

Lampiran 3. Kode Program ESP32 untuk Koneksi ke Firebase

```
1. #include <SoftwareSerial.h>
2. #include <time.h>
3. #include <WiFi.h>
4. #include <Firebase_ESP_Client.h>
5.
6. //Provide the token generation process info.
7. #include "addons/TokenHelper.h"
8. //Provide the RTDB payload printing info and other helper
functions.
9. #include "addons/RTDBHelper.h"
10.
11. // Insert your network credentials
12. #define WIFI_SSID "jeskuyyyyyy"
13. #define WIFI_PASSWORD "3februari2"
14.
15. // Insert Firebase project API Key
16. #define API_KEY "AIzaSyDvyYy2VvSNSk7S8yCFJQKZ-2ZYGZGUqKw"
17.
18. // Insert RTDB URL
19. #define DATABASE_URL "https://parkingroom-6bdfd-default-
rtodb.asia-southeast1.firebaseio.com/"
20.
21. #define USER_EMAIL "jesicapurii32@gmail.com"
22. #define USER_PASSWORD "12345678"
23.
24. #define EN_Port 27
25.
26. #define TombolKeluar 4
27. #define Tombol 5
28.
29. SoftwareSerial espSerial(26, 25);
30.
31. String lp1a = "";
32. String lp1b = "";
33. String lp1c = "";
34.
35. String lp2a = "";
36. String lp2b = "";
37. String lp2c = "";
38.
39. String lp1 = "";
40. String lp2 = "";
41. String Status1 = "";
42. String Status2 = "";
43. String StatusButton = "standby";
44. String StatusButtonKeluar = "standby";
45. String inputString = "";           // a String to hold incoming
data
46. bool stringComplete = false;      // whether the string is
complete
47.
48. const char* ntpServer = "pool.ntp.org";
49. const long  gmtoffset_sec = 0;
50. const int   daylightOffset_sec = 25200;
51.
52. //Define Firebase Data object
53. FirebaseData fbdo;
```

```

54.
55. FirebaseAuth auth;
56. FirebaseConfig config;
57.
58. unsigned long sendDataPrevMillis = 0;
59. int count = 0;
60.
61. bool signupOK = false;
62. int timeout = 0;
63. char buffer[80];
64. int delayButton = 0;
65.
66. void setup() {
67.   delay(10000);
68.   espSerial.begin(9600);
69.   Serial.begin(115200);
70.   pinMode(EN_Port, OUTPUT);
71.   digitalWrite(EN_Port, HIGH);
72.   delay(500);
73.   pinMode(Tombol, INPUT);
74.   pinMode(TombolKeluar, INPUT);
75.   delay(500);
76.   WiFi.begin(WIFI_SSID, WIFI_PASSWORD);
77.   Serial.print("Connecting to Wi-Fi");
78.   while (WiFi.status() != WL_CONNECTED) {
79.     Serial.print(".");
80.     if (timeout < 40) {
81.       timeout++;
82.     } else {
83.       digitalWrite(EN_Port, LOW);
84.     }
85.     delay(500);
86.   }
87.   Serial.println();
88.   Serial.print("Connected with IP: ");
89.   Serial.println(WiFi.localIP());
90.   Serial.println();
91.
92.   configTime(gmtOffset_sec, daylightOffset_sec, ntpServer);
93.
94.   /* Assign the api key (required) */
95.   config.api_key = API_KEY;
96.   auth.user.email = USER_EMAIL;
97.   auth.user.password = USER_PASSWORD;
98.
99.   /* Assign the RTDB URL (required) */
100.  config.database_url = DATABASE_URL;
101.
102.  /* Sign up */
103.  if (Firebase.signUp(&config, &auth, "", "")) {
104.    Serial.println("ok");
105.    signupOK = true;
106.  } else {
107.    Serial.printf("%s\n",
config.signer.signupError.message.c_str());
108.  }
109.
110.  /* Assign the callback function for the long running token
generation task */

```

```

111. config.token_status_callback = tokenStatusCallback; // see
addons/TokenHelper.h
112.
113. Firebase.begin(&config, &auth);
114. Firebase.reconnectWiFi(true);
115.
116. delay(3000);
117. }
118.
119. void loop() {
120.   if (digitalRead(Tombol) == LOW) {
121.     StatusButton = "press";
122.   } else {
123.     if (delayButton >= 200) {
124.       StatusButton = "standby";
125.       delayButton = 0;
126.     } else {
127.       delayButton++;
128.     }
129.   }
130.
131.   if (digitalRead(TombolKeluar) == LOW) {
132.     StatusButtonKeluar = "press";
133.   } else {
134.     if (delayButton >= 200) {
135.       StatusButtonKeluar = "standby";
136.       delayButton = 0;
137.     } else {
138.       delayButton++;
139.     }
140.   }
141.   Serial.println(StatusButton);
142.   Serial.println(StatusButtonKeluar);
143.
144.   // Event Serial Komunikasi
145.   while (espSerial.available()) {
146.     char inChar = (char)espSerial.read();
147.     inputString += inChar;
148.     if (inChar == '\n') {
149.       stringComplete = true;
150.     }
151.   }
152.   delay(50);
153.
154.   if (stringComplete) {
155.     lp1a = inputString.substring(0, 1);
156.     lp1b = inputString.substring(1, 2);
157.     lp1c = inputString.substring(2, 3);
158.
159.     lp2a = inputString.substring(3, 4);
160.     lp2b = inputString.substring(4, 5);
161.     lp2c = inputString.substring(5, 6);
162.
163.     if ((lp1a.toInt() == 1) && (lp1b.toInt() == 1) &&
(lp1c.toInt() == 0)) {
164.       Status1 = "Full";
165.     } else {
166.       Status1 = "Available";
167.     }

```

```

168.
169.     if ((lp2a.toInt() == 1) && (lp2b.toInt() == 1) &&
170.         (lp2c.toInt() == 0)) {
171.         Status2 = "Full";
172.     } else {
173.         Status2 = "Available";
174.     }
175.     inputString = "";
176.     stringComplete = false;
177. }
178. delay(50);
179.
180.     if ((millis() - sendDataPrevMillis > 10000 ||
181.         sendDataPrevMillis == 0)) {
182.         sendDataPrevMillis = millis();
183.         time_t rawtime;
184.         struct tm *timeinfo;
185.         time(&rawtime);
186.         timeinfo = localtime(&rawtime);
187.         strftime(buffer, 80, " %d %B %Y %H:%M:%S ", timeinfo);
188.         if (Firebase.RTDB.setString(&fbdo, "parkir/Button",
189.             StatusButton)) {
190.             Serial.println("PASSED");
191.             Serial.println("PATH: " + fbdo.dataPath());
192.             Serial.println("TYPE: " + fbdo.dataType());
193.         } else {
194.             Serial.println("FAILED");
195.             Serial.println("REASON: " + fbdo.errorReason());
196.         }
197.         if (Firebase.RTDB.setString(&fbdo, "parkir/ButtonKeluar",
198.             StatusButtonKeluar)) {
199.             Serial.println("PASSED");
200.             Serial.println("PATH: " + fbdo.dataPath());
201.             Serial.println("TYPE: " + fbdo.dataType());
202.         } else {
203.             Serial.println("FAILED");
204.             Serial.println("REASON: " + fbdo.errorReason());
205.         }
206.         if (Firebase.RTDB.setInt(&fbdo, "parkir/ruang/Lantai_1",
207.             lp1a.toInt() + lp1b.toInt() + lp1c.toInt())) {
208.             Serial.println("PASSED");
209.             Serial.println("PATH: " + fbdo.dataPath());
210.             Serial.println("TYPE: " + fbdo.dataType());
211.         } else {
212.             Serial.println("FAILED");
213.             Serial.println("REASON: " + fbdo.errorReason());
214.         }
215.         if (Firebase.RTDB.setInt(&fbdo, "parkir/ruang/Lantai_2",
216.             lp2a.toInt() + lp2b.toInt() + lp2c.toInt())) {
217.             Serial.println("PASSED");
218.             Serial.println("PATH: " + fbdo.dataPath());
219.             Serial.println("TYPE: " + fbdo.dataType());
220.         } else {
221.             Serial.println("FAILED");

```

```

221.     Serial.println("REASON: " + fbdo.errorReason());
222. }
223.
224.         if      (Firebase.RTDB.setString(&fbdo,
"parkir/historyIR/Lantai_1/", lp1a + lp1b + lp1c)) {
225.     Serial.println("PASSED");
226.     Serial.println("PATH: " + fbdo.dataPath());
227.     Serial.println("TYPE: " + fbdo.dataType());
228. } else {
229.     Serial.println("FAILED");
230.     Serial.println("REASON: " + fbdo.errorReason());
231. }
232.
233.         if      (Firebase.RTDB.setString(&fbdo,
"parkir/historyIR/Lantai_2/", lp2a + lp2b + lp2c)) {
234.     Serial.println("PASSED");
235.     Serial.println("PATH: " + fbdo.dataPath());
236.     Serial.println("TYPE: " + fbdo.dataType());
237. } else {
238.     Serial.println("FAILED");
239.     Serial.println("REASON: " + fbdo.errorReason());
240. }
241.
242. count++;
243.
244.         if      (Firebase.RTDB.setString(&fbdo,
"parkir/status/Lantai_1", Status1)) {
245.     Serial.println("PASSED");
246.     Serial.println("PATH: " + fbdo.dataPath());
247.     Serial.println("TYPE: " + fbdo.dataType());
248. } else {
249.     Serial.println("FAILED");
250.     Serial.println("REASON: " + fbdo.errorReason());
251. }
252.
253.         if      (Firebase.RTDB.setString(&fbdo,
"parkir/status/Lantai_2", Status2)) {
254.     Serial.println("PASSED");
255.     Serial.println("PATH: " + fbdo.dataPath());
256.     Serial.println("TYPE: " + fbdo.dataType());
257. } else {
258.     Serial.println("FAILED");
259.     Serial.println("REASON: " + fbdo.errorReason());
260. }
261.
262.     if (Firebase.RTDB.setString(&fbdo, "parkir/tanggal",
buffer)) {
263.     Serial.println("PASSED");
264.     Serial.println("PATH: " + fbdo.dataPath());
265.     Serial.println("TYPE: " + fbdo.dataType());
266. } else {
267.     Serial.println("FAILED");
268.     Serial.println("REASON: " + fbdo.errorReason());
269. }
270. }
271. }

```

Lampiran 4. Kode Program Mengambil Data dari Firebase ke PHP

```

1. <?php
2. require __DIR__ . '/../vendor/autoload.php';
3. use Kreait\Firebase\Factory;
4.
5. $factory = (new Factory())
6.     ->withServiceAccount('parkingroom-4073a-firebase-
adminsdk-g0odp-9138dcb6c6.json')
7.     ->withDatabaseUri('https://parkingroom-4073a-default-
rtadb.asia-southeast1.firebaseio.com');
8.
9. /*
10.     withServiceAccount dari config API Firebase
11.     withDatabaseUri dari URL database Firebase
12. */
13.
14. $database = $factory->createDatabase();

```



Lampiran 5. Kuesioner Google Form



Kuesioner Sistem Monitoring Ketersediaan Parkir Berbasis Internet of Things

Halo perkenalkan saya Theresia Jesica mahasiswa program studi Teknologi Informasi Universitas Darma Persada. Saat ini saya sedang melakukan penelitian terkait kesulitan pengunjung ITC Depot dalam mencari tempat parkir mobil terutama pada akhir pekan dan hari libur nasional. Apabila Anda berkenan, silahkan isi form ini sesuai dengan pengalaman Anda. Terima kasih

jesicapurii32@gmail.com [Ganti akun](#)

 Tidak dibagikan

* Menunjukkan pertanyaan yang wajib diisi

Nama Anda *

Jawaban Anda

Umur Anda *

☐ <20 tahun

☐ 20-30 tahun

☐ 31-40 tahun

☐ 41-50 tahun

☐ >50

Jenis Kelamin *

☐ Laki-laki

☐ Perempuan

Pekerjaan *

Jawaban Anda

Berikutnya

Kosongkan formulir

PENILAIAN SISTEM MONITORING KETERSEDIAAN PARKIR

Keterangan Penilaian:

1 = sangat tidak setuju

2 = tidak setuju

3 = netral

4 = setuju

5 = sangat setuju

Apakah saat ini sudah ada sistem monitoring ketersediaan parkir? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Kesan pertama Anda terhadap sistem monitoring ketersediaan parkir? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Kemudahan akses informasi ketersediaan parkir melalui layar LCD? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Kemudahan akses informasi ketersediaan parkir melalui layar LED merah dan LED hijau? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Kemudahan penggunaan scan barcode untuk informasi ketersediaan parkir? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Alat dapat bekerja secara real time? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Efektivitas alat dalam membantu pengunjung menemukan tempat parkir kosong *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Kemudahan pengelolaan sistem parkir secara keseluruhan dengan alat ini? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

penilaian keseluruhan keefektifan sistem monitoring ketersediaan parkir? *

1 2 3 4 5
sangat tidak setuju ☐ ☐ ☐ ☐ ☐ sangat setuju

Apakah Anda memiliki saran atau ide untuk pengembangan sistem monitoring ketersediaan parkir yang lebih baik? *

Jawaban Anda

Kembali

Kirim

Kosongkan formulir

Lampiran 6. Hasil Data Kuesioner

Responden	Penilaian Sistem Monitoring Ketersediaan Parkir									
	X1	X2	X3	X4	X5	X6	X7	X8	X9	Total X
1	1	5	5	5	4	5	4	5	5	39
2	1	5	5	4	4	4	4	5	4	36
3	1	4	4	4	3	4	5	5	4	34
4	1	5	5	5	4	5	5	5	5	40
5	1	5	5	5	5	4	5	4	5	39
6	1	5	5	5	5	5	5	4	5	40
7	5	1	3	4	4	4	4	4	5	34
8	1	3	4	5	5	5	4	4	4	35
9	1	4	5	5	5	4	5	5	4	38
10	1	4	5	5	5	5	5	5	5	40
11	1	4	4	4	4	4	5	4	4	34

Lampiran 7. Hasil Data Kuesioner

Statistic	Score
N	11
Minimum	34
Maximum	40
Mean	37.5
Standard Deviation	2.725540575