

DAFTAR PUSTAKA

- Aditya, Firdaus dan Handoyo, Eko. (2011). Perbandingan Paid Hosting dan Free Hosting Berdasarkan Fasilitas Backup Yang Ada. *Laboratorium Komputer, Teknik Elektro, Fakultas Teknik, Universitas Diponegoro*. Vol.4, no. 1, 1-5
- Blanc, Y., & Dimanico, U. (2014). Electrode Placement in Surface Electromyography (sEMG) “Minimal Crosstalk Area” (MCA). *The Open Rehabilitation Journal*, 3(1), 110126. <https://doi.org/10.2174/1874943701003010110>
- By AllDataSheetCom, P. (n.d.). ILI9488 a-Si TFT LCD Single Chip Driver 320(RGB) x 480 Resolution, 16.7M-color With Internal Gram Specification. <http://www.ilitek.com>
- Cahyono, A. (2016). Rancang Bangun Modul EKG (Elektrokardiograf) Menggunakan Raspberry. *Teknologi Nasional Malang*.
- Destiarini, & Kumara Pius Widya. (2019). Robot Line Follower Berbasis Mikrokontroler Arduino Uno Atmega328. *Jurnal informanika*. Vol 5. No.1.
- E. Systems, “ESP32 Series Datasheet, (2019). ” Espressif Systems, Espressif Systems, hal. 1–61.
- Efendi, Y. (2018). *Internet OF Things (IOT) Sstem Pengendalian Lampu Menggunakan Raspberry PI Berbasis Mobile*. *Jurnal Ilmiah Ilmu Komputer*, 4(1). <https://doi.org/https://doi.org/10.35329/jiik.v4i2.41> .
- Fianti, S. (2017). Transistor Film Tipis Organik [Universitas Negeri Semarang]. <https://textid.123dok.com/document/myj5nl56qprinsipkerja-monitorlcd-tft-thin-film-transistor-transistor-film-tipis-organik.html>
- Ghazali, Telaumbanua Achmad. (2018) “Pengembangan Aplikasi Pemesanan Layanan Kecantikan Berbasis Progressive Web Apps (PWA)”. Skripsi.
- Goetzberger, A. (Adolf), & Hoffmann, V. U. (2005). Photovoltaic solar energy generation. Springer. <http://fulviofrisone.com/attachments/article/455/SpringerVerlag%20Photovoltaic%20Solar%20Energy%20Generation.pdf>

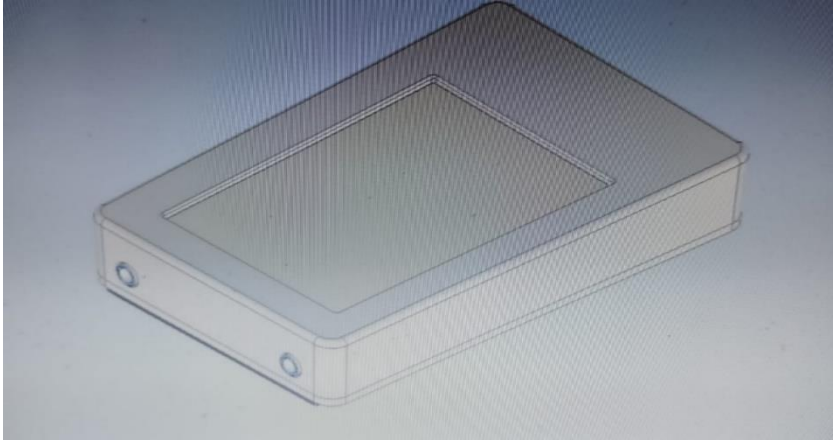
- Imam. (2022). Rancang Bangun Wireles Electrocardiogram (ECG) Berbasis IoT. Makassar : Fakultas Teknik, Universitas Hasanuddin.
- Imam,Muammarul. Dkk. (2019). Pengendalian Suhu Air Menggunakan Sensor Suhu DS18B20. Fakultas Teknik, Universitas Negeri Semarang. Vol 06. No. 01
- Maulana, Ilham Firman. (2020). Penerapan Firebase Realtime Database pada Aplikasi E-Tilang Smartphone berbasis Mobile Android. Jurnal Resti (Rekayasa Sistem dan Teknologi Informasi). Universitas Kristen Satya Wacana. Vol. 4 No. 5. 854 - 863
- Momin, Abdul, dkk. (2021). Rancang Bangun Elektokardiograf Berbasis IoT. Jurnal Fisika 11. Fakultas Matematika dan Ilmu Pengetahuan Alam, Universitas Jenderal Soedirman. 60-76
- Nuryati, Venti. (2010). Rancang Bangun Alat Pendeteksi dan Penghitung Detak Jantung dengan Asas Doppler. Universitas Indonesia, Depok.
- Permana, A. Yudi.(2019).“Perancangan Sistem Informasi Penjualan Perumahan Menggunakan Metode SDLC Pada PT.Mandiri Land Prosperous Berbasis Mobile”.Jurnal Teknologi Pelita Bangsa,vol 10, no. 2
- Pradisti, R., dan Mustaziri, (2018). Rancang Bangun Alat Penghitung Biaya Penggunaan Listrik Kamar Kos Secara Otomatis Berbasis Arduino Menggunakan Sensor Arus, TEKNIKA (Jurnal Ilmiah Bidang Ilmu Rekayasa), 12(2).
- Putri, Alif Fitriani & Widianoro Anang. (2020), Monitoring EKG (Elektrokardiograf) Berbasis Mikrokontroller Dan Pemrograman Delphi 7.0. Jurnal Teknik Elektro dan Komputer Triac. Universitas Muhammadiyah Surabaya, Indonesia. Vol. 7 No. 1
- Putri, Rara Amita. dkk. (2017). Rancang Bangun Wireless Elektrokardiogram (EKG). Jurnal Ilmu dan Inovasi Fisika. Departemen Fisika FMIPA Universitas Padjadjaran. Vol. 01, No. 01, 58 – 64
- Rifali, Mysa, Irmawati Dessy. (2019). Sistem Cerdas Elektrokardiogram (EKG) Untuk Klasifikasi Jantung Normal dan Abnormal Menggunakan Syaraf Tiruan (JST). Teknik Elektronika, Fakultas Teknik Universitas Negeri Yogyakarta. 4(1):49-55

- Silvia, A.F., Haritman, E., dan Muladi, Y., 2014. Rancang Bangun Akses Kontrol Pintu Gerbang Berbasis Arduino Dan Android, *Electrans*, 13(1).
- Sokop, S.J., Mamahit, D.J., dan Sompie, S.R.U.A., 2016. Trainer Periferal Antarmuka Berbasis Mikrokontroler Arduino Uno, *Jurnal Teknik Elektro Dan Komputer*, 5(3).
- Syahril, Muhammad. (2013), "Panduan Mudah Simulasi dan Praktik: Mikrokontroler Arduino." Yogyakarta: Andi Publisher
- Uswarman, Rudi. (2017). Desain dan Implementasi Elektrokardiograf (EKG) Portable Menggunakan Arduino. *Electrician-Jurnal Rekayasa dan Teknologi Elektro*. Institut Teknologi Sumatera. Lampung Selatan.
- Wagya, A., dan Rahmat, 2019. Prototipe Modul Praktik untuk Pengembangan Aplikasi Internet of Things (IoT), *SETRUM: Sistem Kendali Tenaga Elektronika Telekomunikasi Komputer*, 8(2).
- Wahyudi, J., 2014. Audit Energi Di Bidang Tata Cahaya Untuk Gedung Kampus Bonaventura UAJY, Skripsi, Universitas Atma Jaya Yogyakarta.
- Wijaksana, Isma, T., Yuliza, M., Angraini, T., & Susanti, R. (2020). Efektifitas Sensor Elektrokardiograf (EKG) AD8232 Untuk Mendeteksi Kelelahan Pada Saat Penggunaan SMARTPHONE. *Elektron Jurnal Ilmiah*, 12. <https://jie.pnp.ac.id/index.php/jie/article/download/148/107/>
- Zaini, M., dkk. 2020. Perancangan Sistem Monitoring Tegangan, Arus Dan Frekuensi Pada Pembangkit Listrik Tenaga Mikrohidro Berbasis IoT, *TESLA: Jurnal Teknik Elektro*, 22(2).
- Zhang, J. X. J., & Hoshino, K. (2008). Surface Electrode Implantable and wearablesensors.<https://www.sciencedirect.com/topics/engineering/surface-electrode/pdf>
- Bacheramsyah, S. G., Suraatmadja, M. S., & Dwi Kuncoro, C. B. (2015). Perancang Pengolah Sinyal EKG dengan Menggunakan Filter Wiener pada Software LabVIEW Secara *Real Time*
- Desmon, P., Adiwijaya, & Utama, D. Q. (2018). Deteksi Penyakit Gagal Jantung Berdasarkan Sinyal EKG Menggunakan Naive Bayes. *eProceedings of Engineering*, 5(2). Telkom University.

- Winursito, A. (2022). Pengembangan Sistem Monitoring Kesehatan Jantung Tahan Noise Berbasis Sinyal EKG. *Jurnal Sarjana Teknik Informatika*, 10(2), 56-65. Universitas Ahmad Dahlan
- Hapsoro, Emmanuel Juan Patrick, Hapsari, Gita Indah, & Sani, Muhammad Ikhsan(2023). Rancang Bangun Piranti Keras Sistem Deteksi Detak Jantung Berbasis IoT. Universitas Telkom, D3 Teknologi Komputer
- Wiajayanto, D. S. (2017). Implementasi Jaringan Sensor Nirkabel untuk Pemantauan Electrocardiogram di Tenda Medis Menggunakan Raspberry Pi. Undergraduate thesis, Institut Teknologi Sepuluh Nopembe
- HiveMQ. (n.d.). HiveMQ MQTT Broker - Reliable, Scalable, and Secure. Retrieved October 5, 2024, from <https://www.hivemq.com/>

LAMPIRAN

Lampiran 1 Tampilan gambar design 3d case ECG



Lampiran 2 Dokumentasi Pengujian Alat





Lampiran 2 program firmware ecg

```

1  // semua librari yang diperlukan
2  #include <SPI.h>
3  #include <Wire.h>
4  #include <TFT_eSPI.h>
5  #include <Adafruit_ADS1X15.h>
6  #include <DallasTemperature.h>
7  #include <OneWire.h>
8  #include <PubSubClient.h>
9  #include <WiFi.h>
10
11 #define SSID "Eri"
12 #define PASSWORD "09090909"
13 #define MQTTHOST "broker.mqtt-dashboard.com"
14 #define MQTTPORT 1883
15 #define ONE_WIRE_BUS 5
16
17 // deklarasi semua variabel yang dibutuhkan
18 OneWire oneWire(ONE_WIRE_BUS);
19 DallasTemperature sensors(&oneWire);
20
21 TFT_eSPI tft = TFT_eSPI();
22 SPIClass hspi(HSPI);
23 // set nilai awal menjadi 0
24 uint32_t lastBeatTime = 0;
25 uint32_t lastTempTime = 0;
26 bool beatDetected = false;
27 int BPM = 0;
28 float tempC = 0;
29
30 int32_t c1;
31 int graphData[81]; // jumlah titik data untuk grafik
32 float scaling = 1;
33
34 Adafruit_ADS1115 ads;
35 WiFiClient espClient;
36 PubSubClient client(espClient);

```

```

38 // Parameters for moving average filter
39 const int numReadings = 10;
40 int readings[numReadings];
41 int readIndex = 0;
42 int total = 0;
43 int average = 0;
44
45 // fungsi agar bpm lebih akurat
46 const int bpmWindowSize = 3;
47 int bpmReadings[bpmWindowSize];
48 int bpmIndex = 0;
49 int bpmTotal = 0;
50 int smoothedBPM = 0;
51
52 //connecting wifi
53 void setupWifi() {
54     delay(10);
55     Serial.println();
56     Serial.print("connect to ");
57     Serial.println(SSID);
58     WiFi.mode(WIFI_STA);
59     WiFi.begin(SSID, PASSWORD);
60
61 // jika wifi tidak connect maka akan mencoba untuk menyambungkan kembali
62 while (WiFi.status() != WL_CONNECTED) {
63     delay(500);
64     Serial.print(".");
65 }
66 // jika wifi connect
67 Serial.println("");
68 Serial.println("WiFi Connected");
69 Serial.println("IP Address : ");
70 Serial.println(WiFi.localIP());
71 }
72
73 // mengembalikan data dari mqtt ke alat
74 void callback(char* topic, byte* payload, unsigned int length) {
75     payload[length] = '\0';
76     Serial.print("Pesan diterima [");
77     Serial.print(topic);
78     Serial.print("] ");
79     for (int i = 0; i < length; i++) {
80         Serial.print((char)payload[i]);
81     }
82     Serial.println();
83 }
84 // fungsi untuk mengetes apakah alat terhubung ke mqtt atau tidak
85 void reconnect() {
86     while (!client.connected()) {
87         Serial.print("Mencoba koneksi MQTT...");
88         String clientId = "man123";
89         clientId += String(random(0xffff), HEX);
90
91         if (client.connect(clientId.c_str())) { // jika berhasil connect to mqtt
92             Serial.println("Terhubung");
93             client.publish("outTopic", "hello world");
94             client.subscribe("inTopic");
95         } else {
96             Serial.print("Gagal, rc=");
97             Serial.print(client.state());
98             Serial.println(" coba lagi dalam 5 detik");
99             delay(5000); // jika gagal connect to mqtt
100         }
101     }
102 }

```

```

103 void setup() { // fungsi untuk mengatur semua bagian dari program alat
104     Serial.begin(115200);
105     tft.setRotation(1);
106     tft.begin();
107     tft.fillScreen(TFT_BLACK);
108     Home();
109     tft.fillScreen(TFT_BLACK);
110     tft.setTextColor(TFT_GREEN, TFT_BLACK);
111     tft.fillRect(120, 0, 10, 320, TFT_BLUE);
112     tft.fillRect(0, 275, 120, 5, TFT_BLUE);
113
114     if (!ads.begin(0x49)) { // jika ads tidak terhubung
115         Serial.println("Failed to initialize ADS.");
116         while (1);
117     }
118     ads.setGain(GAIN_ONE);
119
120     for (int i = 0; i < 81; i++) { // Menyesuaikan ukuran array data, dan fungsi for untuk mengisi nilai 3 array menjadi 0
121         graphData[i] = 0;
122     }
123
124     for (int i = 0; i < numReadings; i++) {
125         readings[i] = 0;
126     }
127
128     for (int i = 0; i < bpmWindowSize; i++) {
129         bpmReadings[i] = 0;
130     }
131
132     setupWifi();
133     sensors.begin();
134
135     client.setServer(MQTT_HOST, MQTT_PORT);
136     client.setCallback(callback);
137 }

175     average = total / numReadings;
176     c1 = average;
177     return int(c1);
178 }
179 // fungsi untuk menggambar grafik
180 void drawGraph() {
181     int value = ads.readADC_SingleEnded(0); // penggambaran grafik dari data adc yang di peroleh dari pin AIN0
182
183     total = total - readings[readIndex]; // fungsi penggambaran grafik
184     readings[readIndex] = value;
185     total = total + readings[readIndex];
186     readIndex = (readIndex + 1) % numReadings;
187
188     average = total / numReadings;
189     c1 = average;
190     graphData[80] = int(c1); // jumlah data untuk penggambaran grafik
191
192     // Menghapus garis lama dari grafik dengan lebih sedikit titik
193     for (int i = 1; i <= 79; i++) {
194         tft.drawLine(130 + (i - 1) * 3, 200 + (10000 - graphData[i - 1]) / scaling,
195             130 + i * 3, 200 + (10000 - graphData[i]) / scaling, TFT_BLACK); // fungsi untuk scaling
196     }
197
198     // Menghitung nilai maksimum dan minimum untuk scaling
199     int maxValue = 0;
200     int minValue = 10000000;
201     for (int i = 0; i <= 79; i++) {
202         if (graphData[i] > maxValue)
203             maxValue = graphData[i];
204         if (graphData[i] < minValue)
205             minValue = graphData[i]; // rumus scaling
206     }
207
208     // Menggeser data kekiri untuk grafik
209     for (int i = 0; i <= 79; i++) {
210         graphData[i] = graphData[i + 1];

```

```

213 // Menyesuaikan scaling jika perlu jika melebihi pixel yang digunakan
214 int maxPixelsAllowed = 80;
215 if ((maxValue - minValue) >= maxPixelsAllowed) {
216     scaling = (maxValue - minValue) / float(maxPixelsAllowed);
217 }
218
219 // Menggambar ulang grafik dengan data baru
220 for (int i = 1; i <= 79; i++) {
221     tft.drawLine(130 + (i - 1) * 3, 200 + (10000 - graphData[i - 1]) / scaling,
222                 130 + i * 3, 200 + (10000 - graphData[i]) / scaling, TFT_GREEN);
223 }
224 }
225 // menghitung nilai bpm
226 void calculateBPM() {
227     int ecgValue = readEcg(); // Baca nilai ECG dari adc yang terhubung dengan ADS1115
228     const int R_threshold = 2000; // Threshold untuk mendeteksi puncak R
229     static unsigned long lastPeakTime = 0;
230
231     if (ecgValue == 0) { // Jika nilai ADC 0, set BPM ke 0
232         smoothedBPM = 0;
233     } else if (ecgValue > R_threshold) { // Deteksi puncak R
234         unsigned long currentTime = millis();
235         if (currentTime - lastPeakTime > 700) { // Hindari deteksi ganda dalam 300ms
236             if (lastPeakTime > 0) {
237                 unsigned long interval = currentTime - lastPeakTime; // Hitung interval R-R
238                 int newBPM = 60000.0 / interval; // Hitung BPM dari interval R-R
239
240                 // Memasukkan nilai BPM ke dalam array untuk smooth filtering
241                 bpmTotal = bpmTotal - bpmReadings[bpmIndex];
242                 bpmReadings[bpmIndex] = newBPM;
243                 bpmTotal = bpmTotal + bpmReadings[bpmIndex];
244                 bpmIndex = (bpmIndex + 1) % bpmWindowSize;
245
246                 smoothedBPM = bpmTotal / bpmWindowSize; // Hasil BPM yang telah dihaluskan
247                 Serial.print("BPM: ");
248                 Serial.println(smoothedBPM);
249             }
250             lastPeakTime = currentTime; // Set waktu puncak terakhir
251         }
252     }
253 }
254
255 // mengambil data suhu dari sensor suhu
256 float readTemp() {
257     sensors.requestTemperatures();
258     return sensors.getTempCByIndex(0);
259 }
260

```

```

261 void display() { // pengaturan tampilan pada layar tft
262     tft.setTextColor(TFT_GREEN, TFT_BLACK);
263     int padding = tft.textWidth("999", 6);
264     tft.setTextPadding(padding);
265     tft.setTextDatum(TC_DATUM);
266     tft.drawNumber(smoothedBPM, 60, 150, 6);
267     tft.setTextFont(2);
268     tft.drawString("BPM", 60, 200, 2);
269     tft.setTextColor(TFT_WHITE, TFT_BLACK);
270     tft.setTextFont(2);
271     tft.drawString("ECG 02", 55, 290, 4);
272     tft.drawString("Suhu Tubuh :", 55, 40, 2);
273     tft.setTextColor(TFT_YELLOW, TFT_BLACK);
274     tft.setTextDatum(TC_DATUM);
275     String tempString = String(tempC, 1) + " °C";
276     tft.drawString(tempString.c_str(), 60, 70, 4);
277 }
278
279 void Home() { // tampilan awal pada saat layar dinyalakan
280     tft.setTextColor(TFT_RED, TFT_BLACK);
281     tft.setTextPadding(0);
282     tft.setTextDatum(TC_DATUM);
283     tft.drawString("TEKNIK 09", 240, 90, 4);
284     tft.drawString("2018", 240, 130, 4);
285     for (int16_t t = 0; t < 480; t += 1) {
286         tft.fillRect(t, 200, 1, 25, TFT_WHITE);
287         delay(10);
288     }
289 }
---
```