BAB III

PERANCANGAN DAN PEMBUATAN SIMULASI

3.1. Perancangan Simulasi

Untuk membuat simulasi pengontrol lampu lalu lintas pada persimpangan yang berdekatan dibutuhkan suatu simulator. Ada 6 persimpangan yang akan dibuat dengan menggunakan program simulator LABView 6.1 seperti terlihat pada **Gambar 3.1.** Sedangkan **Gambar 3.2** merupakan program simulasi untuk front panel **Gambar 3.1.**

Ada beberapa istilah yang digunakan dalam pengendalian lampu lalu lintas. Untuk sebaran kendaraan antara lain: *short* (pendek), medium, *long* (panjang), dan *very long* (sangat panjang).

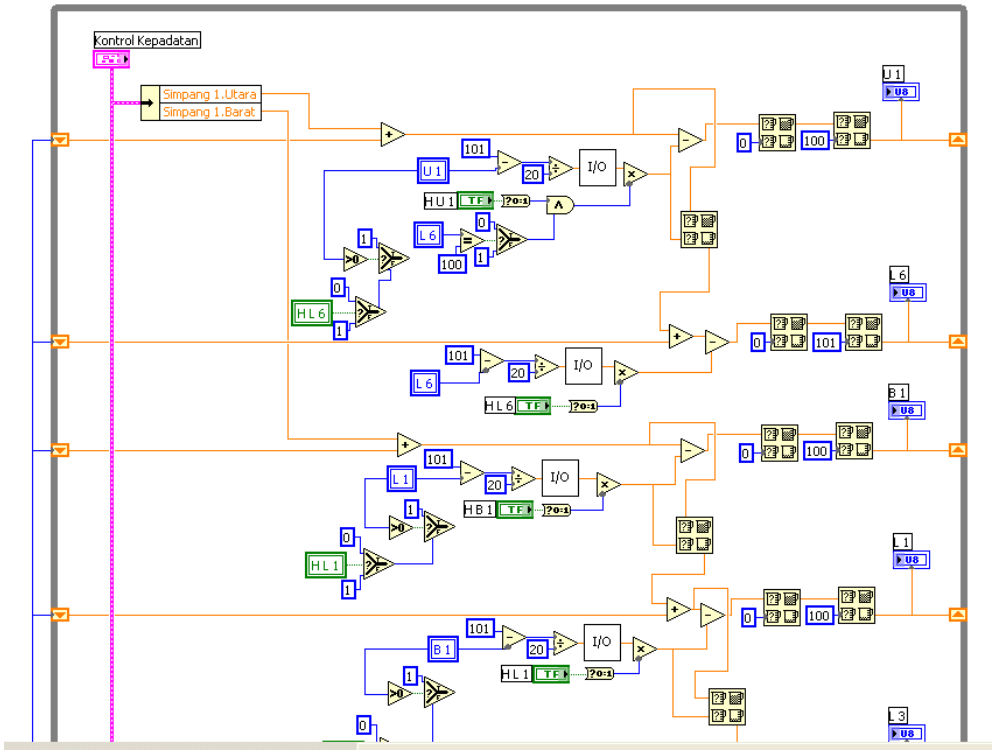
Untuk contoh kita ambil simpangan pertama. Jika U1 ada mobil lalu kontrol kepadatan simpang 1 utara di nol (tidak ada mobil masuk lagi), kemudian HU1 hijau dan HL6 merah, maka jumlah kendaraan pada L6 akan sama dengan jumlah U1 yang tadi. Jika kontrol kepadatan pada simpang 1 utara tidak nol (ada mobil masuk), HU1 hijau, HL6 merah, maka jumlah kendaraan pada L6 akan terus bertambah. Jika HU1 hijau, HL6 merah, dan jumlah kendaraan pada L6 telah mencapai batas maksimal, maka U1 bagai lampu merah (jumlah kendaraan akan terus bertambah seperti saat HL1 merah).

Ket: HU1 : lampu lalu lintas (merah/hijau) pada jalan utara 1

U1 : jumlah kendaraan pada simpang 1 utara

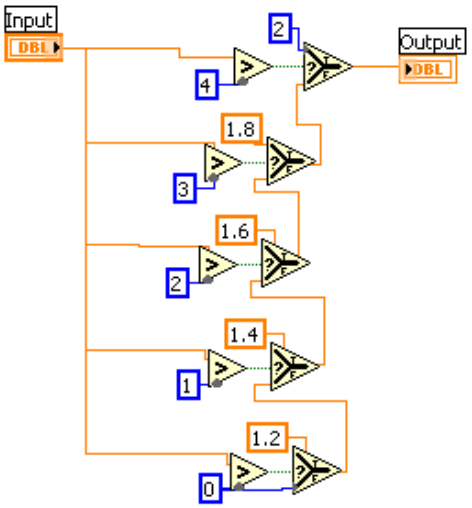
HL6 : lampu lalu lintas (merah/hijau) pada simpang pertama

L6 : jumlah kendaraan pada *link* 6



Gambar 3.2 Program Simulasi Lampu Lalu Lintas

Dalam program simulasi pengontrol lampu lalu lintas terdapat fungsi *input*/*output* (I/O) untuk menentukan keluaran dari setiap kondisi masukan. **Gambar 3.3** merupakan program untuk I/O.



Gambar 3.3 Program I/O

3.2. Perancangan Menggunakan Logika Fuzzy

3.2.1. Next Phase Module

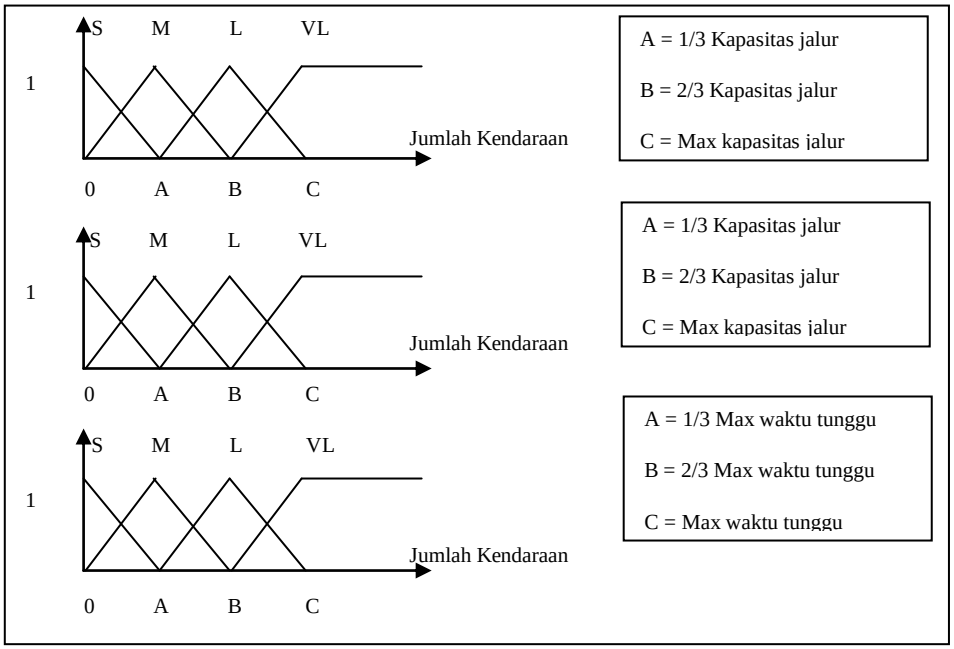
Next Phase Module yaitu fase pada saat lampu merah, memiliki 3 (tiga) *input* dan 2 (dua) *output*. Ketiga *input* tersebut adalah *Queue Num*, *Front Num*, dan *Red Time*. *Queue Num* yaitu jumlah kendaraan pada saat lampu merah menyala. *Front Num* yaitu jumlah kendaraan yang menunggu pada jalan penghubung (*link*). *Red Time* yaitu waktu pada saat lampu merah.

Sedangkan kedua *output*nya yaitu: *Urgency* dan *Phase*. *Urgency* merupakan kondisi lalu lintas yang paling diutamakan seberapa lamakah waktu untuk berpindah ke lampu hijau. *Phase* maerupakan fase yang dipilih untuk fase berikutnya setelah lampu hijau. Nilai dari *Urgency* pada setiap jalur merupakan penjumlahan dari fase tersebut.

Komponen fuzzifikasi variabel masukan dari *Next Phase Module* dapat dilihat pada **Gambar 3.4**. Pada modul ini terdapat 28 aturan dan ada 2 aturan yang harus diperhatikan pada *output Urgency*, yaitu:

a. Jika jumlah kendaraan pada penghubung (*Link*) antar persimpangan mencapai kapasitas maksimum, tidak ada kendaraan yang diijinkan untuk memasuki penghubung tersebut.

b. Jika *Red Time* sangat panjang, kendaraan akan diberikan prioritas untuk melintas. Hal ini ditujukan agar waktu tunggu untuk setiap kendaraan dapat diperkecil.



Gambar 3.4 Fuzzifikasi dari *Queue Num*, *Front Num*, dan *Red Time*

Rule 1 (aturan 1) pada **Table 3.1** menunjukan bahwa bagaimanapun *fuzzy* untuk *Queue Num* di set nol (*Zerro* = Z) dan berapapun *Front Num* dan *Red Time*, keluaran *Urgency* akan tetap nol (Z). jika nilai *Queue Num* dan *Red Time* makin besar, maka *Urgency* pun akan semakin besar secara otomatis. Akan tetapi jika nilai *Qeueu Num* semakin besar, maka nilai *Urgency* akan semakin kecil secara otomatis. Untuk mendapatkan kedua keluaran dari *Next Phase Module* (*Phase* dan *Urgency*), kita harus membandingkan seluruh nilai *Urgency* dari setiap fase kecuali saat fase lampu hijau.

Tabel 3.1 Aturan *Fuzzy* untuk *Next Phase Module*

RULE	INPUT			OUTPUT
	QEUEU NUM	FRONT NUM	RED TIME	URGENCY
1	Z			Z
2	S	S	S	S
3	S	S	M	S
4	S	S	L	M
5	S	M	S	S
6	S	M	M	S
7	S	M	L	M
8	S	L	S	S
9	S	L	M	S
10	S	L	L	S
11	M	S	S	S
12	M	S	M	M
13	M	S	L	L
14	M	M	S	S
15	M	M	M	M
16	M	M	L	L
17	M	L	S	S
18	M	L	M	M
19	M	L	L	M
20	L	S	S	M
21	L	S	M	L
22	L	S	L	VL
23	L	M	S	M
24	L	M	M	L
25	L	M	L	L
26	L	L	S	S
27	L	L	M	M
28	L	L	L	M

Ket : S = *Short* (pendek)

M = *Medium* (menengah)

L = *Long* (panjang)

VL = *Very Long* (sangat panjang)

Z = *Zerro* (nol)

Salah satu contoh cara kerja dari aturan di atas, jika kita lihat aturan ke tujuh. Saat antrian kendaraan pada salah satu persimpangan (*Qeueu Num*) pendek (S), kemudian antrian jumlah kendaraan pada jalan penghubung (*Front Num*) sedang (M) sedangkan waktu lampu merah pada jalan berikutnya (*Red Time*) panjang (L), maka keluaran *Next Phase Module* (*Urgency*) adalah sedang (M).

3.2.2. *Green Phase Module*

Green Phase Module merupakan keadaan atau fase pada saat lampu hijau menyala. Modul ini memiliki dua *input* yaitu *Qeueu Num* dan *Front Num* dan satu *output* yaitu *Extend*. Fuzzifikasi masukan dari modul ini sama dengan *Next Phase Module*, seperti yang ditunjukan **Gambar 3.4**, tetapi keluaran variable *fuzzy* dari modul ini (*Extend*) berisi lima fungsi keanggotaan. *Qeueu Num* merupakan jumlah antrian kendaraan pada saat lampu hijau menyala. *Front Num* merupakan jumlah kendaraan yang menunggu saat lampu hijau pada jalan penghubung. Keluaran *Green Phase Module* yaitu *Extend* merupakan penterjemahan kemungkinan yang terjadi ketika lampu hijau menyala. 10 aturan dari modul ini dapat dilihat pada **Table 3.2**

Tabel 3.2 Aturan Fuzzy untuk *Green Phase Module*

RULE	INPUT		OUTPUT
	QEUEU NUM	FRONT NUM	EXTEND
1	Z		Z
2	S	S	S
3	S	M	S
4	S	L	S
5	M	S	L
6	M	M	M
7	M	L	S
8	L	S	VL
9	L	M	VL
10	L	L	L

Melihat dari aturan diatas, ketika nilai *Queue Num* tinggi, nilai dari keluaran *Extend* juga akan ikut tinggi, dan ketika *Front Num* tinggi, maka *Extend* rendah. Untuk mendapat keluaran dari *Green Phase Module*, kita mengkombinasikan semua nilai *Extend* pada setiap jalur saat lampu hijau menyala.

Contoh cara kerja pada aturan diatas, kita ambil aturan ke lima. Jika jumlah kendaraan pada saat lampu hijau *Qeueu Num* sedang (M), dan kendaraan yang akan memasuki *link* (*Front Num*) pendek (S), maka keluaran dari *Green Phase Module* (*Extend*) panjang (L).

3.2.3. Decision Module

Decision Module membuat keputusan untuk perpindahan ke *Green Phase Module*. Masukan modul ini adalah keruaran dari *Next Phase Module* dan *Green Phase Module* yaitu *Urgency* dan *Extend*. Pada kasus ini kita menggunakan *Yes* untuk mengakhiri sinyal lampu hijau, dan *No* untuk melanjutkan sinyal lampu hijau. Jika *Urgency* lebih besar dari *Extend*, ini

menunjukan bahwa kondisi lalu lintas untuk fase berikutnya sangat besar dibandingkan dengan fase saat lampu hijau menyala, dan keluaran akan berganti ke fase selanjutnya daripada memperpanjang fase lampu hijau. Aturan dari *Decision Module* ini dapat dilihat pada **Tabel 3.3** di bawah ini.

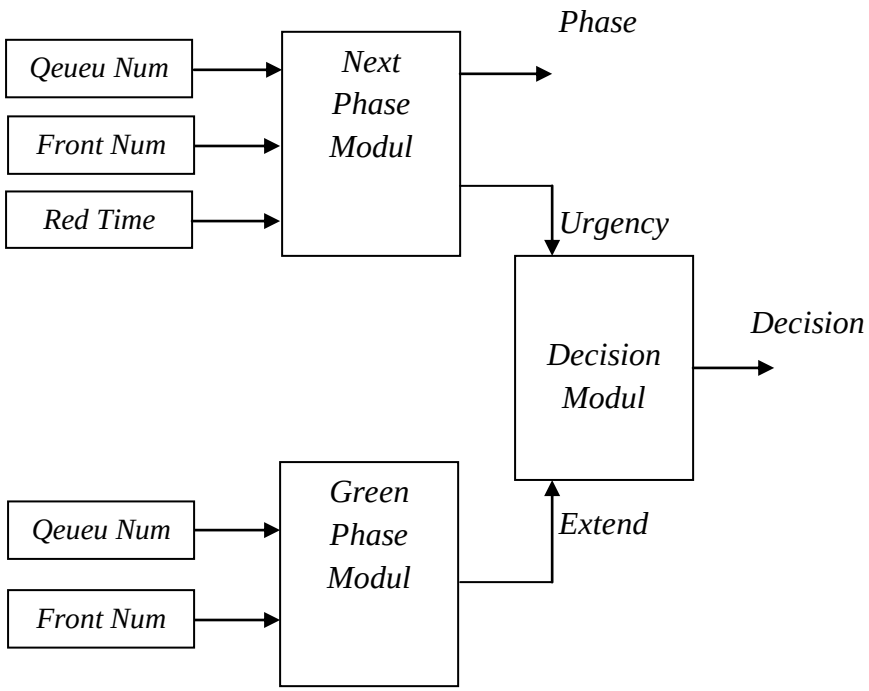
Tabel 3.3 Atruan untuk *Decision Module*

RULE	URGENCY	EXTEND	DECISION
1	Z	Z	NO
2	Z	S	YES
3	Z	M	YES
4	Z	L	YES
5	S	VL	YES
6	S	Z	NO
7	S	S	NO
8	S	M	YES
9	S	L	YES
10	S	VL	YES
11	M	Z	NO
12	M	S	NO
13	M	M	NO
14	M	L	YES
15	M	VL	YES
16	L	Z	NO
17	L	S	NO
18	L	M	NO
19	L	L	NO
20	L	VL	YES
21	VL	Z	NO
22	VL	S	NO
23	VL	M	NO
24	VL	L	NO
25	VL	VL	NO

Cara kerja dari aturan diatas, kita ambil aturan ke 20, jika *Extend* panjang (L), *Urgency* sangat panjang (VL), maka perintah yang keluar pada keluaran *Decision Module* adalah melanjutkan sinyal lampu hijau (*NO*).

3.2.4. Penggabungan Ketiga Modul

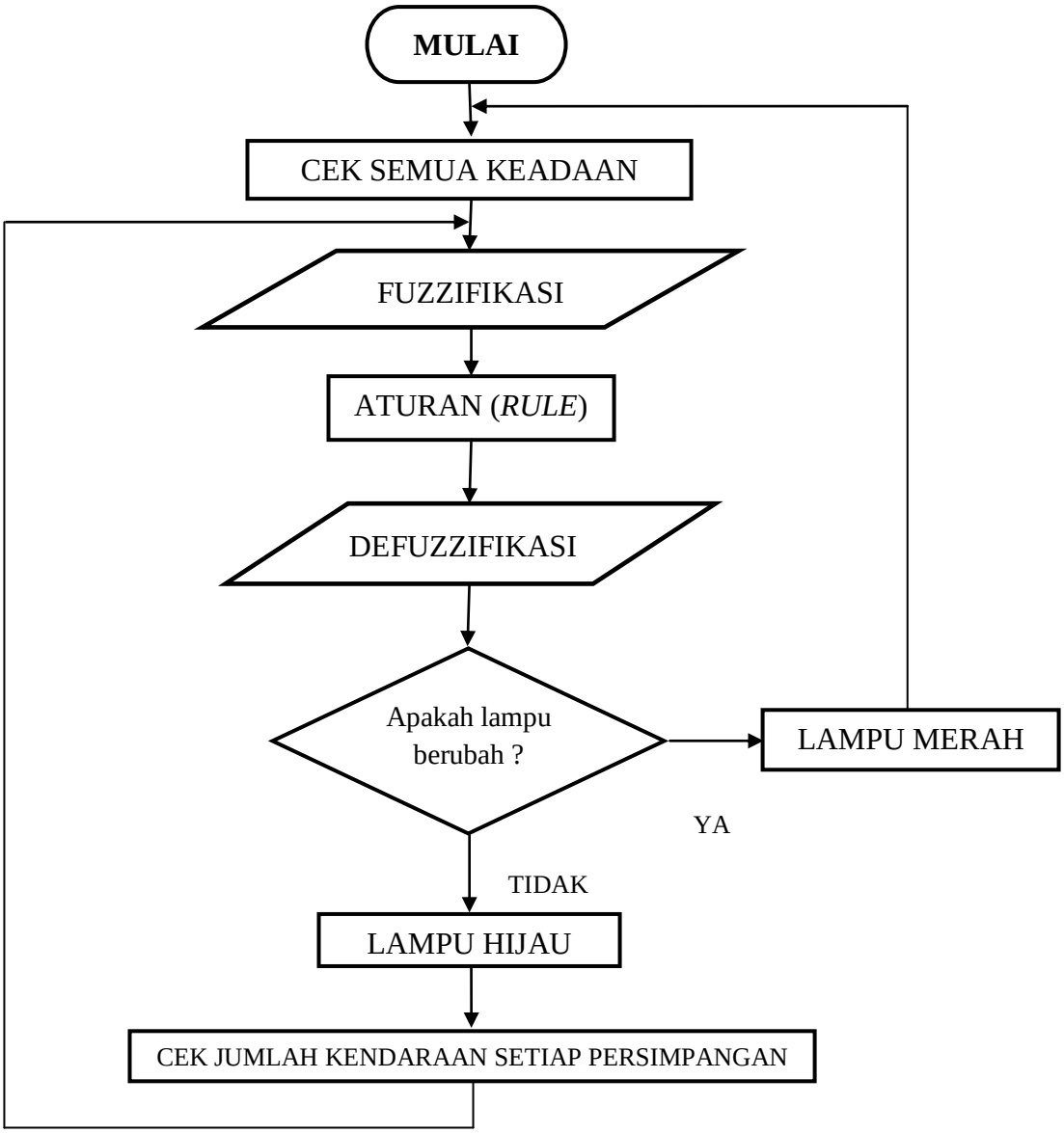
Perancangan logika fuzzy untuk pengaturan lampu lalu lintas pada multi persimpangan yang berdekatan dengan cara menggabungkan ketiga modul (*Grenn Phase Module*, *Next Phase Module*, dan *Decision Module*). **Gambar 3.5** merupakan mekanisme penggabungan dari ketiga modul.



Gambar 3.5 Ilustrasi penggabungan ketiga modul

3.3. Tahapan Simulasi

Tahapan simulasi dari pengontrol lampu lalu lintas pada persimpangan yang berdekatan ini menggunakan LABView 6.1. *Flowchart* dari simulasi ini dapat dilihat pada **Gambar 3.6** dibawah ini.



Gambar 3.6 *Flowchart* Simulasi Lampu Lali Lintas menggunakan *Fuzzy*

Pada saat program mulai dijalankan, ia akan mengecek semua keadaan, baik itu jumlah kendaraan, nyala lampu, dan waktu saat lampu merah. Setelah itu dilakukan proses fuzzifikasi untuk *Next Phase Module*, *Green Phase Module*, dan *Decision Module*. Keluaran dari masing-masing modul akan dicocokan dengan aturan (*rule*) yang ada. Kemudian

dilakukan proses defuzzifikasi berupa nyala lampu (merah atau hijau). Jika lampu merah, maka program akan kembali mengecek semua keadaan. Jika lampu berubah hijau, maka program akan mengecek jumlah kendaraan pada setiap persimpangan untuk menentukan jalan manakah yang akan mendapat priroritas utama untuk lampu hijau.

Tahapan dari simulasi pengontrol lampu lalu lintas dengan menggunakan logika fuzzy adalah.

1. Fuzzifikasi *Next Phase Module*
2. Fuzzifikasi *Green Phase Module*
3. Penentuan keputusan

3.3.1. Fuzzifikasi Next Phase Module

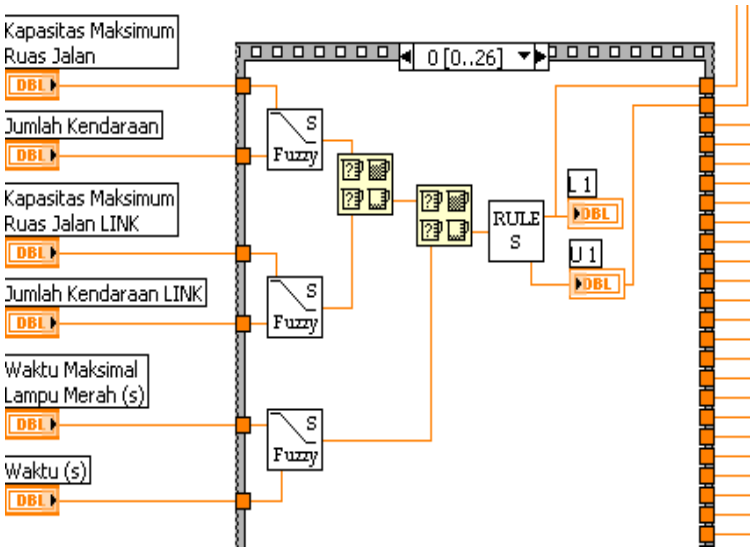
Pada tahapan ini ada beberapa komponen yang di fuzzifikasikan, yaitu:

a. fuzzifikasi *qeueunum* yaitu fuzzifikasi jumlah kendaraan yang menunggu pada saat lampu merah di satu jalur;

b. fuzzifikasi *frontnum* yaitu fuzzifikasi jumlah kendaraan yang berada pada saat lampu hijau;

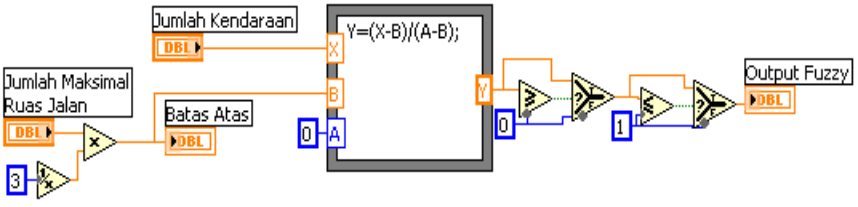
c. fuzzifikasi *redtime* yaitu waktu tunggu pada saat lampu merah menyala;

d. fuzzifikasi *urgency* yaitu fuzzifikasi untuk berubah ke lampu hijau.

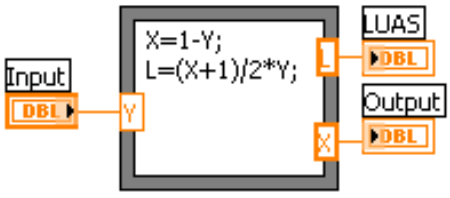


Gambar 3.7 Salah satu program *Next Phase Module*

Pada **Gambar 3.7** (salah satu program *Next Phase Module*) terdapat fungsi S dan aturan (RULE S). Untuk fungsi S digunakan fungsi keanggotaan linier. Persamaan yang digunakan dapat dilihat pada **Gambar 3.8**. sedangkan **Gambar 3.9** merupakan persamaan yang digunakan untuk *Rule S*.



Gambar 3.8 Persamaan untuk fungsi S (*Short*)



Gambar 3.9 Persamaan untuk *Rule S*

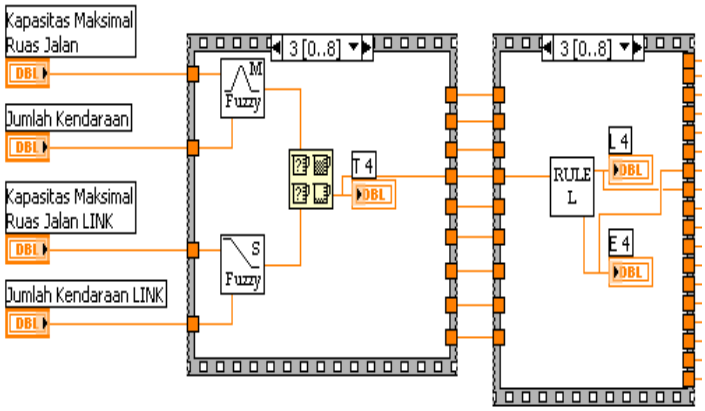
3.3.2. Fuzzifikasi Green Phase Module

Pada tahapan fuzzifikasi *Green Phase Module*, ada beberapa komponen yang difuzzifikasi, yaitu :

a. fuzzifikasi *qeueunum* yaitu fuzzifikasi jumlah kendaraan yang ada pada saat lampu hijau menyala;

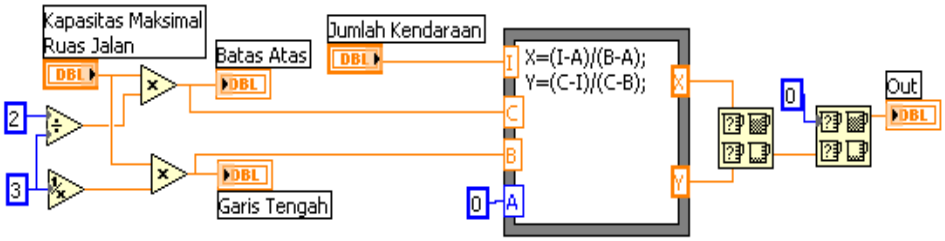
b. fuzzifikasi *frontnum* yaitu fuzzifikasi jumlah kendaraan yang menunggu pada saat lampu hijau;

c. fuzzifikasi *extend* yaitu fuzzifikasi seberapa lama lagi untuk lampu hijau.

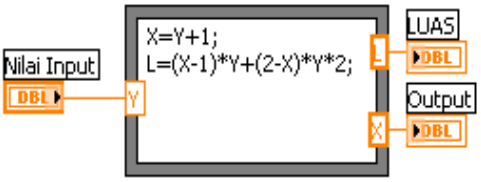


Gambar 3.10 Salah satu program fuzzyfikasi *Green Phase Module*

Pada **Gambar 3.10** (salah satu program *Green Phase Module*) terdapat fungsi M dan aturan (RULE S). Untuk fungsi M digunakan fungsi keanggotaan segitiga. Persamaan yang digunakan dapat dilihat pada **Gambar 3.11**. sedangkan **Gambar 3.12** merupakan persamaan yang digunakan untuk *Rule S*.



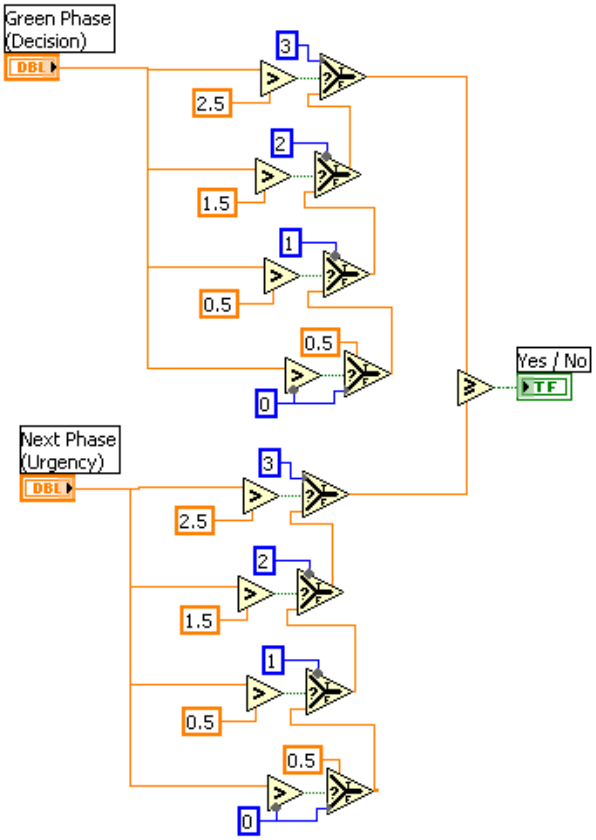
Gambar 3.11 Persamaan untuk fungsi M (*Medium*)



Gambar 3.12 Persamaan untuk *Rule L*

3.3.3. Penentuan Keputusan

Pada tahapan ini, penentuan keputusan menggunakan *Decision Module*. Masukan dari modul ini merupakan nilai keluaran dari *Next Phase Module* yaitu *Urgency* atau kondisi yang lebih diutamakan untuk segera lampu hijau dan nilai keluaran dari *Green Phae* Module yaitu *Extend* atau kondisi untuk mempertahankan lampu hijau. Salah satu program untuk *Decision Module* dapat dilihat pada **Gambar 3.13** dibawah ini.



Gambar 3.13 Program untuk *Decision Module*