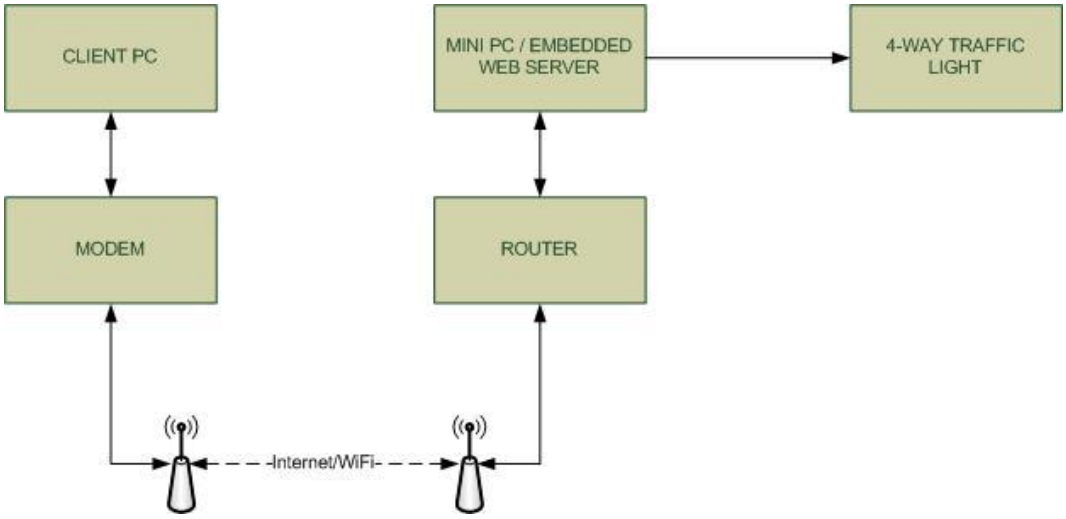
BAB III
PERANCANGAN SISTEM

Pada bab ini akan dibahas tentang perancangan alat yang meliputi diagram blok sistem beserta penjelasan atau cara kerja dari masing-masing blok, spesifikasi perangkat keras, dan algoritma pemrograman, yang digunakan pada penerapan *embedded web server* pada perancangan sistem kendali lampu lalu lintas.

3.1 Diagram Blok Sistem

Diagram blok sistem secara keseluruhan diperlihatkan pada gambar 3.1.



Gambar 3.1 Diagram Blok Sistem

Dari gambar dapat diuraikan penjelasan atau cara kerja tiap-tiap blok sistem yaitu sebagai berikut:

1. *PC client* ialah komputer yang berperan sebagai pengakses situs web pengendali lampu lalu lintas. PC client yang digunakan ialah laptop.
2. *Modem* digunakan untuk menghubungkan komputer ke layanan internet/*WiFi*. Modem yang digunakan ialah Huawei 3G Telkomsel Flash.

3. *Router* digunakan untuk menghubungkan *mini PC* ke layanan internet/*WiFi*. *Router* yang digunakan ialah TP-Link TL-MR3020.
4. *Mini PC/Embedded Web Server* ialah komputer mini yang berperan sebagai *embedded web server*. Selain itu, komputer mini ini berisi data-data dan program yang berkenaan dengan sistem kendali lampu lalu lintas. Program ini dirancang menggunakan algoritma *Welch-Powell. Mini PC* yang digunakan ialah raspberry pi b+.
5. *4-way Traffic Light* ialah sistem lampu lalu lintas yang akan dikendalikan.

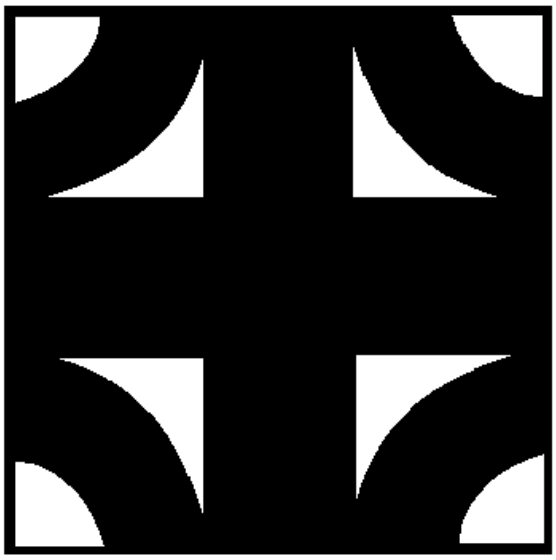
3.2 Perancangan Perangkat Keras (*Hardware*)

3.2.1 Perancangan Mekanik

Pada perancangan mekanik akan digambarkan pemodelan dari persimpangan yang menjadi pembahasan penelitian ini.

a. Bentuk Persimpangan

Bentuk dari persimpangan lampu lalu lintas yang digunakan pada perancangan tugas akhir ini dapat dilihat pada gambar 3.2.



Gambar 3.2 Model Persimpangan (tampak atas)

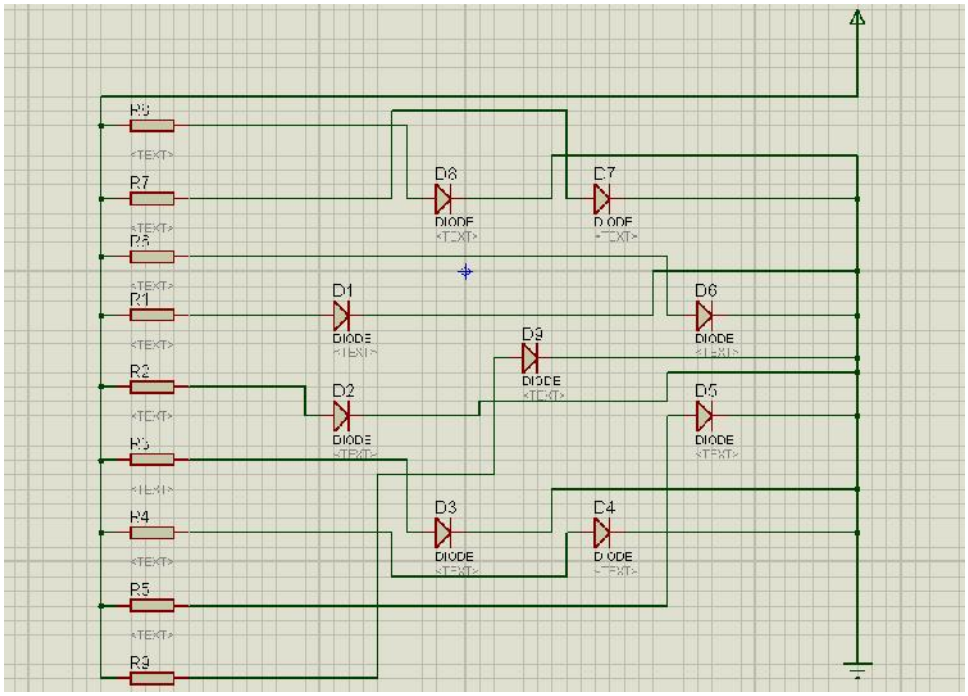
b. Dimensi

Simulasi dari model persimpangan lampu lalu lintas ini menggunakan bahan karton tebal dengan ukuran sebagai berikut:

- Panjang lahan : 30 cm
- Lebar lahan : 30 cm
- Panjang set lampu : 9,5 cm
- Lebar set lampu : 5,5 cm

3.2.2 Perancangan Sistem lampu lalu lintas

Lampu pada lampu lalu lintas akan dirancang menggunakan 9 led. Adapun rangkaian 1 lampu lalu lintas dapat dilihat pada gambar III.3.



Gambar 3.3 Rangkaian 1 lampu

Pada gambar tersebut dapat diperhatikan bahwa 9 led disusun secara paralel. Hal ini bertujuan apabila salah satu led mengalami kerusakan dan padam maka, led yang lainnya tetap menyala.

Berdasarkan *datasheet*, pin GPIO pada raspberry pi b+ mengalirkan arus dengan tegangan sebesar 5 V saat di-*set* sebagai *output*. Tegangan sebesar itu akan merusak led karena tegangan tersebut di atas ambang tegangan maksimal led. Oleh sebab itu, dibutuhkan resistor sebagai

pengaman untuk setiap led. Berikut besar resistor yang akan digunakan pada rangkaian lampu:

$R_{Merah} = \frac{(5-2{,}1)\,V}{20\,mA}$

= 145 ohm

$R_{Kuning} = \frac{(5-2{,}4)\,V}{20\,mA}$

= 130 ohm

$R_{Hijau} = \frac{(5-2{,}6)\,V}{20\,mA}$

= 120 ohm

Dari perhitungan diatas, diperoleh 3 besaran resistor led untuk masing-masing lampu lalu lintas. Namun, sangat sulit menemukan resistor yang tepat sebesar itu di pasaran. Solusi yang paling tepat diambil ialah menggunakan resitor yang besarnya mendekati nilai-nilai tersebut. Di pasaran, resistor terkecil yang mudah ditemukan sebesar 180 ohm. Oleh karena itu, besar resistor yang digunakan pada penelitian ini sama untuk semua warna lampu yaitu 180 ohm.

3.2.3 Sistem Mini PC

Sistem menggunakan *mini PC* Raspberry Pi B+ sebagai *embedded web server*. Adapun spesifikasi dari Raspberry Pi B+ dapat dilihat pada Tabel 3.1.

Tabel 3.1 Spesifikasi Raspberry Pi B+

CPU	700MHz Low Power ARM11
GPU	Dual Core VideoCore IV
Memory	512MB SDRAM
Power	Micro USB socket 5V, 2A
Dimension	85 x 56 17 mm
USB	4 x USB 2.0
GPIO	40 pins (include +3.3V pin, +5V

	pin, GND pin)
Ethernet	10/100 BaseT Socket

Dari tabel diatas dapat dilihat kategori-kategori spesifikasi yang dibutuhkan terutama jika dilihat dari CPU dan GPU yang dimilikinya. Raspberry Pi B+ memperlihatkan kemampuannya yang dapat memproses data dengan cepat. Selain itu, jumlah pin GPIO yang banyak dan RAM-nya yang besar memperlihatkan banyaknya data yang dapat ditanganinya. Raspberry pi b+ memiliki GPIO (*General Purpose Input Output*) yang berfungsi untuk mempengaruhi rangkaian atau sistem elektronika lain di luar sistem *mini PC*. Untuk memprogram pin-pin tersebut, raspberry pi b+ menggunakan bahasa pemrograman python. Bahasa pemrograman ini tergolong bahasa tingkat tinggi yang memiliki sintaks-sintaks sederhana.

3.3 Perancangan Perangkat Lunak (*Software*)

3.3.1 Perancangan Algoritma Pengendali Lampu Lalu Lintas

Pada perancangan algoritma pengendali lampu lalu lintas akan tergambarkan bagaimana pengaruh sistem *mini PC* terhadap rangkaian elektronika sistem kendali lampu lalu lintas. Sebelum merancang suatu algoritma, perlu adanya identifkasi input dan output dari algoritma tersebut.

Tabel 3.2 Siklus Umum Pergantian Warna Lampu Lalu Lintas

Siklus	Jalur 1			Jalur 2			Jalur 3			Jalur 4			Delay
	H	K	M	H	K	M	H	K	M	H	K	M	
1*	1	0	0	0	0	1	0	0	1	0	0	1	N
2	0	1	0	0	0	1	0	0	1	0	0	1	O
3	0	0	1	0	0	1	0	0	1	0	0	1	P
4	0	0	1	1	0	0	0	0	1	0	0	1	Q
5	0	0	1	0	1	0	0	0	1	0	0	1	R
6	0	0	1	0	0	1	0	0	1	0	0	1	S
7	0	0	1	0	0	1	1	0	0	0	0	1	T

8	0	0	1	0	0	1	0	1	0	0	0	1	U
9	0	0	1	0	0	1	0	0	1	0	0	1	V
10*	0	0	1	0	0	1	0	0	1	1	0	0	W
11*	0	0	1	0	0	1	0	0	1	0	1	0	X
12*	0	0	1	0	0	1	0	0	1	0	0	1	Y

Tabel 3.3 Keterangan Tabel 3.2

1	ON
0	OFF
H	Hijau
K	Kuning
M	Merah
N-Y	Variasi delay
(Number)*	Pengecualian

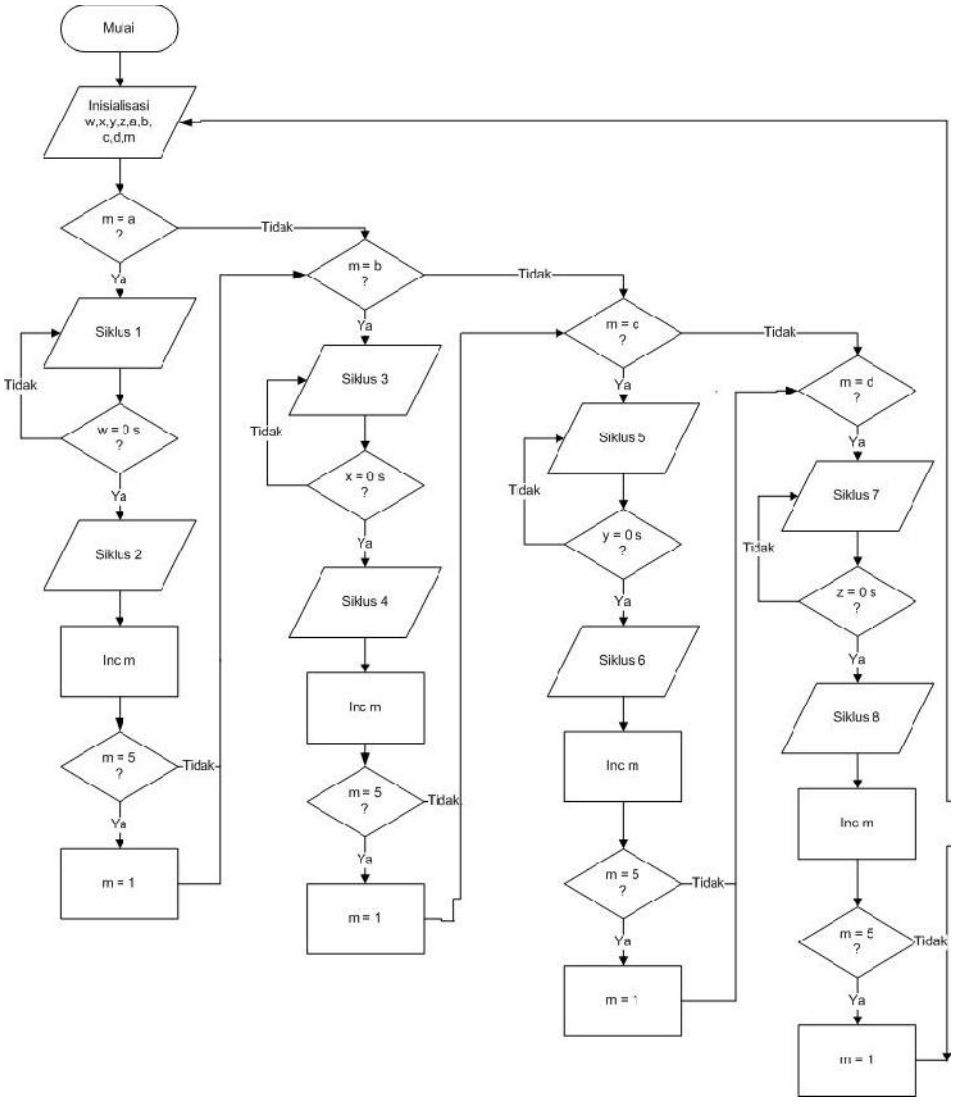
Tabel diatas menunjukkan siklus umum pergantian warna lampu lalu lintas di persimpangan 4 arah. Siklus tersebut masih kurang sesuai dengan urutan penyalaan lampu yang sebenarnya. Perlu diperhatikan bahwa pada siklus ini tidak terdapat urutan ke-2 pada penyalaan lampu sesuai aturan yang digunakan. Oleh sebab itu, siklus ini akan dimodifikasi menggunakan algoritma *Welch-Powell* sehingga diperoleh siklus yang dapat dilihat pada tabel 3.4.

Tabel 3.4 Modifikasi Siklus Umum Pergantian Warna Lampu Lalu Lintas

Siklus	Jalur 1			Jalur 2			Jalur 3			Jalur 4			Delay
	H	K	M	H	K	M	H	K	M	H	K	M	
1	1	0	0	0	0	1	0	0	1	0	0	1	A
2	0	1	0	0	1	1	0	0	1	0	0	1	
3	0	0	1	1	0	0	0	0	1	0	0	1	B
4	0	0	1	0	1	0	0	1	1	0	0	1	

5	0	0	1	0	0	1	1	0	0	0	0	1	C
6	0	0	1	0	0	1	0	1	0	0	1	1	
7	0	0	1	0	0	1	0	0	1	1	0	0	D
8	0	1	1	0	0	1	0	0	1	0	1	0	

Dari tabel diatas diperolehlah siklus yang lebih sedikit daripada siklus sebelumnya. Dengan delay sebagai input dan siklus kondisi lampu sebagai output, didapatkan algoritma pemrograman sistem kendali lampu lalu lintas sebagai berikut:



Gambar 3.4 Diagram Alir Pemrograman Lampu Lalu Lintas

3.3.2 Pemrograman Lampu Lalu Lintas

Diagram alir pemrograman tersebut akan diterapkan dalam bahasa pemrograman python. Dalam pemrograman perlu ditentukan 3 bagian utama yaitu nama, bagian dekalrasi, dan bagian deskripsi.

3.3.2.1 Nama Program

Pemrograman python dalam raspberry pi menggunakan nama program bereksitensi .py sebagai inisial yang dipanggil dari LXterminal agar dapat dieksekusi. LXterminal pada OS raspberry pi berfungsi sama seperti halnya command prompt pada *OS Windows*. Pada pemrograman ini, program akan diberi nama trafficlight.py. Untuk mengeksekusi program ini, maka ketik

sudo python trafficlight.py

pada LXterminal dan enter. Program akan berjalan sesuai dengan kode-kode program di dalamnya.

3.3.2.2 Dekalrasi

Pada diagram alir pemrograman, terdapat huruf-huruf yang berfungsi sebagai variabel. Umumnya pada pemrograman, variabel harus di tentukan memiliki nilai atau satuan besarannya seperti integer, karakter, dan lain sebagainya. Namun pada python khususnya pada raspberry pi, hal itu tidak dibutuhkan terutama pada pemrograman ini. Pada pemrograman ini, hal yang perlu dideklasikan ialah library dan port-port yang akan digunakan. Adapun deklarasi library pada pemrogram ini.

```
import RPi.GPIO as GPIO        #Mengambil library GPIO
import time                    #Mengambil library time
```

Library berfungsi sebagai fungsi yang akan dijalankan terhadap kode-kode program berikutnya. Adapun kode-kode program tersebut salah satunya ialah mendeklarasikan port-port.

```
GPIO.setmode(GPIO.BCM)    #Memilih mode BCM saat pembacaan pin
GPIO.setup(16,GPIO.OUT)   #set pin 16 sebagai output
GPIO.setup(20,GPIO.OUT)   #set pin 20 sebagai output
GPIO.setup(21,GPIO.OUT)   #set pin 21 sebagai output
```

Perlu diketahui bahwa terdapat dua mode pembacaan pin atau port oleh python yaitu *mode* BCM dan *mode* BOARD. Yang dimaksud dengan *mode* BCM ialah pin-pin atau port-port yang dimaksud dalam kode-kode program merupakan pin-pin atau port-port yang penomorannya tidak terlihat pada papan modul raspberry pi. Sedangkan *mode* BOARD, penomoran pin atau port dapat dilihat pada papan modul atau berdasarkan urutan pin atau port yang terlihat.

3.3.2.3 Deskripsi

Bagian deskripsi merupakan bagian utama dalam pemrograman. Bagian ini akan menjalankan sistem lampu lalu lintas yang telah diintegrasikan terhadap modul raspberry pi. Adapun contoh yang diambil dari pemrograman ini ialah sebagai berikut:

```
GPIO.output(16,True)      # lampu merah di jalur A padam
time.sleep(w)             #waktu pembukaan jalur A
GPIO.output(16,False)     #lampu merah di jalur A padam
```