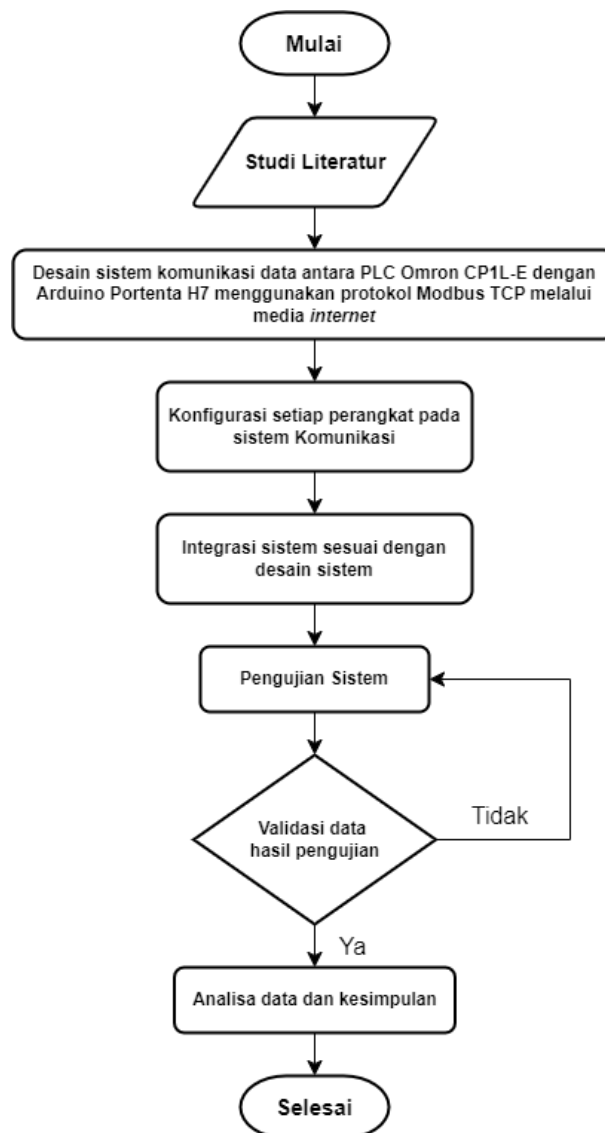


BAB III

METODE PENELITIAN

3.1 Flowchart Penelitian



Gambar 3. 1 *Flowchart* penelitian

Gambar 3.1 menjelaskan alur dari penelitian mengenai perancangan model sistem komunikasi data antara PLC Omron CP1L-E dan Arduino Portenta H7 menggunakan protokol Modbus TCP melalui media *Wi-Fi*. Alur tersebut dapat

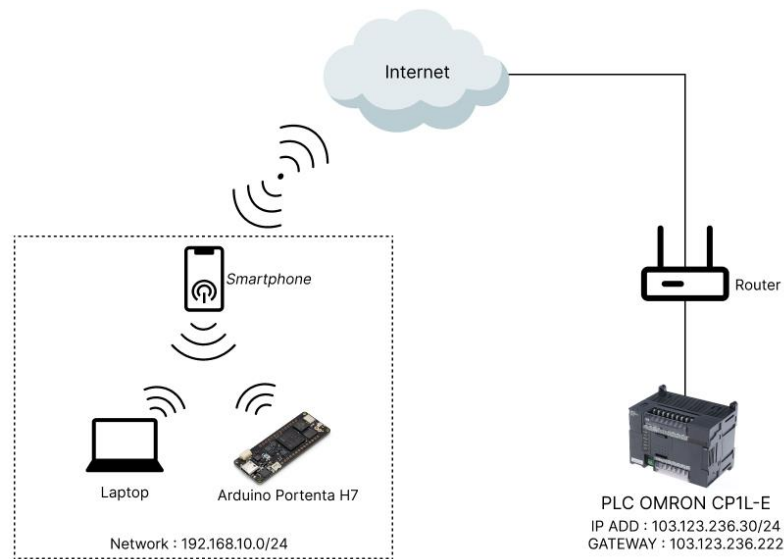
memberikan gambaran tentang metodologi penelitian yang digunakan dalam studi ini.

3.1.1 Studi Literatur

Studi literatur pada diagram alir penelitian ini merupakan kajian teori yang berkaitan dengan tema yang dijadikan acuan untuk topik yang dibahas pada penelitian ini. Referensi yang diambil berasal dari jurnal, buku, dan *website*. Kajian teori terkait penelitian ini antara lain *Programmable Logic Control* khususnya PLC Omron CP1L-E, Arduino Portenta H7, Protokol Modbus TCP/IP, data apa saja yang didapatkan dalam komunikasi data tersebut dan bagaimana pengujian komunikasi data tersebut.

3.1.2 Desain Sistem Komunikasi Data

Tahapan ini akan dilaksanakan setelah mendapatkan referensi di atas. Pada tahapan ini dilakukan realisasi konsep dan teori yang didapat pada tahap sebelumnya. Merealisasikan konfigurasi komunikasi PLC dan Arduino Portenta menggunakan protokol Modbus TCP, serta pengiriman data yang akan dilakukan melalui media internet.



Gambar 3. 2 Diagram sistem komunikasi

Gambar 3.2 merupakan diagram blok yang menjelaskan sistem kerja dari sistem yang akan dibuat menggunakan protokol Modbus TCP. Perincian dari diagram blok tersebut adalah sebagai berikut.

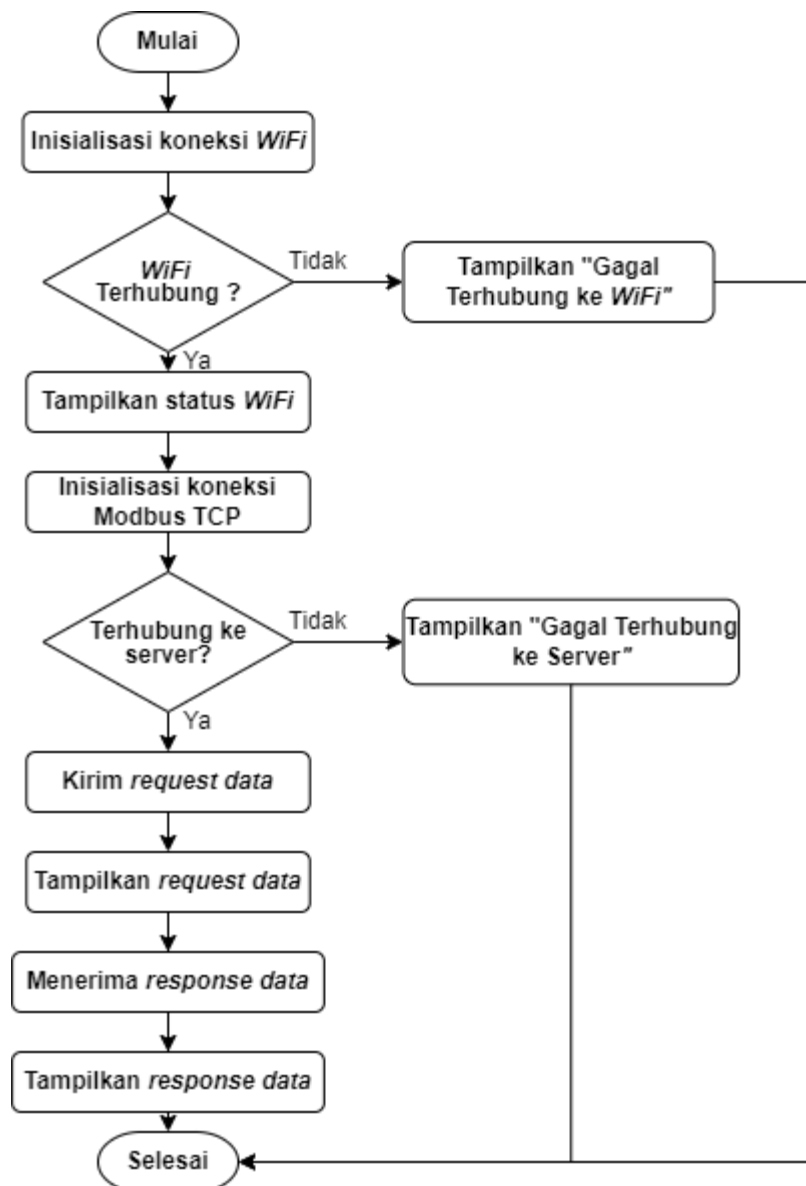
1. Arduino Portenta, berfungsi sebagai perangkat *client* mengirimkan *query* atau *request data* ke PLC untuk diproses. Data tersebut dikirim menggunakan Protokol Modbus TCP melewati internet.
2. PLC Omron CP1L-E, digunakan sebagai perangkat server yang akan menerima *request data* dari *client* dan memproses data tersebut sesuai permintaan. Setelah data diproses, maka PLC akan mengirimkan *response data* ke perangkat yang mengirimkan *request*.
3. Smartphone, digunakan sebagai perangkat penyedia akses internet ke Arduino Portenta.

4. Laptop, digunakan sebagai perangkat *monitoring* dan menangkap data komunikasi yang dilakukan antara Arduino Portenta dan PLC menggunakan *software Wireshark*.

3.1.3 Perancangan Perangkat Lunak

a. Arduino Portenta H7

Proses pengiriman data dari Arduino Portenta H7 ke PLC menggunakan protokol Modbus TCP dengan menggunakan media *internet* dijelaskan dengan diagram alir pada Gambar 3.3. Algoritma dari program pada Arduino Portenta H7 dimulai dengan menginisialisasi koneksi *WiFi* dan koneksi Modbus TCP, pada penelitian ini Arduino bertindak sebagai *Client*. Jika inisialisasi koneksi *WiFi* dan Modbus gagal, maka arduino akan memberi *output* berupa teks berisi pesan error dan program akan berhenti. Jika kedua inisialisasi tersebut berhasil, maka program dilanjutkan dengan mengirimkan *query* atau *request* ke *server* dan menampilkan isi data dari *request* tersebut pada serial monitor. Kemudian Arduino akan menunggu jawaban dari *server* (*response*). Data response yang diterima akan ditampilkan pada serial monitor. Alur pemrograman dari Arduino Portenta H7 dijelaskan pada Gambar 3.3.

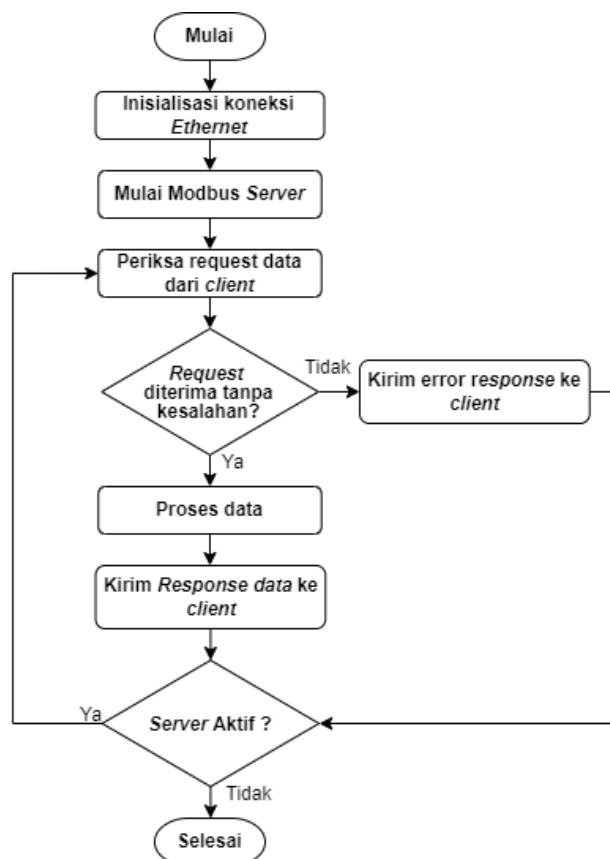


Gambar 3. 3 Flowchart pemrograman Arduino Portenta H7

b. PLC Omron CP1L

Pada sistem komunikasi ini, PLC Omron CP1L bertindak sebagai *Server* pada protokol Modbus TCP. Program PLC diawali dengan inisialisasi koneksi *Ethernet* dan Modbus TCP *Server*. Kemudian perangkat ini menunggu *request* dari *client*. Setelah data request diterima, maka PLC dapat memproses data tersebut dan menyimpannya pada memori. Kemudian PLC akan mengirimkan *response* ke *client*

berupa *function code* dan *data response* atau *exception function code* dan *exception code*. Algoritma pemrograman pada PLC dijelaskan dengan diagram alir pada Gambar 3.4.



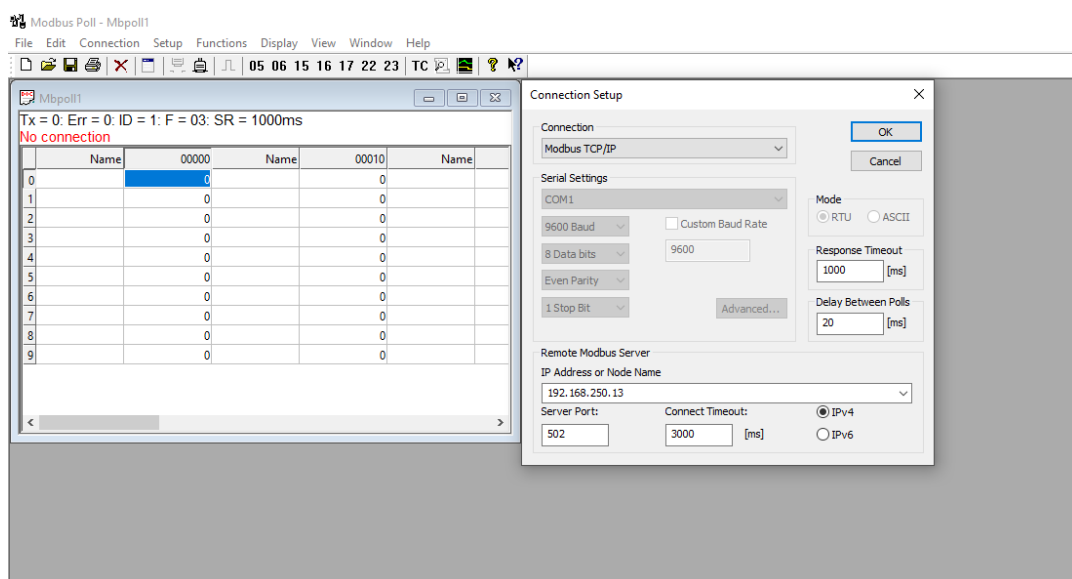
Gambar 3. 4 *Flowchart* Pemrograman PLC

3.1.4 Pengujian Unit

Pengujian unit berfokus pada pengujian perangkat dalam sistem secara individual. Tujuan dari pengujian ini untuk memastikan setiap perangkat dalam sistem berfungsi sesuai dengan tugasnya masing-masing. Pengujian dilakukan dengan mensimulasikan komunikasi Modbus pada PLC dan Arduino.

a. Pengujian PLC Omron CP1L-E

Pengujian komunikasi PLC dilakukan dengan mensimulasikan komunikasi data dengan Protokol Modbus TCP menggunakan *software* ModbusPoll. *Software* ini memiliki kemampuan untuk mensimulasikan komunikasi dengan Protokol Modbus TCP pada setiap perangkat. Tampilan awal dari *Software* ini diperlihatkan pada Gambar 3.5.



Gambar 3. 5 Tampilan *Software* ModbusPoll

Sumber : (Dokumen Pribadi)

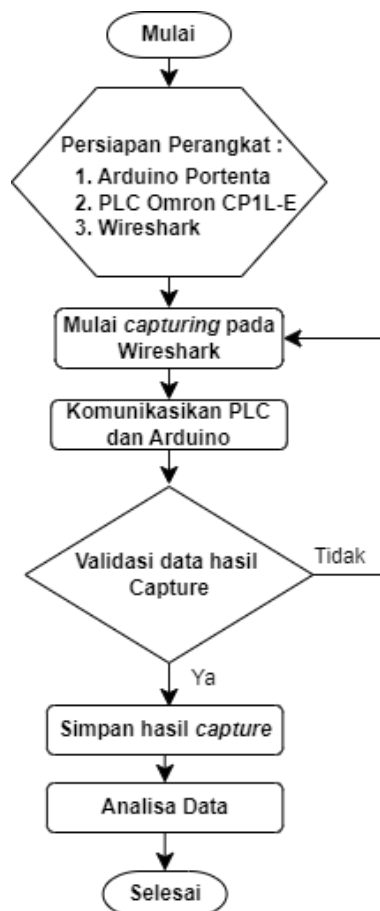
b. Pengujian Arduino Portenta H7

Pengujian pada perangkat Portenta H7 sama halnya dengan pengujian pada PLC Omron CP1L-E. Komunikasi pada Portenta H7 disimulasikan menggunakan ModbusPoll. Pada pengujian ini, Portenta H7 mengirimkan *request* ke *software* tersebut untuk mengetahui apakah Portenta dapat mengirimkan berkomunikasi melalui protokol tersebut sebelum diimplementasikan pada sistem.

3.1.5 Pengujian Sistem

a. Skenario Pengujian Komunikasi Data

Skenario pengujian bertujuan untuk menguji kinerja komunikasi data antara PLC Omron CP1L-E dan Arduino Portenta H7 menggunakan protokol Modbus TCP. Proses pengambilan data pengujian dijelaskan pada Gambar 3.6.



Gambar 3. 6 *Flowchart* pengambilan data pengujian

Dari Gambar 3.6 dapat diketahui bahwa pengambilan data diawali dengan menyiapkan perangkat yang digunakan. Tiga perangkat utama yang digunakan adalah PLC Omron CP1L-E, Portenta H7 dan PC yang sudah terpasang *software*

Wireshark. Semua perangkat harus sudah terhubung ke jaringan sesuai dengan desain dari sistem komunikasi yang telah dibahas sebelumnya.

Kemudian Wireshark dijalankan untuk memulai memonitoring komunikasi data Modbus TCP yang terjadi antara PLC dan Arduino Portenta H7. Filter pada Wireshark dikonfigurasi supaya hanya menangkap paket dari protokol Modbus TCP. Filter yang dipakai pada pengujian ini adalah *port 502*, karena komunikasi berjalan pada *port* tersebut.

Setelah Wireshark siap, maka komunikasi antara kedua perangkat dijalankan. Dalam hal ini, Arduino Portenta yang bertindak sebagai *client* akan mengirimkan *request* atau *query* dengan *function code* 0x16 (Write Multiple Register) dan isi datanya merupakan data *dummy* untuk menuliskan 123 alamat *register* pada PLC. Kemudian *request* tersebut akan diterima oleh PLC Omron CP1L-E untuk diproses. Setelah data diproses, maka PLC akan mengirimkan *Response code* ke Arduino Portenta.

Selanjutnya, data hasil *capture* pada Wireshark divalidasi. Pengujian dapat disebut valid jika waktu pengujian sudah mencapai 24 Jam, atau jika terjadi *bottleneck* sebelum 24 jam. Variasi pengujian berdasarkan pada waktu jeda antar *request* atau *poll rate*. Pengujian dimulai dengan mengkomunikasikan sistem dengan jeda antar *request* 0 ms, 15 ms, 30 ms, 45 ms, 55 ms, 60 ms, dan 65 ms.

Setelah data hasil *capture* divalidasi, maka data tersebut disimpan pada perangkat untuk selanjutnya di analisis berdasarkan variasi-variasi pengujian tersebut. Adanya variasi waktu jeda antar *request* dimaksudkan untuk meneliti

karakteristik parameter-parameter QoS terhadap waktu jeda tersebut dan bagaimana kinerja sistem komunikasi tersebut.

b. Parameter Pengujian

Pengujian komunikasi data dilaksanakan untuk mengetahui kemampuan sistem dalam berkomunikasi menggunakan Protokol Modbus TCP. Pengujian ini berfokus pada komunikasi antara PLC dan Arduino Portenta H7. Pengujian dilaksanakan dengan metode Quality of Service (QoS) (Nuryani et al., 2024).

Quality of Service merupakan metode pengujian kualitas jaringan, parameter yang diuji berdasarkan Standar TIPHON (*Telecommunications and Internet Protocol Harmonization Telecommunications Standards Institute*). Dimana standar ini umum digunakan dalam peengujian kualitas jaringan (Wulandari, 2016). Parameter yang diuji adalah *Throughput*, *Delay (Latency)*, *Jitter*, dan *Packet Loss*.

1. Throughput

Throughput merupakan kecepatan transfer data efektif dimana jumlah total kedatangan paket dalam interval waktu tertentu dibagi interval waktu tersebut (Nisa et al., 2024; Wulandari, 2016). Biasanya, *throughput* memiliki satuan *bits per second (bps)* atau *bytes per second (Bps)*. Persamaan dari parameter ini adalah sebagai berikut.

$$\text{Throughput} = \frac{\text{Total data yang ditransfer}}{\text{Periode pengambilan data}} \dots\dots\dots (3.1)$$

Dari rumus di atas, total data yang ditransfer merupakan ukuran data yang dikomunikasikan barik dari paket *request* maupun *response*. Kategori dari *Throughput* dijelaskan pada Tabel 3.1 berikut.

Tabel 3. 1 Kategori *Throughput*

Kategori	<i>Throughput</i>	Indeks
Sangat bagus	> 2,1 Mbps	4
Bagus	1,2 – 2,1 Mbps	3
Sedang	338 – 1200 Kbps	2
Jelek	0 – 338 Kbps	1

Sumber : (Nisa et al., 2024)

2. Delay (Latency)

Delay (*Latency*) merupakan waktu yang dibutuhkan data untuk menempuh jarak dari perangkat sumber ke perangkat tujuan (Nisa et al., 2024). Pada penelitian ini *delay* yang dipakai merupakan *Round Trip Time (RTT) delay*. Karena sistem kerja Modbus TCP berdasarkan atas *Request* dan *Response*, maka *RTT delay* dapat dipakai untuk menilai berapa cepat responsivitas total dari komunikasi (Putranto, Ichsan Edi , Maman Abdurrohman, 2016). Nilai rata-rata *delay* ini dapat dicari dengan menggunakan persamaan 3.2.

$$\text{Rata – rata Delay RTT} = \frac{\text{Total delay Round Trip Time}}{\text{Total siklus request-response}} \dots\dots\dots (3.2)$$

Kategori *delay* diperlihatkan pada Tabel 3.2 berikut

Tabel 3. 2 Kategori *Delay*

Kategori	<i>Delay (ms)</i>	Indeks
Sangat bagus	< 150	4
Bagus	150 - 300	3
Sedang	300 - 450	2
Jelek	> 450	1

Sumber : (Nisa et al., 2024)

3. Jitter

Jitter adalah nilai yang menunjukan banyaknya variansi atau fluktuasi dari *delay* pada komunikasi data (Nisa et al., 2024; Wulandari, 2016). Jika jumlah total

siklus *request-response* adalah N , maka *Jitter* dapat dihitung menggunakan persamaan 3.3.

$$Jitter = \frac{\sum_{n=2}^N |Delay\ RTT_n - Delay\ RTT_{n-1}|}{N-1} \dots\dots\dots (3.3)$$

Kategori *Jitter* diperlihatkan pada Tabel 3.3.

Tabel 3. 3 Kategori *Jiiter*

Kategori	<i>Jitter</i> (ms)	Indeks
Sangat bagus	0	4
Bagus	0 - 75	3
Sedang	75 - 125	2
Jelek	125 - 255	1

Sumber : (Nisa et al., 2024)

4. *Packet Loss*

Packet Loss merupakan kondisi yang merepresentasikan jumlah total paket yang hilang pada komunikasi data (Nisa et al., 2024; Wulandari, 2016). *Packet loss* merupakan perbandingan antara paket yang diterima dengan paket yang dikirim (Rachman et al., 2023). Hal ini direpresentasikan melalui persamaan berikut.

$$Packet\ Loss = \frac{Paket\ dikirim - Paket\ diterima}{Paket\ dikirim} \times 100\% \dots\dots\dots (3.4)$$

Kategori *Packet Loss* diperlihatkan pada Tabel 3.4 berikut.

Tabel 3. 4 Kategori *Packet Loss*

Kategori	<i>Packet Loss</i> (%)	Indeks
Sangat bagus	< 3	4
Bagus	3-15	3
Sedang	15-25	2
Jelek	> 25	1

Sumber : (Nisa et al., 2024)

Setelah semua parameter di atas diukur, maka nilai rata-rata indeks dari semua parameter dikategorisasikan ke standar persentase nilai QoS berdasarkan standar dari TIPHON. Standar persentase nilai tersebut dimuat dalam Tabel 3.5.

Tabel 3. 5 Standar persentase nilai QoS

Nilai	Persentase	Kategori
3,8 - 4	95 - 100	Sangat Baik
3 – 3,79	75 - 95	Baik
2 – 2,99	50 - 75	Kurang Baik
1 – 1,99	25 - 50	Buruk

Sumber : (Nisa et al., 2024)

3.2 Waktu dan Tempat Penelitian

Penelitian dilakukan di Lingkungan Kampus 2 Universitas Siliwangi yang berlokasi di Mugarsari, Kec. Tamansari, Kota Tasikmalaya. Adapun estimasi waktu penelitian dijelaskan dengan Tabel 3.6.

Tabel 3. 6 Waktu Penelitian

[illegible]