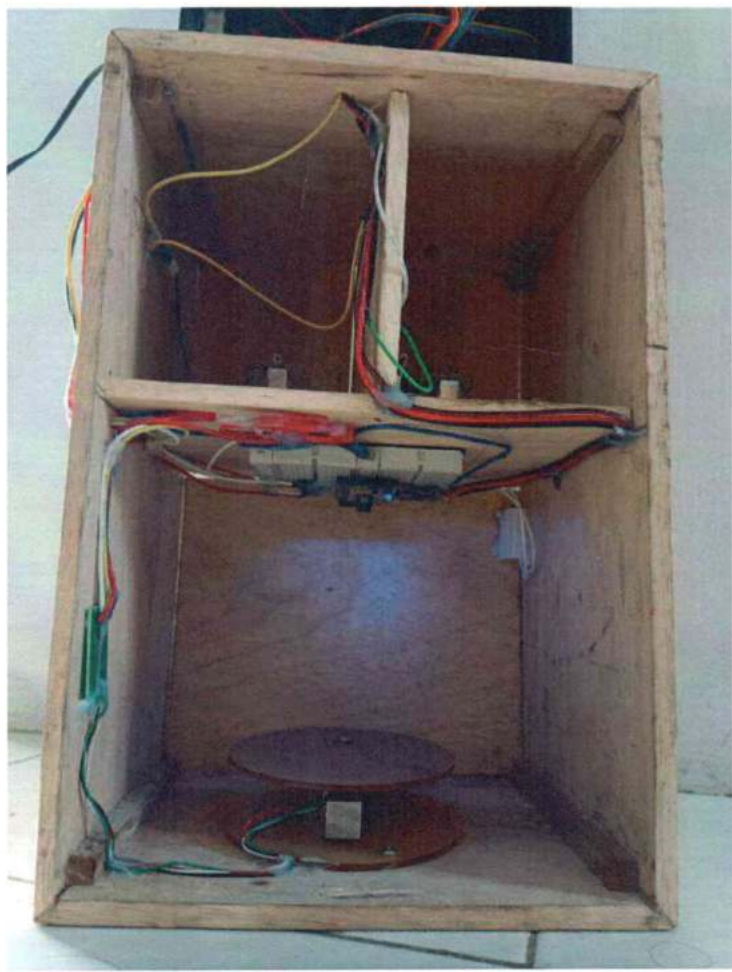
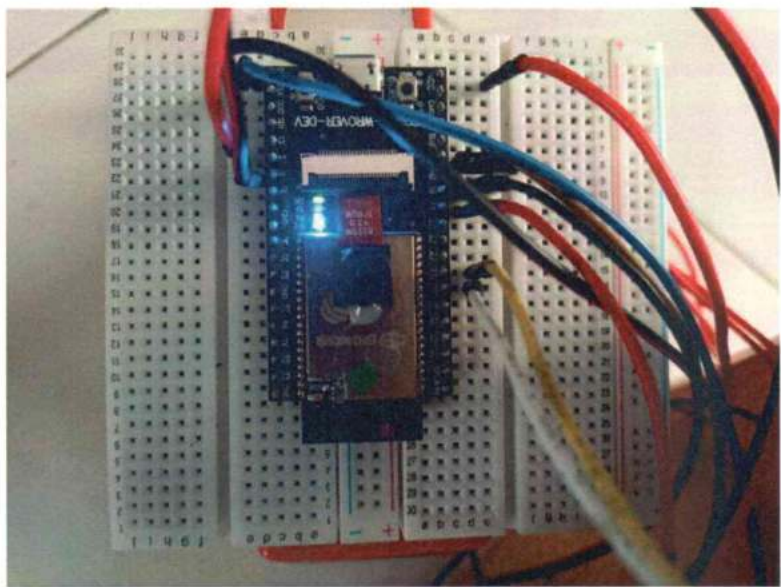


c. Tampak Samping

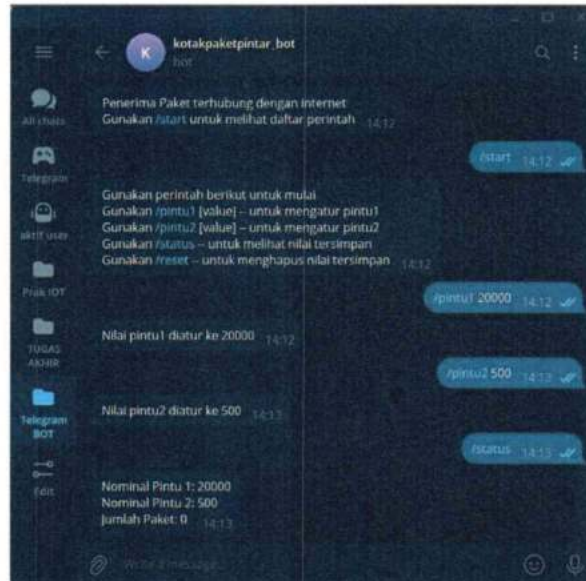


d. Mikrokontroller





2. Aplikasi Telegram



PETUNJUK PENGGUNAAN

1. PENYIAPAN ALAT

- Rakit perangkat mengikuti instalasi kabel seperti rangkaian sistem
- Hubungkan perangkat ke adaptor dan nyalakan perangkat
- Aktifkan Wi-Fi Hotspot sebagai penyedia internet untuk perangkat
- Ketika sudah terhubung perangkat akan mengirimkan pesan ke telegram
- Kirim pesan /pintu1 [value] dan atau /pintu2 [value] melalui aplikasi telegram. Sebagai contoh /pintu1 2000 dan /pintu2 500
- Siapkan uang dengan nominal sesuai untuk disimpan di pintu uang 1 dan atau pintu uang 2
- Setelah pesan diterima perangkat akan merespon dengan membuka pintu 1 dan 2 sesuai pesan yang akan terbuka dan tutup kembali pintu setelah uang disimpan dengan aman.

2. PETUNJUK PENGANTARAN

- Pengantar membuka pintu paket, meletakkan paket didalam kotak lalu menutup kembali pintu paket.
- Paket yang dimasukkan tidak melebihi ukuran dimensi kotak paket dan berat tidak melebihi 2 kilogram.
- Pengantar membaca pesan pada tampilan LCD adalah “MENU UTAMA TEKAN KEYPAD” kemudian menekan salah satu tombol pada keypad.
- Tampilan LCD akan menampilkan pesan “MASUKKAN BIAYA PAKET: ” Pengantar menekan tombol sesuai nominal paket, jika tidak ada biaya paket bisa menginputkan nilai 0.
- Gunakan tombol “C” untuk menghapus nilai yang diinput dan tombol “D” untuk enter atau menyimpan nilai yang diinput.
- Jika pengantar memasukkan nilai biaya paket sama dengan 0 maka perangkat akan menyimpan paket dan menyelesaikan proses pengantaran.
- jika pengantar memasukkan nilai biaya paket lebih besar dari 0, tampilan LCD akan menampilkan pesan “MENUNGGU KONFIRMASI PENERIMA”.
- Jika penerima membalas /terima maka transaksi selesai dan perangkat menyelesaikan proses pengantaran
- Jika penerima membalas /tolak maka pengantar diminta mengulang proses pengantaran dari awal.

3. PETUNJUK PENERIMAAN

- Ketika pengantar menekan tombol D setelah memasukkan nilai biaya paket, maka perangkat akan mengirimkan pesan informasi “Ada Paket dengan Biaya ... dan berat ... gram”.
- Jika biaya paket sama dengan 0 maka perangkat akan mengirimkan pesan paket diterima setelah transaksi diselesaikan oleh pengantar.
- Jika biaya paket lebih besar dari 0 maka perangkat akan mengirimkan foto setelah mengirimkan pesan informasi paket.
- Selanjutnya perangkat akan mengirimkan pesan konfirmasi “Kurir menagihkan biaya paket sebesar : ... balas dengan /terima atau /tolak”
- Pada tahap ini, perangkat akan terus menunggu pesan balasan masuk. Setelah membalas pesan konfirmasi maka transaksi selesai.

SURAT KETERANGAN
PERSETUJUAN PUBLIKASI

Bahwa yang bertanda tangan dibawah ini:

Nama : Dwi Apriyanto
Nomer Induk Mahasiswa : 213310043
Jurusan : Teknologi Komputer
Jenjang : D3
Judul : Implementasi Penerima Paket Berbasis Internet of Things (IoT) menggunakan NodeMCU ESP32 Wrover Cam

Menyerahkan karya ilmiah kepada pihak perpustakaan Universitas Teknologi Digital Indonesia untuk **diunggah ke Digital Library** Universitas Teknologi Digital Indonesia sesuai dengan ketentuan yang berlaku untuk kepentingan riset dan pendidikan.

Yogyakarta, 20 Agustus 2024

Penulis,



Dwi Apriyanto
Nim: 213310043

Catatan Pendadaran

Hari, tanggal : Senin, 12 Agustus 2024
 Waktu : 08.00
 Nama : Dwi Apriyanto
 No. Mahasiswa / Prodi : 213310043 / TK D3

	Hal yang harus diperbaiki	Pemberi Catatan
1.	<ul style="list-style-type: none"> - Tambahkan informasi pintu uang mana yang harus dibuka oleh kurir. - Naskah di abstrak isi 3 paragraf, 1. permasalahan, 2. solusi yang ditawarkan, 3. Hasil nya. - Kalau internet off/ngandat bagaimana sarannya. - Tujuan pengamanan belum terimplementasi, masukkan saja dalam saran. - Pembatas masalah: Internet dan Listrik On - Batas paket dari sisi berat dan dimensi. - Sitasi data atau kalimat disebutkan dalam naskah. - Referensi disesuaikan dengan ketentuan penulisan referensi - Keunggulan mengirimkan gambar dapat dituliskan di pendahuluan, Tujuan, masalah - Penulisan nomor halaman judul di tengah bawah, selain itu di kanan atas. - Komunikasi load cell dengan ESP menggunakan apa? 	Pak Adiyuda
2.	<ul style="list-style-type: none"> - Tinjauan Pustaka minimal sesuai dengan ketentuan di panduan. 	Pak Adiyuda Pak Yudhi
3.	<ul style="list-style-type: none"> - Koneksi internet jangan menjadi syarat awal operasionalnya alat - LCD yang kacau karakternya dapat dimasukkan saran untuk mengatasi. - Pembahasan implementasi kode program mengacu nobaris dari listing program keseluruhannya. - menyelaraskan kesimpulan dengan tujuan - kendala yang dialami selama pengujian dapat sebagai saran perbaikan - atur ukuran gambar/logo/ilustrasi dan tata letaknya sedemikian rupa 	Pak Yudhi

*coret yang tidak perlu

KEPUTUSAN HASIL UJIAN PENDADARAN

Sesuai dengan hasil sidang pendadaran pada tanggal 12 Agustus 2024 maka

Nama Mahasiswa
NIM / Program Studi
Jenjang

Dwi Apriyanto
213310043 / TK D3
D3

dinyatakan LULUS dengan kriteria 2

Ketua Penguji

Yudhi Kusnanto, S.T., M.T.



KARTU BIMBINGAN TUGAS AKHIR DIPLOMA TIGA
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA

Nama : Dwi Apriyanto
NIM : 213300043
Program Studi : teknologi komputer
Pembimbing : Totok Budioko, S.T., M.T.
Judul : Implementasi Penerima Paket Berbasis
Internet of Things (IoT) Menggunakan
Node MCU ESP32 Whover Cam.

NO	HARI / TANGGAL	TOPIK	PARAF PEMBIMBING
1	Kamis, 27 Juni 2024.	Pengajuan judul tugas akhir.	Tu
2	Selasa, 16 Juli 2024.	Bimbingan pengerjaan tugas akhir cada revisi	Tu
3	Kabu, 24 Juli 2024	Laporan progres pengerjaan tugas akhir (naskah revisi)	Tu
4	Selasa, 30 Juli 2024	Laporan proses pengerjaan naskah (ditanggapi lagi)	Tu
5	Selasa, 6 Agustus 2024	Naskah, Alat ACC didaftarkan pendadaran.	Tu
6			
7			
8			
9			
10			

Dosen Pembimbing TGA


Totok Budioko, S.T., M.T.

NIDN : 0522017102

Ace Ruzi
19/8-2024
Y

TUGAS AKHIR
IMPLEMENTASI PENERIMA PAKET
BERBASIS *INTERNET OF THINGS* (IOT) MENGGUNAKAN
NODEMCU ESP32 WROVER CAM



DWI APRIYANTO

NIM : 213310043

PROGRAM STUDI TEKNOLOGI KOMPUTER
PROGRAM DIPLOMA TIGA
FAKULTAS TEKNOLOGI INFORMASI
UNIVERSITAS TEKNOLOGI DIGITAL INDONESIA
YOGYAKARTA

2024