

Tugas 1 IF4051 Pengembangan Sistem IoT

Semester II Tahun 2023/2024

Ujian Tengah Semester : Electronic Wallet



Disusun oleh:

Roby Purnomo (13520106)

Program Studi Teknik Informatika

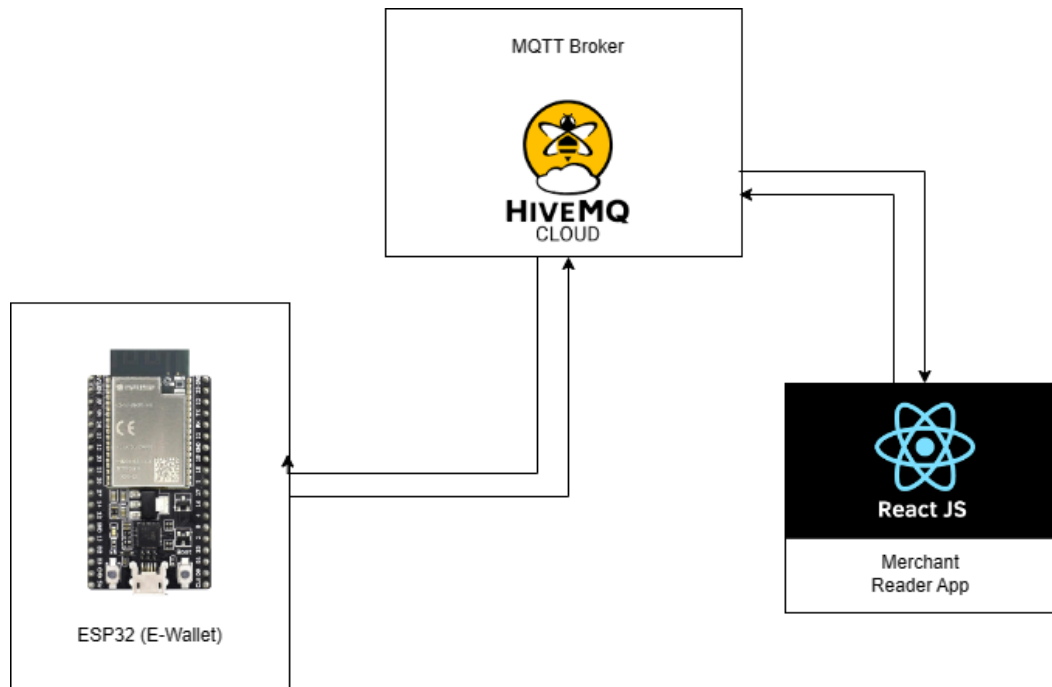
Sekolah Teknik Elektro dan Informatika - Institut Teknologi Bandung

Jl. Ganesha 10, Bandung 40132

2024

Deskripsi Fungsional dan Arsitektur Sistem

Pada tugas ini akan dibuat **Electronic wallet** berbasis hardware yang menggunakan perangkat ESP board dan berbasis teknologi IoT yang dapat digunakan untuk pembayaran secure *cashless transaction*.



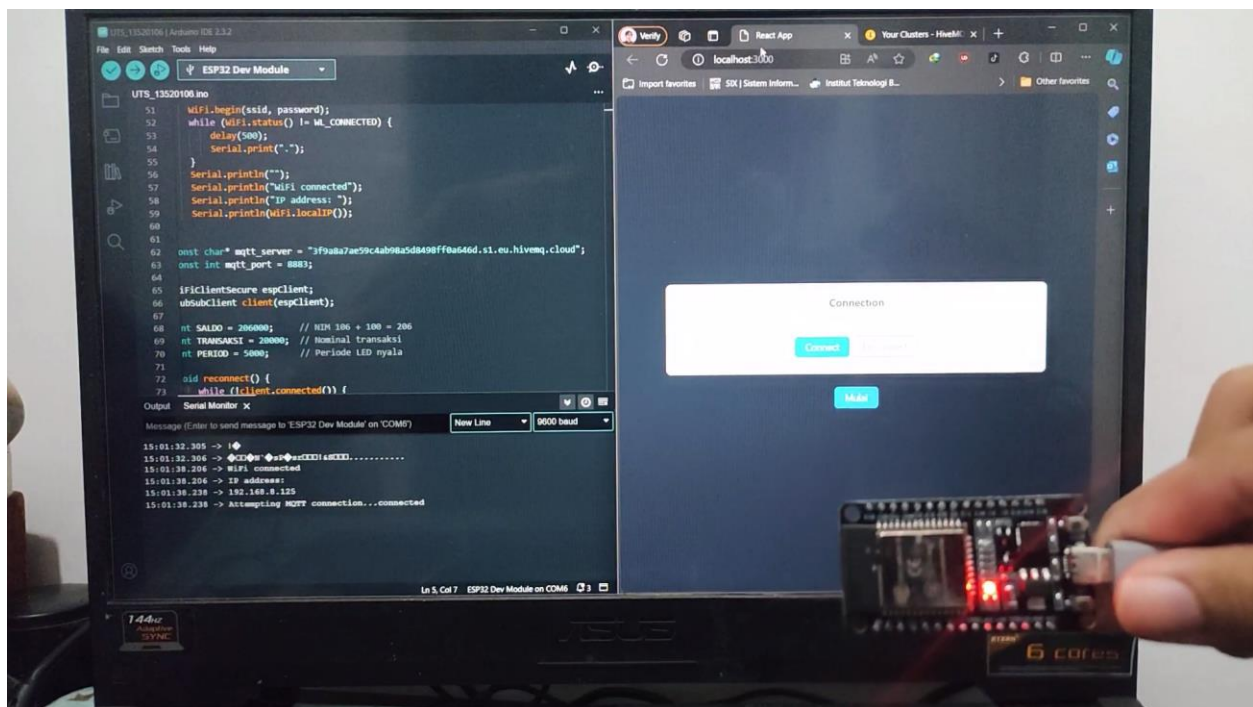
Sistem ini dibuat dengan menggunakan :

1. ESP32 dengan Arduino IDE sebagai E-Wallet
2. React App sebagai Merchant Reader
3. HiveMQ sebagai Server MQTT Broker

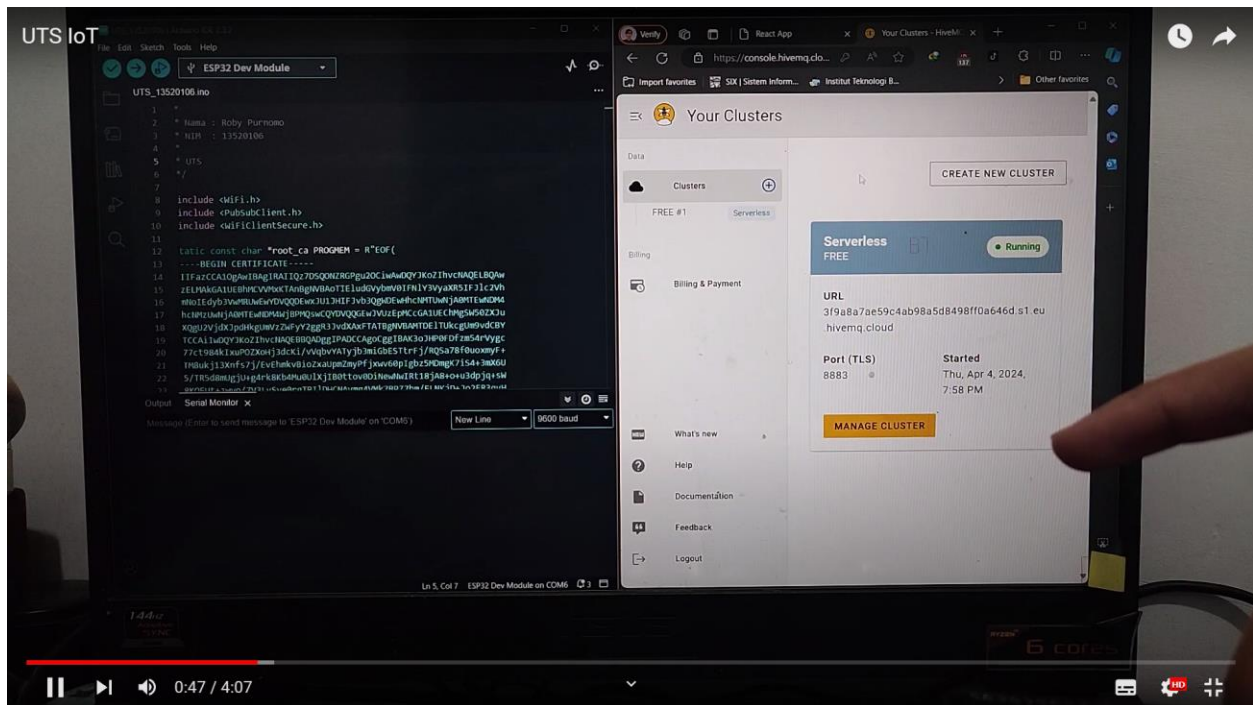
Fitur Sistem

E-Wallet digunakan untuk melakukan transaksi dengan menekan tombol **Boot**, lalu akan melakukan publish kepada broker dan melakukan **transaksi** dengan mengurangi saldo sebesar **Rp 20.000**. Saldo awal E-Wallet adalah **Rp 206.000**. Merchant App sebagai subscriber akan menerima pesan dari E-Wallet melalui broker, jika **berhasil** LED E-Wallet akan menyala selama 5 detik dan Merchant akan mengeluarkan pesan “TRANSAKSI BERHASIL, SISA SALDO Rp. XXX” dan jika **gagal** Merchant akan mengeluarkan pesan “SALDO TIDAK MENCUKUPI”. Di awal penggunaan, E-Wallet akan mengirimkan saldo awal kepada Merchant dan akan menampilkan pesan “SALDO AWAL ANDA Rp. XXX”. Semu pesan akan disertai **timestamp** untuk melakukan tracking.

Rancangan Hardware dan Software



(ESP32 board dan Merchant APP)



(MQTT Broker HiveMQ)

Kode Program

Berikut adalah kode program untuk Arduino IDE

```
/*
 * Nama : Roby Purnomo
 * NIM : 13520106
 *
 * UTS
 */

#include <WiFi.h>
#include <PubSubClient.h>
#include <WiFiClientSecure.h>

static const char *root_ca PROGMEM = R"EOF(
-----BEGIN CERTIFICATE-----
MIIFazCCA10gAwIBAgIRAIQz7DSQONZRGpGu20CiwAwDQYJKoZIhvcNAQELBQAw
TzELMAkGA1UEBhMCVVMxKTAnBgNVBAoTIEludGVybmV0IFN1Y3VyaXR5IFJlc2Vh
cmNoIEdyb3VwMRUwEwYDVQQDEwxJU1JHIFJvb3QgWDEwHhcNMTUwNjA0MTEwNDM4
WhcNMzUwNjA0MTEwNDM4WjBPMQswCQYDVQQGEwJVUzEpMCcGA1UEChMgSW50ZXJu
ZXQgU2VjdXJpdHkgUmVzZWZyY2ggR3JvdXAxFtATBgNVBAMTDElUkcGUm9vdCBY
MTCCAiIwDQYJKoZIhvcNAQEBBQADggIPADCCAgoCggIBAK3oJHP0FDfzm54rVygC
h77ct984kIxuPOZXoHj3dcKi/vVqbvYATyjb3miGbESTtrFj/RQSa78f0uoxmyF+
0TM8ukj13Xnfs7j/EvEhmkvBioZxaUpmZmyPfjxwv60pIgbz5MDmgK7iS4+3mX6U
A5/TR5d8mUgJU+g4rk8Kb4Mu0ULXjIB0ttov0DiNewNwIRt18jA8+o+u3dpjq+sW
T8K0Eut+zwvo/7V3LvSye0rgTBIldHCNAymg4VMk7BPZ7hm/ELNKjD+Jo2FR3qyH
B5T0Y3HsLuJvW5iB4YlcnHlsdu87kGJ55tukmi8mxdAQ4Q7e2RCOFvu396j3x+UC
B5iPNgiV5+I3lg02dZ77DnKxHZu8A/lJBdiB3QW0KtZB6awBdpUKD9jf1b0SHzUv
KBds0pjBqAlkd25HN7r0rFleaJ1/ctaJxQZBKT5ZPt0m9STJEadao0xAH0ahmbWn
OlFuhjuefXKnEgV4We0+UXgVCwOPjdAvBbI+e0ocS3MFEVzG6uBQE3xDk3SzynTn
jh8BCNAw1FtxNrQHusEwMFxIt4I7mKZ9YIqioymCzLq9gwQbooMDQaHwBfEbwrBw
qHyG00aoSCqI3Haadr8faqu9GY/rOPNk3sgrDQoo//fb4hVC1CLQJ13hef4Y53CI
rU7m2Ys6xt0nUW7/vGT1M0NPAGMBAAGjQjBAMA4GA1UdDwEB/wQEAwIBBjAPBgNV
HRMBAf8EBTADAQH/MB0GA1UdDgQWBRR5tFnme7b15AFzgAiIyBpY9umbbjANBgkq
hkiG9w0BAQsFAAOCAgEAVR9YqbyyqFDQDLHYGmkGjYkIrGF1XIpU+ILlaS/V9lZL
ubhzEFnTIZd+50xx+7LSYK05qAvqFyFWhfFQDlnrzuBZ6brJFe+GnY+EgPbk6ZGQ
3BebYhtF8GaV0nxvwuo77x/Py9auJ/GpsMiu/X1+mvoiB0v/2X/qkSsisRcOj/KK
NftY2PwByVS5uCbMioGziUwthDyC3+6WVwW6LLv3xLfHTjuCvjHIInNzktHCgKQ5
ORAZI4JMPJ+Gs1WYHb4phowim57iazT0XoJwTdwJx4nLCgdNb0hdjsnvzqvHu7Ur
TkXWStAmzOVyyghqpZXjFaH3p03JLF+l+/sKAIuvtd7u+Nxe5AW0wdeR1N8NwdC
jNPElpzVmbUq4JUagEiUTDkHszxHpFKVK7q4+63SM1N95R1NbdWhscdCb+ZAJzVc
oyi3B43njT0Q5y0f+1CceWxG1bQVs5ZufpsM1jq4Ui0/1lvh+wjChP4kqK0J2qxq
4RgqsahDYVvTH9w7jXbyLeiNdd8XM2w9U/t7y0Ff/9yi0GE44Za4rF2LN9d11TPA
```

```

mRGunUHBcnWEvgJBQl9nJEiU0Zsnvgc/ubhPgXRR4Xq37Z0j4r7g1SgEEzwxA57d
emyPxgcYxn/eR44/KJ4EBs+lvDR3veyJm+kXQ99b21/+jh5Xos1AnX5iItreGCc=
-----END CERTIFICATE-----
)EOF";

const char* ssid = "Tselhome-D4BB";
const char* password = "62853673";

void setup_wifi() {
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());
}

const char* mqtt_server = "3f9a8a7ae59c4ab98a5d8498ff0a646d.s1.eu.hivemq.cloud";
const int mqtt_port = 8883;

WiFiClientSecure espClient;
PubSubClient client(espClient);

int SALDO = 206000;    // NIM 106 + 100 = 206
int TRANSAKSI = 20000; // Nominal transaksi
int PERIOD = 5000;     // Periode LED nyala

void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect("ESP32Client1", "robyp", "Rp123456")) {
            Serial.println("connected");
            String tempString = String(SALDO);
            client.publish("esp32/saldo", tempString.c_str());
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println(" try again in 5 seconds");
            delay(5000);
        }
    }
}

```

```

}

void callback(char* topic, byte* payload, unsigned int length) {
    Serial.print("Message arrived [");
    Serial.print(topic);
    Serial.print("] ");
    for (int i = 0; i < length; i++) {
        Serial.print((char)payload[i]);
    }
    Serial.println();
}

#define LED 2
#define BUTTON_PIN 0

// Variables
int lastState = 1;    // last state from the input button
int currentState;     // current state from the input button

void setup() {
    Serial.begin(9600);
    setup_wifi();
    espClient.setCACert(root_ca);
    client.setServer(mqtt_server, mqtt_port);
    client.setCallback(callback);
    reconnect();
    // client.subscribe("topic");

    // Set pin mode
    pinMode(LED, OUTPUT);
    // initialize the push button pin as an input
    pinMode(BUTTON_PIN, INPUT_PULLUP);
}

void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();

    currentState = digitalRead(BUTTON_PIN);

    if (lastState == LOW && currentState == HIGH) {
        if ((SALDO - TRANSAKSI) > 0) {
            SALDO = SALDO - TRANSAKSI;

```

```
String tempString = String(SALDO);
client.publish("esp32/transaksi", tempString.c_str());

Serial.print("SALDO : ");
Serial.println(SALDO);

digitalWrite(LED,HIGH);
delay(PERIOD);
digitalWrite(LED,LOW);
} else {
    client.publish("esp32/transaksi", "-1");

    Serial.print("SALDO : ");
    Serial.println(SALDO);
}

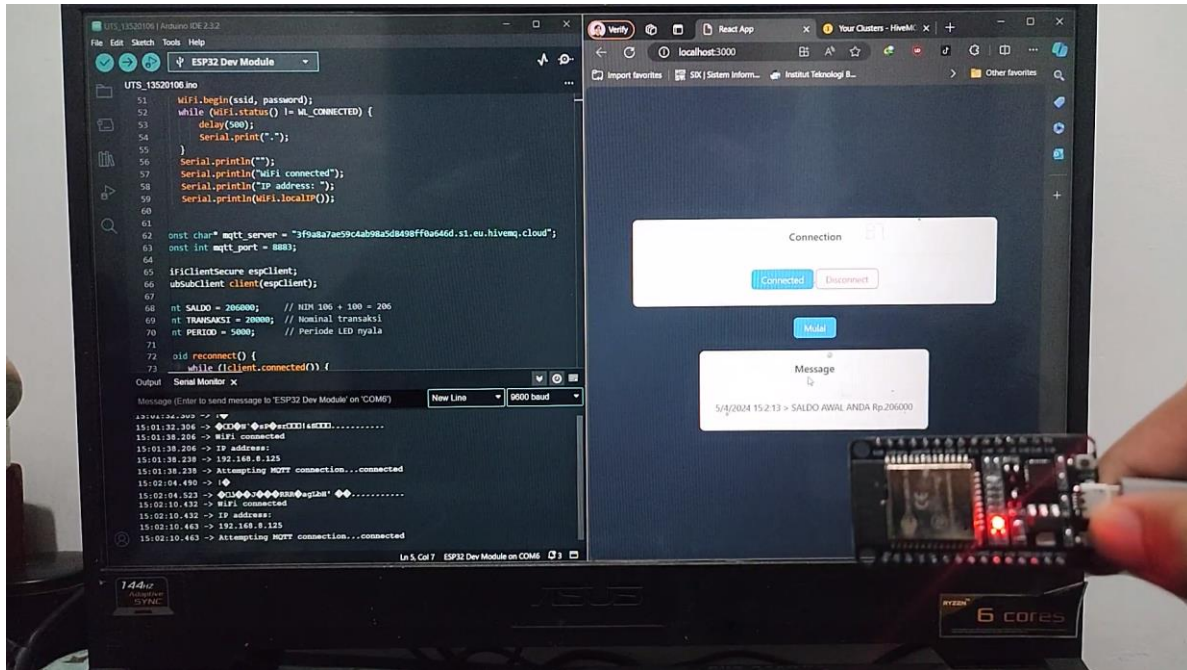
}

// save the the last state
lastState = currentState;
}
```

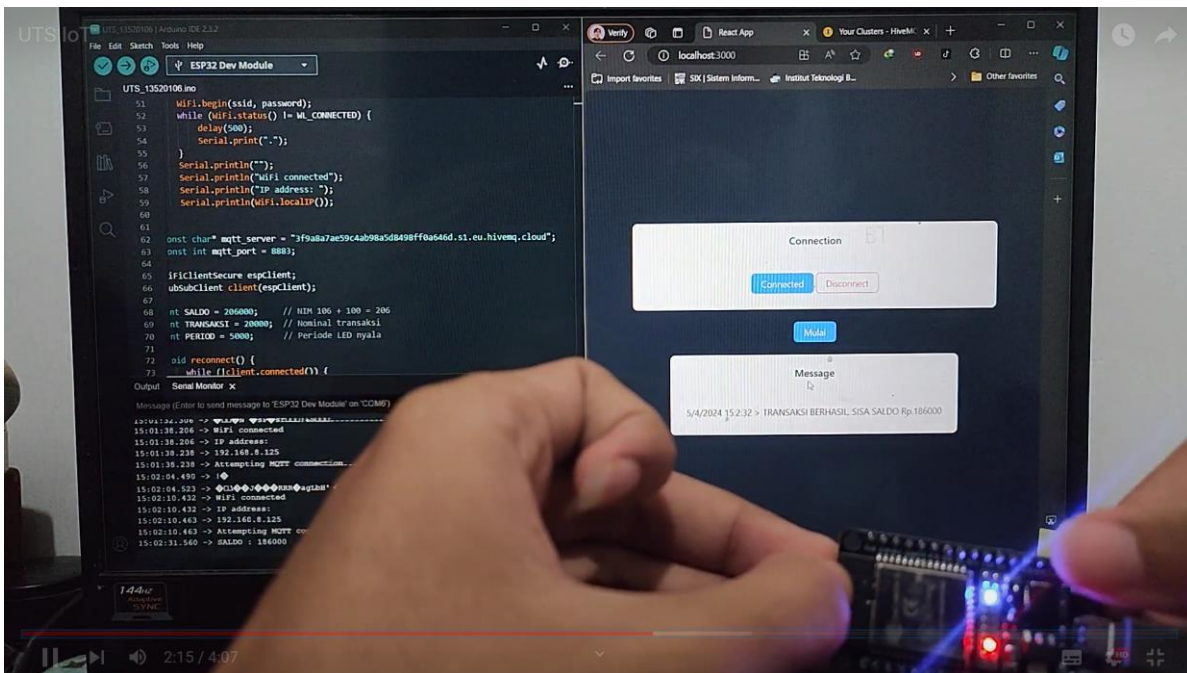
Penjelasan :

Hasil

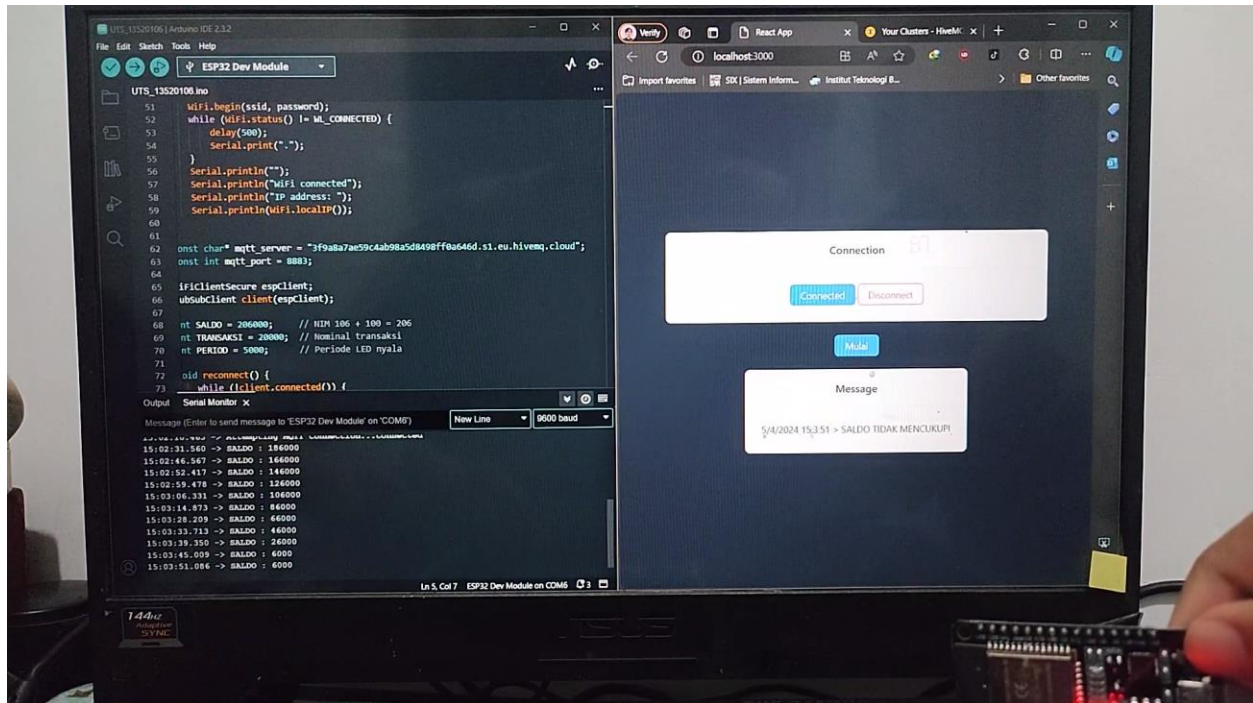
Hasil yang didapatkan pada Merchant adalah sebagai berikut.



(Saldo Awal)



(Transaksi Berhasil)



(Transaksi Gagal)

Simpulan dan Refleksi

Pada tugas kali ini, saya hanya mengerjakan fungsionalitas utama saja dan beberapa tambahan I/O dan web, dikarenakan esp32 saya yang sempat hilang, oleh karena itu saya hanya mengerjakan di beberapa hari terakhir saja. Untuk program yang sudah dibuat berjalan secara lancar tanpa hambatan. Mungkin kedepannya akan lebih hati-hati dan mungkin mengerjakan di awal-awal tugas diberikan agar tidak terulang lagi. Selain itu, menurut saya tugas ini sangat asyik dan dapat berkembang lebih jauh lagi jika ada waktu lebih.

Link Source Code

Berikut adalah link untuk source code

[robypurnomo/uts-iot \(github.com\)](https://github.com/robypurnomo/uts-iot)

Video Demonstrasi

Untuk lebih detail demonstrasinya dapat dilihat pada link video berikut ini.

[UTS IoT \(youtube.com\)](#)