BAB III
IMPLEMENTASI

3.1. Analisis Kebutuhan

Untuk memulai analisis kebutuhan penulis akan melakukan wawancara ke beberapa narasumber lalu membuat hasil dari wawancara tersebut menjadi sebuah Use Case yang akan menjadi kebutuhan utama yang akan di terapkan pada penelitian ini.

3.1.1 Hasil Wawancara

Terdapat 5 narasumber yang terlibat di wawancara ini, kelima narasumber ini terdiri dari anggota organisasi mahasiswa dan dosen pembimbing. Wawancara ini bertujuan untuk mendapatkan pandangan mengenai proses login yang berjalan saat ini, dan kekhawatiran terhadap keamanan khususnya saat proses login.

a) Proses *Login* Saat Ini

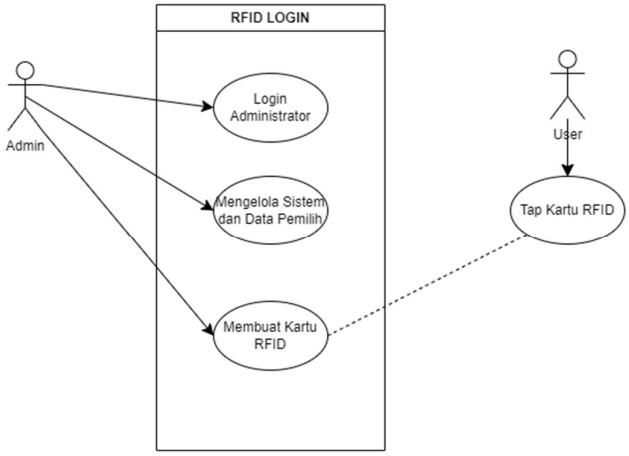
2 narasumber berpendapat bahwa proses login saat ini sudah cukup aman untuk di gunakan, dan merasakan kekhawatiran akan kekurangan kinerja e-voting dengan menggunakan kartu RFID, namun 3 narasumber lainnya mendukung dan merasa bahwa sistem passwordless login RFID ini dapat menambah keamanan dan kecepatan dalam menggunakan sistem e-voting.

b) Kebutuhan

Narasumber mengungkapkan bahwa sistem login RFID membutuhkan sebuah proses yang dimana user dapat berinteraksi langsung dengan menggunakan kartu RFID.

3.1.2 *Use Case*

Berdasarkan hasil wawancara yang sudah di lakukan sebelumnya. Penulis mendapatkan hasil analisis kebutuhan yang di bagi menjadi dua yaitu proses login dan proses register yang di gambarkan melalui Use Case.



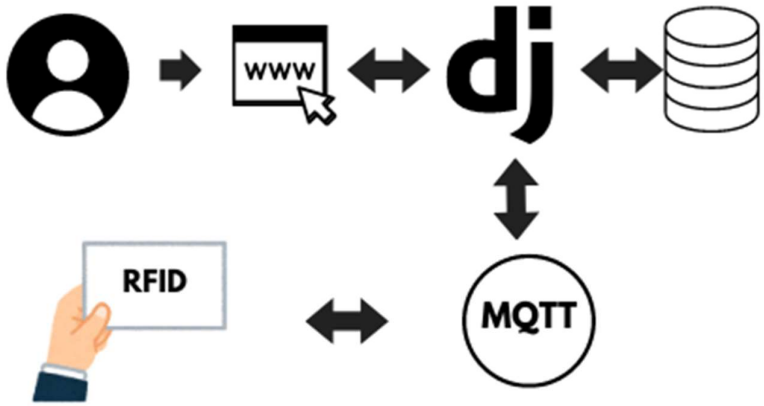
Gambar 3. 1 *Use Case Diagram*

Pada Gambar 3. 1 dapat dilihat terdapat use case login dimana ada 2 aktor yaitu administrator, dan user. Administrator mempunyai kebutuhan yaitu mengelola data pemilih serta membuatkan kartu RFID dengan menulis data ke dalam kartu rfid melalui sistem django. Sementara itu user memiliki kebutuhan untuk dapat login di alat RFID secara langsung dengan menggunakan kartu RFID tanpa menggunakan proses penginputan username dan password.

3.2. Desain (Perancangan)

3.2.1. Desain Arsitektur Sistem

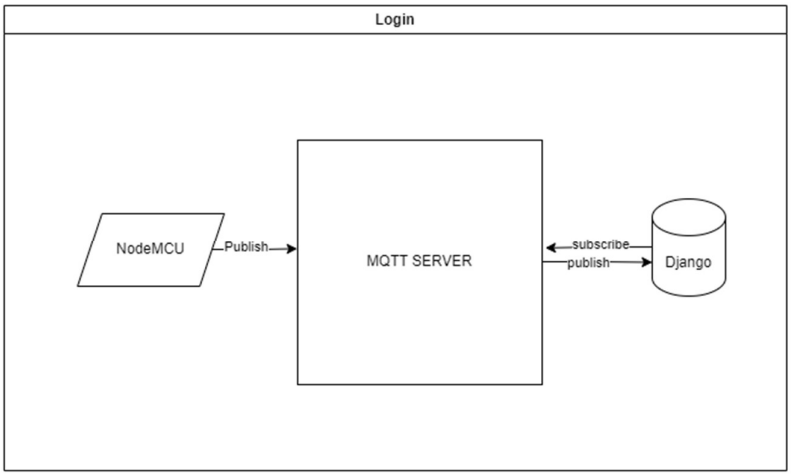
Perancangan ini dilakukan agar desain dari sistem informasi dan analisis kebutuhan penelitian dapat saling terhubung(Adi Kurniawan et al., 2022). Adapun desain arsitektur sistem pada penelitian ini dapat di lihat pada gambar 3.2.



Gambar 3. 2 Desain Sistem

Berdasarkan desain arsitektur sistem, program Django berfungsi untuk menerima dan menjalankan perintah dari situs web yang digunakan oleh admin. Selanjutnya, Django akan menghubungkan situs web dengan server database. Server database akan melakukan query CRUD (Create, Read, Update, Delete) pada database MySQL sesuai dengan perintah dari program Django. Melalui protokol MQTT, akan dilakukan proses registrasi dengan mengirimkan data dari program Django untuk dituliskan ke kartu RFID. Protokol MQTT juga akan mengirimkan data dari kartu RFID ke program Django untuk proses login.

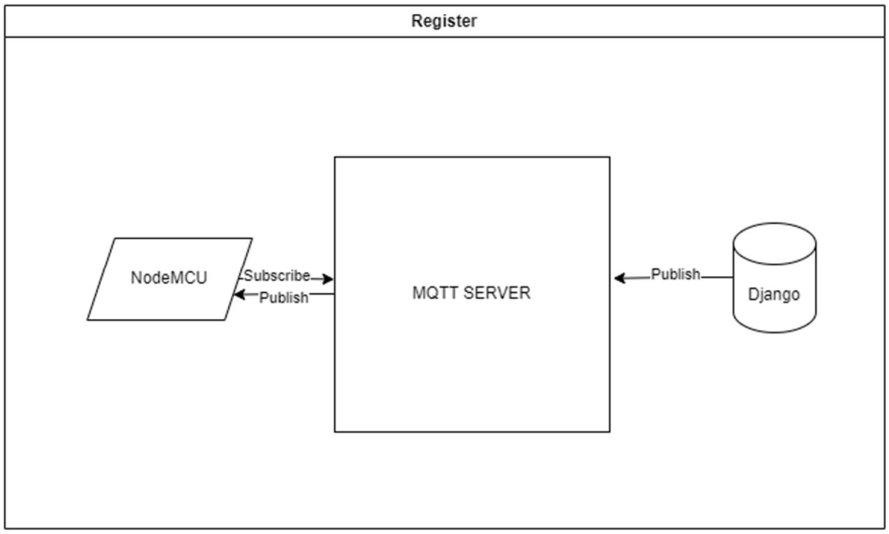
3.2.2. Desain Ptotokol MQTT



Gambar 3. 3 Desain *Login Server MQTT*

Gambar 3. 3 menunjukkan proses MQTT yang terjadi saat sistem login sedang berlangsung. Pada proses ini, terdapat NodeMCU yang berperan sebagai publisher yang akan

mengunggah data ke server MQTT. Selanjutnya, server MQTT akan meneruskan data tersebut ke database dalam program Django.

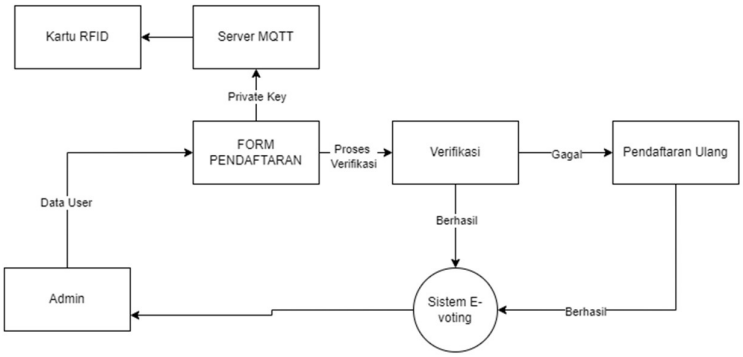


Gambar 3. 4 Desair Registerasi *Server MQTT*

Sedangkan Gambar 3. 4 menunjukkan proses registrasi yang merupakan kebalikan dari proses login, di mana NodeMCU berperan sebagai subscriber dan MySQL menjadi publisher.

3.2.3. Desain Aliran Data

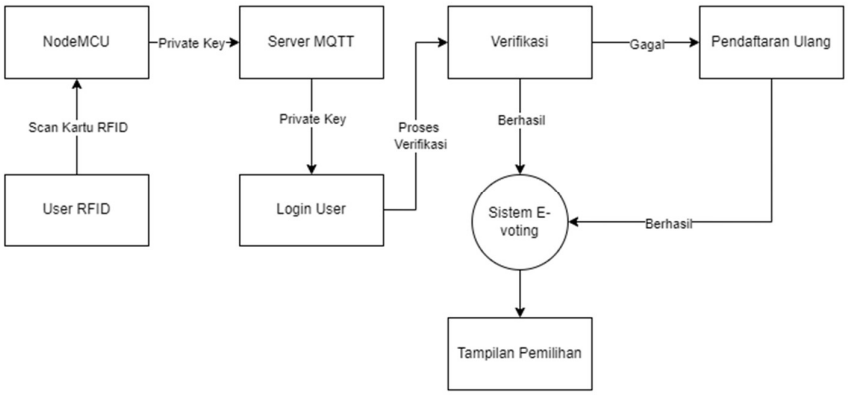
Desain Aliran Data akan di ilustrasikan menggunakan Data Flow Diagram(DFD) yang terdiri dari 2 diagram Berikut adalah diagram yang sudah di buat:



Gambar 3. 5 DFD Aliran Data Proses Registrasi

Dari Gambar 3. 5 Proses Registrasi User dimulai dari admin yang memasukan data user ke dalam form pendaftaran, sesudah itu admin akan menempelkan kartu rfid ke rfid reader untuk menuliskan data yang akan di registrasikan ke database server, setelah itu sistem akan

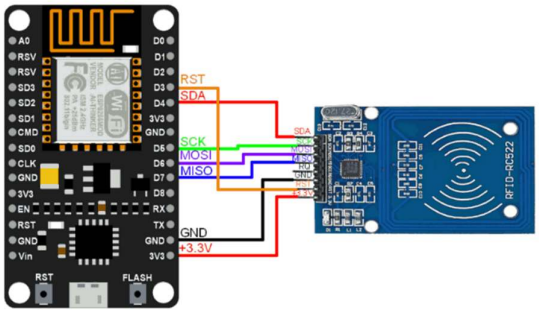
mengverifikasi data tersebut, apa bila data berhasil terverifikasi maka data akan masuk ke database server, apa bila tidak maka akan di lakukan pendaftaran ulang



Gambar 3. 6 DFD Aliran Data Proses *Login User*

Dari Gambar 3. 6, proses *login* pengguna dimulai ketika pengguna RFID menempelkan kartu RFID ke alat pembaca RFID untuk membaca data yang ada di dalam kartu tersebut. Data kemudian akan menjalani proses verifikasi untuk memastikan apakah data tersebut ada dalam database pengguna RFID atau tidak. Jika data tidak ditemukan dalam database pengguna RFID, verifikasi dianggap gagal dan *login* harus diulang. Namun, jika data ditemukan dalam database pengguna RFID, verifikasi berhasil dan pengguna dapat melanjutkan untuk melakukan voting.

3.2.4. Desain Rangkaian Perangkat Keras



Gambar 3. 7 Desain Rangkaian Perangkat Keras

Gambar 3. 7 adalah desain gambaran asli rangkaian kabel yang akan di implementasikan ke perangkat yang ada

NodeMCU	*RFID Reader*
3v3	3v3
D3	RST
Gnd	G
D6	Miso
D7	Mosi
D5	SCK
D4	SDA

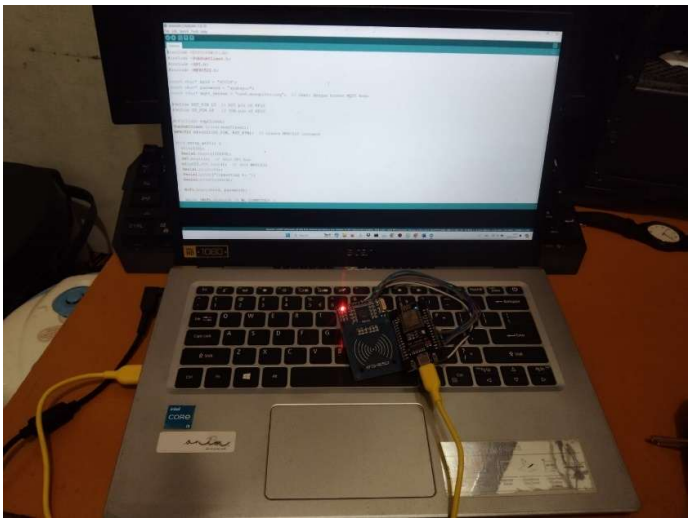
Tabel 2. 1 Desain Konfigurasi Kabel

Dapat di lihat pada tabel 2, terdapat 7 pin yang di gunakan yaitu 3v3 yang terhubung dengan 3v3, D3 dengan RST, Gnd dengan G, D6 dengan Miso, D7 dengan Moso, D5 dengan SCK, dan D4 dengan SDA.

3.3. Implementasi

3.3.1. Implementasi Hardware

Tahap ini di mulai dengan menghubungkan NodeMCU dan RFID reader seperti gambar 3. 7. lalu Konfigurasinya di sesuaikan dengan Tabel 2.



Gambar 3. 8 Implementasi Hardware

Pada gambar 3. 8 terlihat proses menghubungkan perangkat NodeMCU ESP8266 ke RFID reader lalu di hubungkan ke leptop.

3.3.2 Implementasi Software

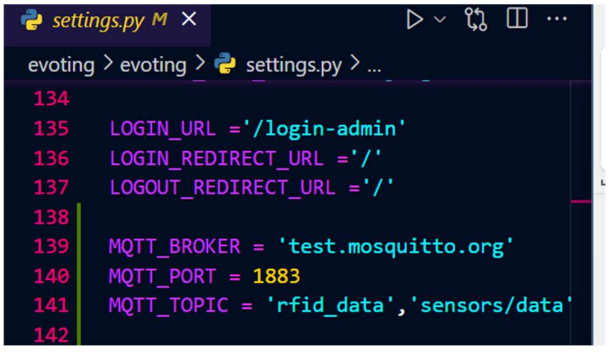
Implementasi perangkat lunak dalam penelitian ini dibagi menjadi tiga tahapan: program Django, program Arduino, dan konfigurasi server MQTT. Ketiga tahapan ini saling terhubung dan saling membutuhkan untuk membangun program yang berfungsi dengan baik.

a) Program Django

Konfigurasi di mulai dengan mengimport paho-mqtt server ke visual studio code dengan memasukan perintah, “pip install paho-mqtt client” dengan menginstall paho-mqtt client, django dapat terhubung ke server mqtt yang akan di gunakan,

Gambar 3. 9 Instalasi paho-mqtt client

Setelah menginstall paho-mqtt client penulis akan mengkonfigurasi program setting.py untuk mendaftarkan server mqtt yang di gunakan yaitu server ‘test.mosquitto.org’ beserta port dan topik yang di gunakan. Berikut adalah hasil konfigurasinya.



Gambar 3. 10 Konfigurasi Setting.py

Setelah melakukan konfigurasi maka program sudah siap untuk di implementasikan ke dalam proses registrasi dan login, berikut adalah hasil implementasinya.

```python
def createPemilih(request):
    if request.method == 'POST':
        form = PemilihForm(request.POST)
        if form.is_valid():
            # Generate RSA keys
            public_key, private_key = generate_keys()

            # Save the form but do not commit to the database yet
            pemilih = form.save(commit=False)
            pemilih.public_key = public_key
            pemilih.private_key = private_key

            # Generate DSA keys
            private_dsa_key, public_dsa_key = generate_dsa_keys()
            if private_dsa_key is None or public_dsa_key is None:
                return render(request, 'back/home/pemilih.html', {'form': form, 'error': 'Failed to generate DSA keys'})

            # Save DSA keys to the pemilih object
            pemilih.dsa_private_key = private_dsa_key.decode('utf-8')
            pemilih.dsa_public_key = public_dsa_key.decode('utf-8')
            pemilih.save()

            # MQTT Client setup and publish
            mqtt_client = mqtt.Client()
            mqtt_client.connect("test.mosquitto.org", 1883, 60)  # Replace with your MQTT broker

            # Publish NIM to topic rfid/nim
            nim_json = json.dumps({"nim": pemilih.nim})
            mqtt_client.publish("rfid/nim", nim_json)
```

```python
            # Split private DSA key into 16-byte chunks
            chunk_size = 16
            chunks = [private_dsa_key[i:i + chunk_size] for i in range(0, len(private_dsa_key), chunk_size)]
            total_chunks = len(chunks)

            # Publish each chunk to a separate topic
            for i, chunk in enumerate(chunks):
                chunk_json = json.dumps({"chunk_index": i, "chunk_data": chunk.decode('utf-8'), "total_chunks": total_chunks})
                topic = f"rfid/private_key/chunk_{i}"
                mqtt_client.publish(topic, chunk_json)

            mqtt_client.disconnect()

            return redirect('pemilih')
    else:
        form = PemilihForm()

    return render(request, 'back/home/pemilih.html', {'form': form, 'error': 'Error Data Sudah Terdaftar'})
```

Gambar 3. 11 Program Registrasi

Pada Gambar 3. 11 dapat di lihat program yang berfungsi untuk mengirimkan data hasil registrasi berupa *private key* DSA ke topik rfid/*private_key* di *server* mqtt dengan cara membagikan *private key* DSA yang awalnya memiliki ukuran 2048 bit menjadi 76 *chunk* dengan masing masing chunk berukuran 16 byte sehingga dapat di tuliskan ke kartu RFID. Selain itu code ini juga mengirimkan nim ke topik rfid/nim, nim ini berfungsi sebagai kunci yang nantinya akan di gunakan untuk melakukan proses login.

```python
def on_message(client, userdata, msg):
    topic = msg.topic
    payload = msg.payload.decode()
    userdata[topic] = payload

def pemilih_login(request):
    if request.method == 'POST':
        # Set up MQTT client to receive NIM and private key parts
        userdata = {}
        mqtt_client = mqtt.Client(userdata=userdata)
        mqtt_client.on_message = on_message
        mqtt_client.connect("test.mosquitto.org", 1883, 60)  # Replace with your MQTT broker
        mqtt_client.subscribe("rfid/nim")
        mqtt_client.subscribe("rfid/private_key")

        # Start the MQTT loop to process network traffic and dispatch callbacks
        mqtt_client.loop_start()

        # Wait for the messages to be received
        timeout = 5  # seconds
        start_time = time.time()
        while time.time() - start_time < timeout and ("rfid/nim" not in userdata or "rfid/private_key" not in userdata):
            time.sleep(0.1)  # sleep a short while to avoid busy-waiting

        mqtt_client.loop_stop()
        mqtt_client.disconnect()

        nim = userdata.get("rfid/nim", None)
        private_key = userdata.get("rfid/private_key", None)

        if nim:
            try:
                pemilih = Pemilih.objects.get(nim=nim)  # Assuming you have a field for NIM
                request.session['pemilih_id'] = pemilih.id
                request.session['pemilih_nim'] = pemilih.nim
                request.session['pemilih_nama'] = pemilih.nama
```

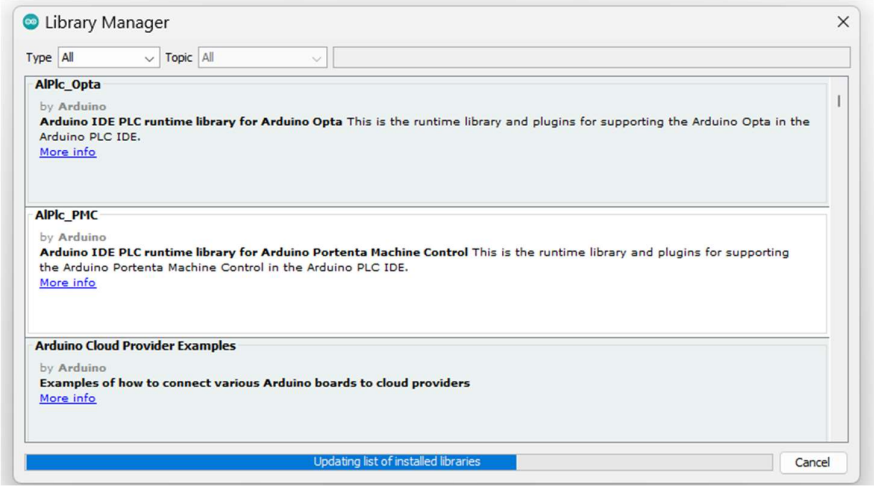
```python
                # Store private key for voting process
                if private_key:
                    request.session['pemilih_private_key'] = private_key

                return redirect('home')
            except Pemilih.DoesNotExist:
                return render(request, 'accounts/login.html', {'error': f'No Pemilih matches the given NIM'})
        else:
            return render(request, 'accounts/login.html', {'error': 'Failed to retrieve NIM from MQTT'})

    return render(request, 'accounts/login.html')
```

Gambar 3. 12 Program Login User

Pada Gambar 3. 12 dapat di lihat program yang berfungsi untuk melakukan proses login, dengan mengambil *private key* yang sudah di kirimkan oleh nodeMCU ke topik yang berada di *server* mqtt dengan melakukan tap kartu RFID di RFID reader.

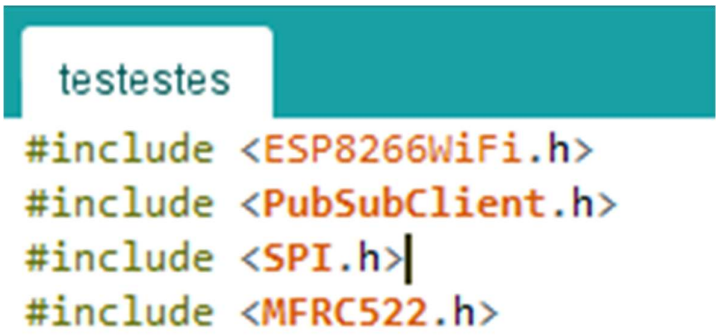
b) Program Arduino IDE

Konfigurasi dimulai dengan menghubungkan NodeMCU ESP8266 ke leptop lalu menginstall beberapa library yang akan di gunakan pada software Arduino IDE seperti ESP8266WiFi.h yang berperan untuk menghubungkan ESP8266 ke wifi, PubsucClient.h berfungsi untuk menghubungkan arduino IDE ke *server* MQTT, SPI.h, dan MFRC522.h berfungsi untuk menghubungkan RFID reader ke arduino IDE.



Gambar 3. 13 Instalasi *Library Arduino IDE*

Setelah melakukan instalasi, *library* akan di *import* pada program Arduino dengan menggunakan perintah *include* agar bisa di gunakan. Seperti yang terlihat pada Gambar 3. 14.



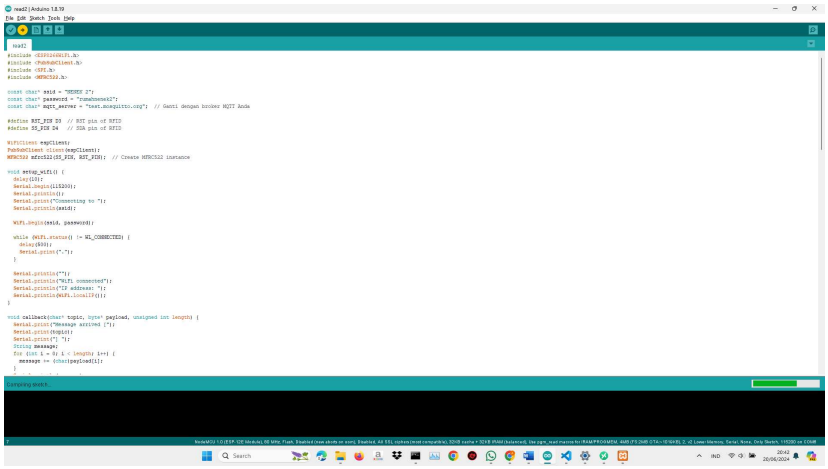
Gambar 3. 14 *Import Library* Yang Di Butuhkan

Setelah semua konfigurasi di lakukan, maka selanjutnya yang akan di lakukan adalah membuat program untuk mengambil dan mengirim data dari *server* mqtt serta menuliskannya ke kartu RFID.

```
#include <ESP8266WiFi.h>
#include <PubSubClient.h>
#include <SPI.h>
#include <MFRC522.h>
#include <ArduinoJson.h>

const char* ssid = "NENEK 2";
const char* password = "rumahnenek2";
const char* mqtt_server = "test.mosquitto.org";

#define RST_PIN D2  // Pin RST on RFID module
#define SS_PIN D3   // Pin SDA on RFID module

WiFiClient espClient;
PubSubClient client(espClient);

MFRC522 mfrc522(SS_PIN, RST_PIN);  // Instance of MFRC522

const unsigned int MAX_PAYLOAD_SIZE = 2048; // Maximum payload size expected to receive
char keyBuffer[MAX_PAYLOAD_SIZE];
unsigned int keyIndex = 0;
unsigned int expectedChunks = 0;
unsigned int receivedChunks = 0;
bool keyComplete = false; // Flag to indicate the private key has been completely received

void setup_wifi() {
  delay(10);
  Serial.begin(115200);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);

  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
```

Gambar 3. 15 Kode Pemrograman Mekanisme *Registrasi*

Pada Gambar 3. 15 dapat di lihat adalah program untuk menerima semua *private key* yang sudah menjadi beberapa *chunk* dari *server* mqtt lalu menuliskan setiap *chunk* ke dalam setiap block kartu RFID sesuai jumlah *chunk* yang di kirimkan .

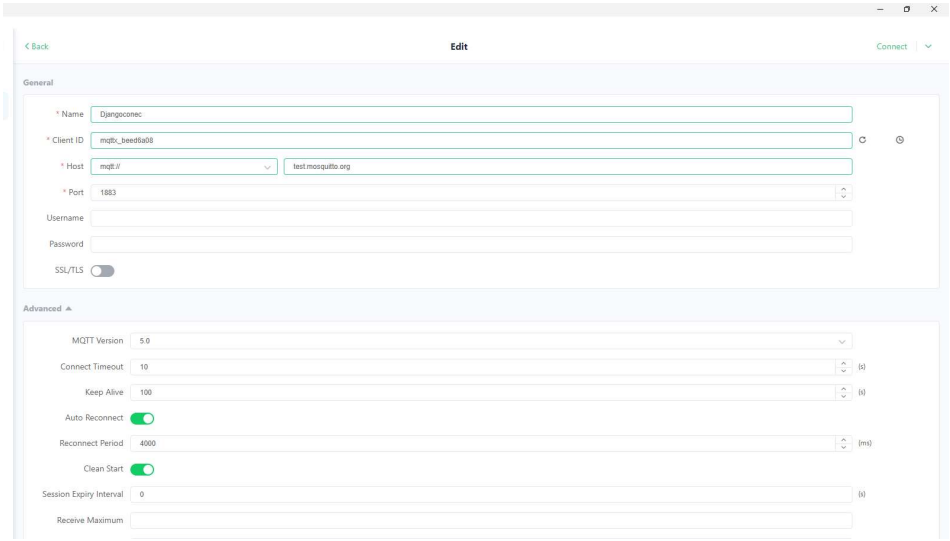


Gambar 3. 16 Program Mekanisme *Passwordless Login RFID*

Pada Gambar 3. 16 yang terlihat adalah code pemrograman login yang berfungsi untuk membaca setiap block yang sudah terisi dengan *private key*, lalu menyatukannya kembali sebelum mengirimnya ke *server* mqtt.

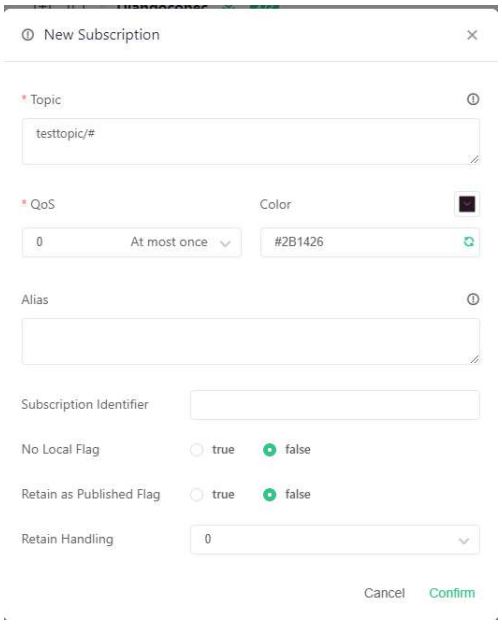
c) Konfigurasi *Server* MQTT

Untuk menggunakan *server* mqtt di perlukan monitoring data, penulis menggunakan mqttx untuk proses monitoring data pada server mqtt.



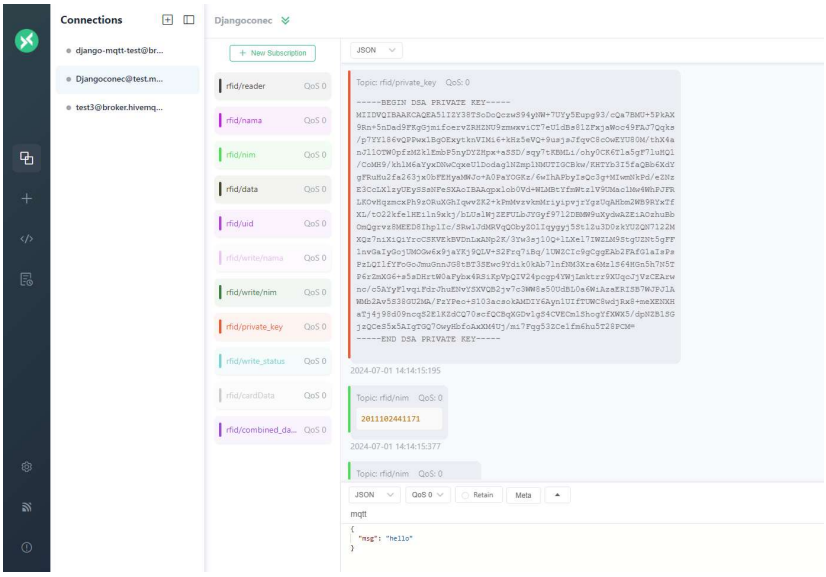
Gambar 3. 17 Konfigurasi MQTTX

Pada Gambar 3. 17 dapat di lihat proses konfigurasi berupa pemberian nama client ID, host, dan port dari mqtt *server* yang akan di gunakan, dapat di lihat penulis menggunakan test.mosquitto.org sebagai *server* mqttnya dan port 1883 sebagai port yang digunakan.



Gambar 3. 18 Konfigurasi Topik

Setelah proses konfigurasi *server* mqtt, akan di lakukan proses pembuatan topik yang nantinya akan menjadi tempat untuk mengirim dan menerima data dari django dan nodeMCU ESP8266. Berikut adalah monitoring penerimaan data pada server mqtt di topik rfid/private_key



Gambar 3. 19 Hasil Monitoring Data

Pada Gambar 3. 19 dapat di lihat tampilan data *private key* dsa dan nim yang masuk ke *server* mqtt maka proses mengirim data ke *server* mqtt di nyatakan sukses.

3.4. Pengujian

3.4.1. Pengujian Alpha

Pada pengujian alpha akan di gunakan metode *Black Box,* dimana pengujian akan berfokus pada fungsi utama dari perangkat lunak dengan memeriksa hasil output dan input apakah sudah sesuai yang di inginkan atau tidak. Berikut adalah skenario pengujian yang akan di lakukan.

Tabel 3. 1 Hasil Pengujian Registrasi

No	Deskripsi Pengujian	Langkah Pengujian	Hasil Diharapkan	Status
1	Registrasi dengan data yang valid	Mengisi form registrasi dengan data yang valid sambil memindai kartu rfid ke RFID reader	Data berhasil di registrasi dan data tertulis ke dalam kartu RFID	[√] Berhasil [ ] Tidak Berhasil

2	Registrasi dengan data yang sama seperti sebelumnya	Mengisi form registrasi dengan data yang tidak valid sambil memindai kartu rfid ke RFID reader	Pesan kesalahan ditampilkan	[√] Berhasil [ ] Tidak Berhasil

Tabel 3. 2 Hasil Pengujian Validasi *Login*

No	Deskripsi Pengujian	Langkah Pengujian	Hasil Diharapkan	Status
1	Login dengan kartu RFID valid	Pindai kartu RFID yang valid pada RFID reader	Pengguna berhasil masuk	[√] Berhasil [ ] Tidak Berhasil
2	Login dengan kartu RFID tidak Valid	Pindai kartu RFID yang tidak valid ke RFID reader	Pesan kesalahan ditampilkan	[√] Berhasil [ ] Tidak Berhasil

Kesimpulannya. Mekanisme yang di terapkan pada sistem RFID terdiri dari 2 mekanisme, yaitu registrasi, dan login. Kedua mekanisme tersebut berhasil melewati pengujian alpha dengan hasil yang di inginkan.

3.4.2. Pengujian Betha

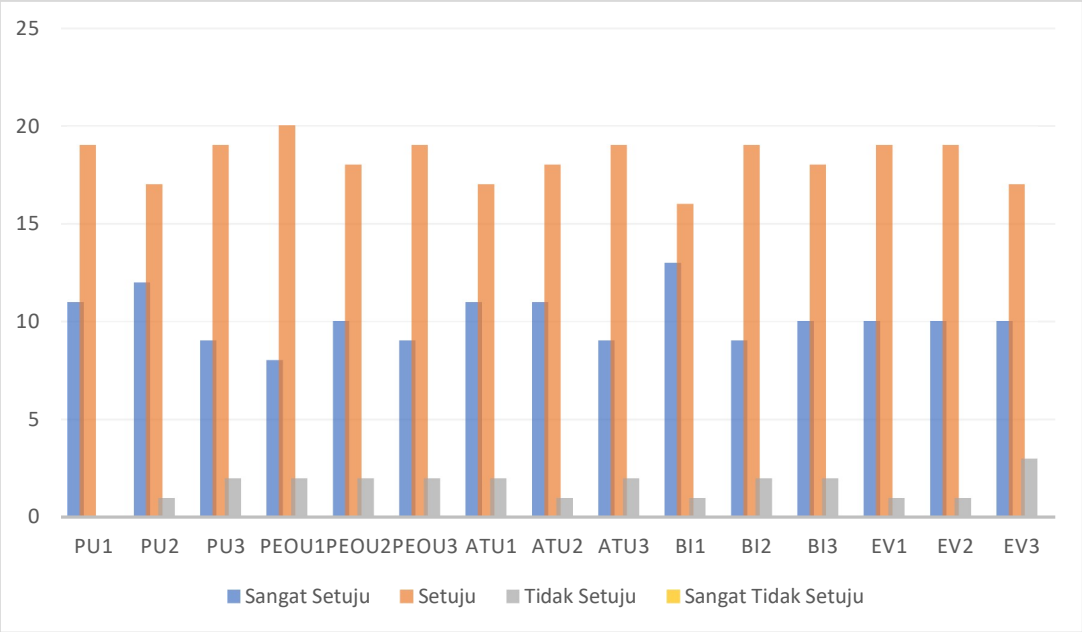
Pada pengujian betha penulis menggunakan metode pengujian FGD dengan membuat quesioner yang menjuru ke *framework* TAM (technology acceptance method) untuk mendapatkan respon yang berfokus pada 5 variabel penting yaitu, *Perceiver Usefelness*(PU), *Perceived Ease of Use*(PEOU), *Attitude Toward Using*(ATU), *Behavioral Intention to Use* (BI), dan *External Variables*(EV) dari 5 variabel tersebut, penulis akan mendapatkan respon terhadap kegunaan, kemudahaan, sikap, keinginan untuk memakai, dan variabel external(Mulyanto et al., 2020). Berikut adalah quesioner yang sudah di berikan.

Tabel 3. 3 Kuesioner

Variable	Pertanyaan	Symbol
Perceived Usefelness (PU)	Anda merasa bahwa menggunakan RFID dalam e-voting akan meningkatkan efisiensi proses pemilihan	PU1
	Anda percaya bahwa sistem e-voting berbasis RFID akan membuat proses pemilihan menjadi lebih cepat ?	PU2

	Anda merasa bahwa penggunaan RFID akan mengurangi kesalahan dalam penghitungan suara	PU3
Perceived Ease of Use (PEOU)	Anda merasa bahwa sistem e-voting berbasis RFID mudah untuk dipelajari	PEOU1
	Anda merasa bahwa antarmuka sistem e-voting berbasis RFID intuitif dan mudah di mengerti	PEOU2
	Anda merasa bahwa mengoprasikan sistem e-voting berbasis RFID tidak memerlukan banyak usaha	PEOU3
Attitude Toward Using (ATU)	Anda merasa senang menggunakan RFID dalam sistem e-voting	ATU1
	Anda merasa puas menggunakan sistem e-voting berbasis RFID	ATU2
	Anda memberikan nilai yang baik secara keseluruhan terhadap penggunaan RFID dalam sistem e-voting	ATU3
Behavioral Intention to Use (BI)	besar kemungkinan Anda akan menggunakan sistem e-voting berbasis RFID di masa mendatang?	BI1
	besar keinginan Anda untuk merekomendasikan sistem e-voting berbasis RFID kepada orang lain?	BI2
	Anda akan memilih sistem e-voting berbasis RFID jika tersedia sebagai pilihan?	BI3
External Variables (EV)	infrastruktur teknologi yang tersedia sudah baik untuk mendukung sistem e-voting berbasis RFID?	EV1
	Anda merasa bahwa lingkungan sosial (misalnya teman, keluarga, kolega) mendukung penggunaan sistem e-voting berbasis RFID?	EV2
	Anda percaya bahwa data pribadi Anda aman saat menggunakan sistem e-voting berbasis RFID?	EV3

Hasil jawaban responden terhadap kinerja dari sistem berdasarkan pertanyaan yang diajukan dapat di lihat pada *chart* berikut :



Gambar 3. 20 *Chart* Hasil Kuesioner

3.4.3. Kesimpulan Hasil Implementasi

Sesuai Jawaban yang sudah di terima dari 30 responden,dapat disimpulkan menggunakan skala likert dengan memberikan bobot ke 4 jawaban yang tersedia, yakni ST(Sangat Setuju) = 4 , S (Setuju) = 3, TS(Tidak Setuju) = 2, STS(Sangat Tidak Setuju) = 1.

Tabel 3. 4 Hasil Pengujian Betha

no	Pertanyaan	Frekuensi Jawaban				Jumlah Skor				Total Skor
		SS	S	TS	STS	SS	S	TS	STS	
1	PU1	11	19	0	0	44	57	0	0	101
2	PU2	12	17	1	0	48	51	2	0	101
3	PU3	9	19	2	0	36	57	4	0	97

4	PEOU1	8	20	2	0	32	60	4	0	96
5	PEOU2	10	18	2	0	40	54	4	0	98
6	PEOU3	9	19	2	0	36	57	4	0	97
7	ATU1	11	17	2	0	44	51	4	0	99
8	ATU2	11	18	1	0	44	54	2	0	100
9	ATU3	9	19	2	0	36	57	4	0	97
10	BI1	13	16	1	0	52	48	2	0	102
11	BI2	9	19	2	0	36	57	4	0	97
12	BI3	10	18	2	0	40	54	4	0	98

13	EV1	10	19	1	0	40	57	2	0	99
14	EV2	10	19	1	0	40	57	2	0	99
15	EV3	10	17	3	0	40	51	6	0	97
Total Akhir Skor										1478
Total Skor Tertinggi (Skor Skala Tertinggi x Jumlah Responden x Jumlah Soal)										1800
Persentase Rata-Rata (Total Akhit Skor / Total Skor Tertinggi x 100)										82%

Dari hasil persentase *table* 6 sistem Passwordles login RFID memiliki nilai persentase rata-rata sebesar 82%, yang berarti bahwa sebagian besar responden setuju atau sangat setuju atas implementasi sistem passwordless login RFID pada sistem e-voting.