



## **SISTEM MIKROPROSESSOR**

### **LAPORAN UAS SISTEM MIKROPROSESSOR**

**DOSEN PENGAMPU :**

**BPK. MUHAMMAD IKHWAN FATHULLOH, S.KOM.**

**NASWA MUTIARA**

**23552011185 | TIF K 23A**

**PROGRAM SARJANA DEPARTEMEN TEKNIK INFORMATIKA  
FAKULTAS INDUSTRI KREATIF UNIVERSITAS TEKNOLOGI BANDUNG**

**2026**

# Laporan Sistem Smart Gorden Berbasis ESP32

## 1. Pendahuluan

Sistem ini dirancang untuk mengontrol gorden di rumah secara otomatis maupun manual melalui web dashboard dan protokol MQTT. ESP32 digunakan sebagai controller utama, terhubung ke WiFi, dapat beroperasi dalam mode STA (terhubung ke router) atau AP mode (akses langsung melalui ESP32). Sistem juga mendukung notifikasi status gorden melalui MQTT.

Fitur utama:

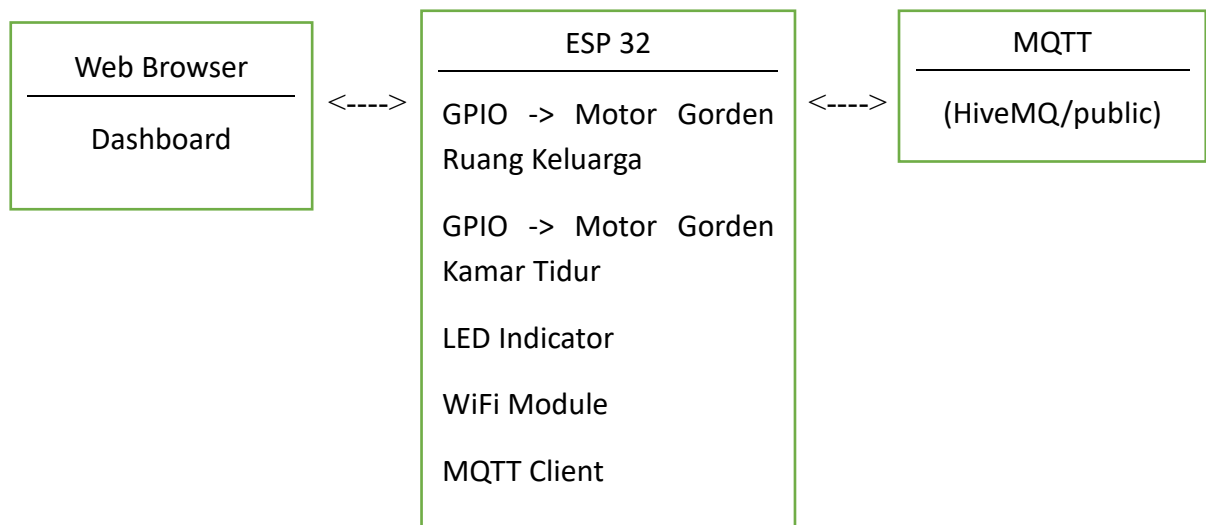
- Kontrol gorden Ruang Keluarga dan Kamar Tidur.
- Kontrol manual melalui web browser.
- Kontrol jarak jauh via MQTT (memakai HiveMQ)
- Status LED indikator (ON jika gorden terbuka).

## 2. Alur Sistem

### 2.1 Alur Kerja

- 1) ESP32 booting dan membaca konfigurasi WiFi dari Preferences.
- 2) Jika WiFi tersedia maka ESP32 menghubungkan ke jaringan (STA mode).
- 3) Jika WiFi gagal maka ESP32 membuat AP sendiri dengan nama ESP32-Config.
- 4) MQTT diinisialisasi untuk berlangganan topik:
  - smarthome/esp32/gorden/keluarga
  - smarthome/esp32/gorden/kamar
- 5) Web server dijalankan pada port 80, menampilkan dashboard:
  - Status WiFi, IP, SSID
  - Status gorden dan LED
  - Kontrol buka/tutup gorden
  - Form input WiFi configuration
- 6) Interaksi:
  - Tekan tombol di web maka ESP32 mengubah status gorden dan publish MQTT.
  - Terima pesan MQTT maka ESP32 mengubah status gorden sesuai perintah.
  - LED menyala jika salah satu gorden terbuka.

### 3. Diagram Sistem



Penjelasan:

- ESP32 bertindak sebagai controller utama dan interface ke gorden.
- Web dashboard dan MQTT broker bisa mengontrol ESP32 secara paralel.
- LED menunjukkan status gorden: ON jika salah satu gorden terbuka.

### 4. Potongan Kode Utama

- 1) Library : Ini memastikan sistem bisa mengontrol gorden melalui web, serta menerima perintah dari platform MQTT secara real-time

```
#include <WiFi.h>
#include <WebServer.h>
#include <Preferences.h>
#include <ESPmDNS.h>
#include <ArduinoJson.h>
#include <PubSubClient.h>
```

- 2) Definisi Pin dan PWM : Ini memungkinkan motor berjalan dengan duty cycle yang bisa dikontrol, meskipun dalam contoh ini hanya ON/OFF untuk gorden

```
/* ===== PIN ===== */
#define LED_PIN 2
#define BTN_PIN 0
```

```

/* ===== PWM ===== */
#define LED_CHANNEL 0
#define LED_FREQ    5000
#define LED_RES      8    // 8-bit (0-255)

```

- 3) Global state dan Pointer : Ini menyimpan status gorden menggunakan boolean global. Pointer digunakan untuk memudahkan pengubahan nilai dari beberapa tempat, seperti interrupt, MQTT, dan web server. Ini contoh manajemen memori yang sederhana tapi efisien, karena tidak perlu menyalin nilai gorden, cukup ubah lewat pointer.

```

String ssid, password;

bool gordenKeluargaState = false;
bool gordenKamarState    = false;

bool *gordenKeluarga = &gordenKeluargaState;
bool *gordenKamar     = &gordenKamarState;

```

- 4) Interrupt : menggunakan interrupt pada tombol untuk membuka atau menutup gorden ruang keluarga. Interrupt ini memastikan respons sistem tetap cepat meskipun ada proses lain berjalan.

```

void IRAM_ATTR toggleGordenKeluarga() {
    *gordenKeluarga = !(*gordenKeluarga);
    Serial.println("[INTERRUPT] Toggle Gorden Ruang Keluarga");
}

```

- 5) Komunikasi MQTT : ESP32 terhubung ke broker MQTT untuk menerima perintah dari platform seperti HiveMQ. Callback MQTT ini mengatur status gorden sesuai topik yang diterima.

```

void mqttCallback(char* topic, byte* payload, unsigned int length) {
    String msg;
    for (uint8_t i = 0; i < length; i++) msg += (char)payload[i];
}

```

```

Serial.printf("[MQTT] %s => %s\n", topic, msg.c_str());

if (String(topic) == "smarthome/esp32/gorden/keluarga") {
    *gordenKeluarga = (msg == "OPEN");
    Serial.println("[MQTT] Gorden Ruang Keluarga " + msg);
}

if (String(topic) == "smarthome/esp32/gorden/kamar") {
    *gordenKamar = (msg == "OPEN");
    Serial.println("[MQTT] Gorden Kamar " + msg);
}
}

```

- 6) RTOS Task untuk MQTT dan LED : menggunakan FreeRTOS untuk manajemen proses. Ada task untuk MQTT agar koneksi selalu terjaga, dan task LED untuk menampilkan status PWM motor.

```

void taskMQTT(void *pv) {
    for (;;) {
        if (WiFi.status() == WL_CONNECTED) {
            if (!mqttClient.connected()) connectMQTT();
            mqttClient.loop();
        }
        vTaskDelay(10 / portTICK_PERIOD_MS);
    }
}

```

- 7) Web Server dan Dashboard : Pembuatan web server dengan halaman dashboard interaktif. Status WiFi, IP, dan gorden ditampilkan real-time, dan pengguna bisa mengontrol gorden langsung dari web.

```

server.on("/", [](){ server.send_P(200,"text/html",webpage); });

server.on("/gorden/keluarga/open", HTTP_POST, [](){
    *gordenKeluarga=true;
    mqttClient.publish("smarthome/esp32/gorden/keluarga","OPEN");
    Serial.println("[WEB] Gorden Ruang Keluarga OPEN");
});

```

```

    server.send(200);
});

server.on("/gorden/keluarga/close", HTTP_POST, [](){
    *gordenKeluarga=false;
    mqttClient.publish("smarthome/esp32/gorden/keluarga","CLOSE");
    Serial.println("[WEB] Gorden Ruang Keluarga CLOSE");
    server.send(200);
});

server.on("/gorden/kamar/open", HTTP_POST, [](){
    *gordenKamar=true;
    mqttClient.publish("smarthome/esp32/gorden/kamar","OPEN");
    Serial.println("[WEB] Gorden Kamar Tidur OPEN");
    server.send(200);
});

server.on("/gorden/kamar/close", HTTP_POST, [](){
    *gordenKamar=false;
    mqttClient.publish("smarthome/esp32/gorden/kamar","CLOSE");
    Serial.println("[WEB] Gorden Kamar Tidur CLOSE");
    server.send(200);
});

```

- 8) Manajemen WiFi : Sistem ini juga menangani koneksi WiFi. Jika gagal, ESP32 otomatis masuk ke mode Access Point sehingga pengguna tetap bisa konfigurasi SSID dan password.

```

void connectWiFi(){
    Serial.println("[WIFI] Connecting...");
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid.c_str(),password.c_str());
    unsigned long t=millis();
    while(WiFi.status()!=WL_CONNECTED && millis()-t<15000){
        delay(300);
    }
    if(WiFi.status()==WL_CONNECTED){

```

```

        Serial.println("[WIFI] Connected to: "+WiFi.SSID());
        Serial.println("[IP] "+WiFi.localIP().toString());
    } else {
        Serial.println("[WIFI] Failed, switching to AP");
        startAPMode();
    }
}

void startAPMode(){
    WiFi.mode(WIFI_AP);
    WiFi.softAP("ESP32-Config");
    Serial.println("[AP MODE] ESP32-Config");
    Serial.println("[IP] "+WiFi.softAPIP().toString());
}

```

Dengan kombinasi interrupt, MQTT, FreeRTOS task, dan manajemen WiFi, sistem ESP32 ini berjalan stabil dan responsive, sehingga sistem ini siap digunakan.