Bab 5
Pembangunan dan Validasi Model

5.1. Pembangunan Model

5.1.1. Elemen-elemen Dasar

5.1.1.1. *Locations*

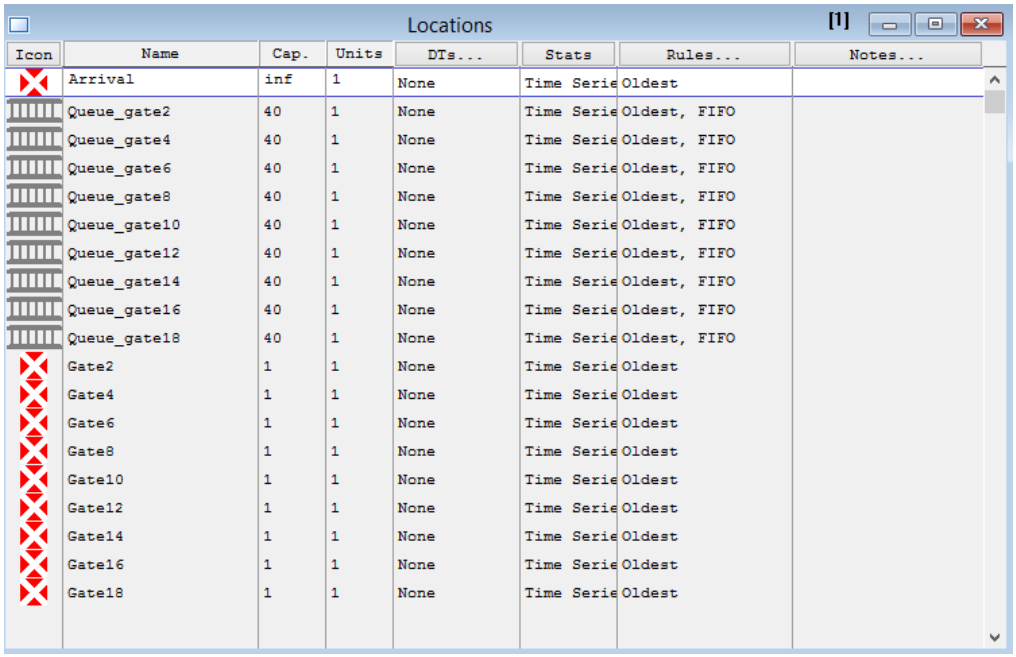
Locations pada ProModel ditetapkan berdasarkan data sturktural sistem. Lokasi pada data stuktural sistem meliputi tempat kedatangan, tempat antrian, dan gardu pembayaran tol. Data lokasi tersebut kemudian ditranslasikan menjadi input pada model yang dibuat. Adapun data lokasi yang akan diinputkan ditunjukkan oleh tabel 5.1.

Tabel 5.1. Translasi data lokasi ke *locations* ProModel

Data lokasi	Nama elemen	Kapasitas	Jumlah unit	Jenis lokasi	Aturan *routing*
Tempat kedatangan	Arrival	*Infinite*	1	*Entity Spot*	*Oldest*
Tempat antrian 1	Queue_gate2	40	1	*Conveyor*	*First in first out*
Tempat antrian 2	Queue_gate4	40	1	*Conveyor*	*First in first out*
Tempat antrian 3	Queue_gate6	40	1	*Conveyor*	*First in first out*
Tempat antrian 4	Queue_gate8	40	1	*Conveyor*	*First in first out*
Tempat antrian 5	Queue_gate10	40	1	*Conveyor*	*First in first out*
Tempat antrian 6	Queue_gate12	40	1	*Conveyor*	*First in first out*
Tempat antrian 7	Queue_gate14	40	1	*Conveyor*	*First in first out*
Tempat antrian 8	Queue_gate16	40	1	*Conveyor*	*First in first out*
Tempat antrian 9	Queue_gate18	40	1	*Conveyor*	*First in first out*
Gardu 1	Gate2	1	1	*Entity Spot*	*Oldest*
Gardu 2	Gate4	1	1	*Entity Spot*	*Oldest*
Gardu 3	Gate6	1	1	*Entity Spot*	*Oldest*
Gardu 4	Gate8	1	1	*Entity Spot*	*Oldest*
Gardu 5	Gate10	1	1	*Entity Spot*	*Oldest*
Gardu 6	Gate12	1	1	*Entity Spot*	*Oldest*
Gardu 7	Gate14	1	1	*Entity Spot*	*Oldest*
Gardu 8	Gate16	1	1	*Entity Spot*	*Oldest*
Gardu 9	Gate18	1	1	*Entity Spot*	*Oldest*

(Sumber: Hasil pengolahan data)

Pada tabel 5.1, tempat kedatangan ditranslasikan menjadi Arrival, tempat antrian menjadi Queue_gateN dan gardu menjadi GateN dengan N adalah nomor identifikasi gardu. Jenis lokasi pada model adalah *entity spot* untuk tempat kedatangan dan *conveyor* untuk tempat antrian dengan aturan *oldest* untuk *entity spot* dan FIFO (*first in first out*) untuk *conveyor*. Untuk *conveyor* panjangnya adalah 200 meter dengan kecepatan 500 mpm. Pada model ditambahkan satu gardu tambahan menjadi sembilan gardu untuk keperluan eksperimentasi model. Data tersebut kemudian diinputkan pada ProModel seperti yang ditunjukan pada gambar 5.1.



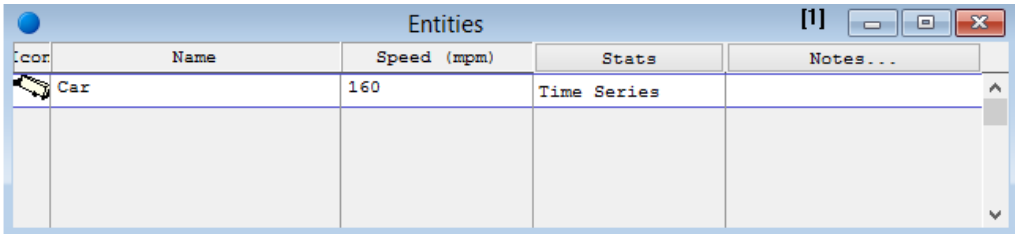
Icon	Name	Cap.	Units	DTs...	Stats	Rules...	Notes...
	Arrival	inf	1	None	Time Serie	Oldest	
	Queue_gate2	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate4	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate6	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate8	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate10	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate12	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate14	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate16	40	1	None	Time Serie	Oldest, FIFO	
	Queue_gate18	40	1	None	Time Serie	Oldest, FIFO	
	Gate2	1	1	None	Time Serie	Oldest	
	Gate4	1	1	None	Time Serie	Oldest	
	Gate6	1	1	None	Time Serie	Oldest	
	Gate8	1	1	None	Time Serie	Oldest	
	Gate10	1	1	None	Time Serie	Oldest	
	Gate12	1	1	None	Time Serie	Oldest	
	Gate14	1	1	None	Time Serie	Oldest	
	Gate16	1	1	None	Time Serie	Oldest	
	Gate18	1	1	None	Time Serie	Oldest	

Gambar 5.1. *Model Locations*

(Sumber: Hasil pengolahan data)

5.1.1.2. *Entities*

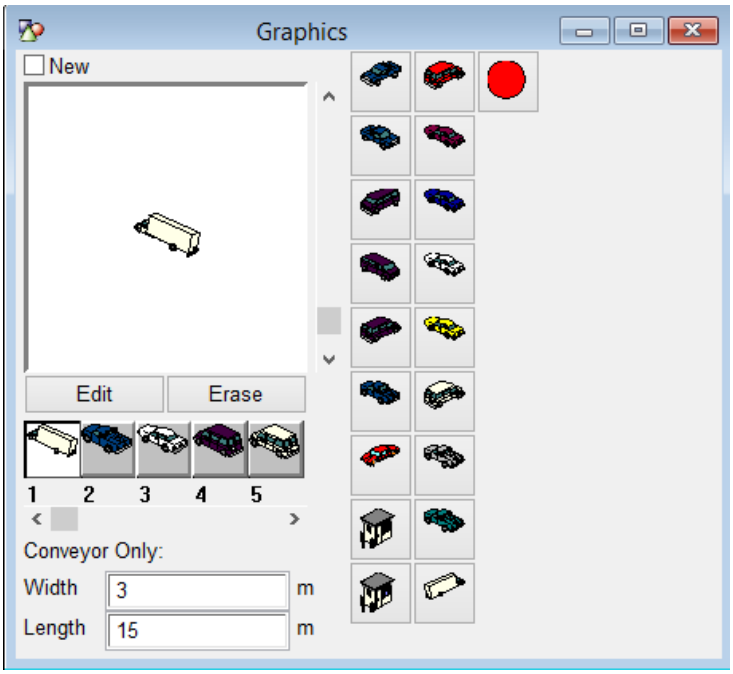
Berdasarkan bab sebelumnya entitas yang ditetapkan untuk model simulasi adalah mobil dengan klasifikasi mobil besar dan mobil kecil. Pada model simulasi ProModel, entitas mobil diwakili oleh satu *Entities* Car dengan klasifikasi ditetapkan sebagai *Attribute*. Penetapan entitas pada ProModel ditunjukan pada gambar 5.2.



Gambar 5.2. *Model Entities*

(Sumber: Hasil pengolahan data)

Pada gambar 5.1 kolom *name* menunjukan nama entitas dalam hal ini yaitu Car, kolom *speed* menunjukan kecepatan entitas pada saat berpindah dari satu lokasi ke lokasi lain, dibiarkan 160 meter per menit karena kecepatan mobil pada antrian ditetapkan pada kecepatan lintasan *conveyor* di *locations*, dan kolom *stats* menunjukan ukuran statistik yang dihitung oleh ProModel, dalam hal ini statistik dihitung secara *Time Serie*s.



Gambar 5.3. *Entities graphics*

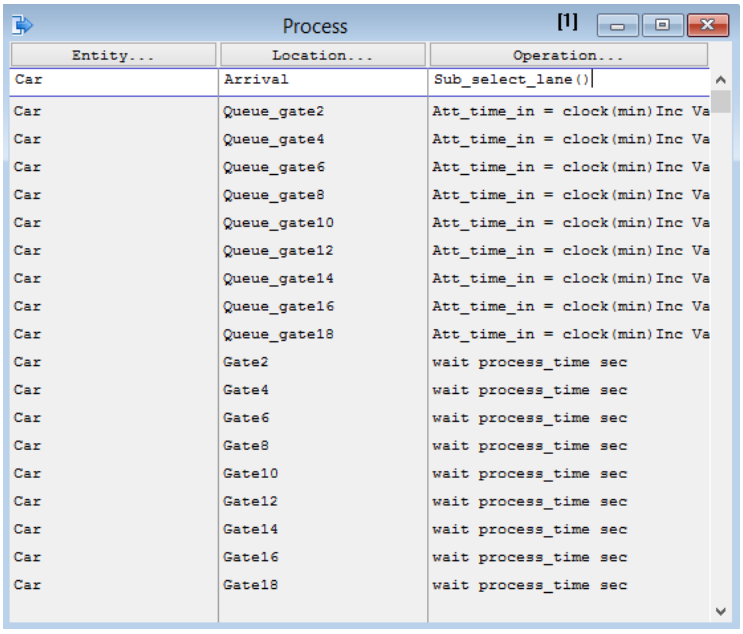
(Sumber: Hasil pengolahan data)

Entitas Car pada model memiliki delapan *graphic/icons* yang menunjukan variasi jenis kendaraan. Setiap *graphic/icons* tersebut memiliki satu ID yang digunakan

untuk identifikasi jika digunakan pada logika pemrosesan model. ID tersebut juga digunakan untuk menggambarkan klasifikasi jenis mobil. ID 1 mewakili klasifikasi mobil besar sedangkan ID 2 hingga 7 mewakili jenis mobil kecil. Selain itu ditetapkan juga ukuran entitas jika entitas melewati lintasan *conveyor*. Untuk mobil besar (ID 1) ukurannya dianggap sama yaitu *width* (lebar) tiga meter dan *length* (panjang) 15 meter. Untuk mobil kecil *width* (lebar) tiga meter dan *length* (panjang) 6 meter. Gambar 5.3 menunjukan *graphics/icons* dan ukuran entitas pada ProModel.

5.1.1.3. *Processing*

Processing menunjukan bagaimana alur proses entitas dari awal hingga akhir proses simulasi. Elemen *processing* ditetapkan berdasarkan data EFD dan deskripsi proses. Elemen *processing* pada ProModel meliputi *Process* dan *Routing* pada setiap *Process* yang dialami entitas selama berada di dalam sistem yang dimodelkan. Elemen *Process* yang terlibat di dalam model ditunjukkan pada gambar 5.4



Gambar 5.4. Model *Processing*

(Sumber: Hasil pengolahan data)

Setiap *Process* yang terlibat menjelaskan aktivitas operasi yang dilakukan entitas pada lokasi yang ditunjukan, selain itu setiap *Process* juga memiliki *Routing* yang menunjukan lokasi tujuan entitas setelah aktivitas dilakukan dan logika pada saat terjadi perpindahan lokasi. Berikut adalah rincian *Process* pada gambar 5.4 beserta *Routing* untuk masing-masing aktivitas yang terlibat:

1. Entitas Car berada di lokasi Arrival

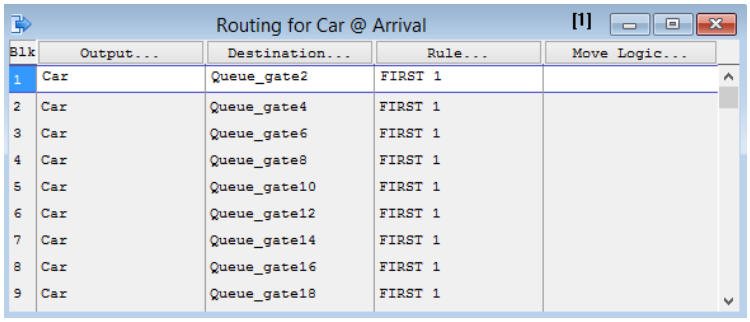
   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.2.

Tabel 5.2. Tabel *Process* dan *Routing* Proses ke-1

Entitas	Car
Lokasi	Arrival
Operasi	Deskripsi: Jalankan prosedur pemilihan lajur, dalam model menggunakan *subroutine*. *Operation Logic:* `Sub_select_lane()`
Output Entitas	Car
Lokasi Tujuan	1. Queue_gate2 2. Queue_gate4 3. Queue_gate6 4. Queue_gate8 5. Queue_gate10 6. Queue_gate12 7. Queue_gate14 8. Queue_gate16 9. Queue_gate18
Aturan Perpindahan	1. Queue_gate2 : First 1 2. Queue_gate4 : First 1 3. Queue_gate6 : First 1 4. Queue_gate8 : First 1 5. Queue_gate10 : First 1 6. Queue_gate12 : First 1 7. Queue_gate14 : First 1 8. Queue_gate16 : First 1 9. Queue_gate18 : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.2 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.2 pada *Routing* ditunjukan pada gambar 5.5



Blk	Output...	Destination...	Rule...	Move Logic...
1	Car	Queue_gate2	FIRST 1	
2	Car	Queue_gate4	FIRST 1	
3	Car	Queue_gate6	FIRST 1	
4	Car	Queue_gate8	FIRST 1	
5	Car	Queue_gate10	FIRST 1	
6	Car	Queue_gate12	FIRST 1	
7	Car	Queue_gate14	FIRST 1	
8	Car	Queue_gate16	FIRST 1	
9	Car	Queue_gate18	FIRST 1	

Gambar 5.5. Model *Routing* Proses ke-1

(Sumber: Hasil pengolahan data)

2. Entitas Car berada di lokasi Queue_gate2

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.3.

Tabel 5.3. Tabel *Process* dan *Routing* Proses ke-2

Entitas	Car
Lokasi	Queue_gate2

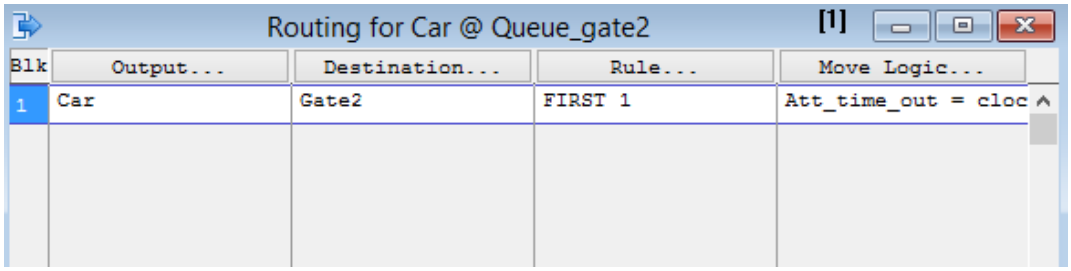
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 2 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate2, 1`
	`Var_qty_queue_gate2 = contents(Queue_gate2)`

Output Entitas	Car
Lokasi Tujuan	Gate2
Aturan Perpindahan	Gate2 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 2
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate2, Att_time_in_line`
	`Var_avgtime_gate2 = Var_totaltime_gate2 /`
	`Var_car_processed_gate2`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.3 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.3 pada *Routing* ditunjukan pada gambar 5.6



Gambar 5.6. Model *Routing* Proses ke-2

(Sumber: Hasil pengolahan data)

3. Entitas Car berada di lokasi Queue_gate4

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.4.

Tabel 5.4. Tabel *Process* dan *Routing* Proses ke-3

Entitas	Car
Lokasi	Queue_gate4

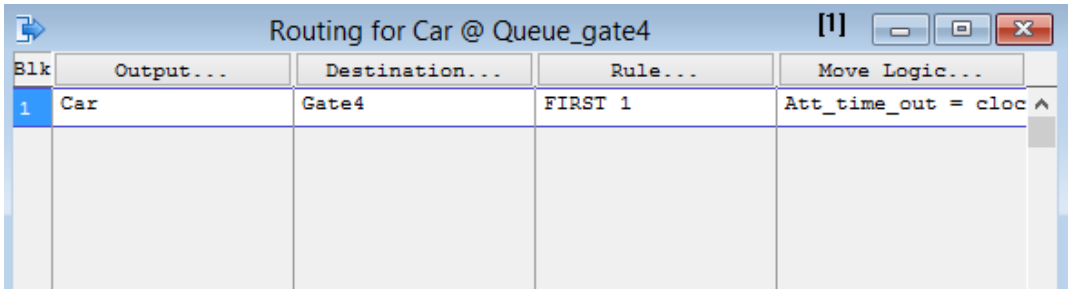
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 4 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate4, 1`
	`Var_qty_queue_gate4 = contents(Queue_gate4)`

Output Entitas	Car
Lokasi Tujuan	Gate4
Aturan Perpindahan	Gate4 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 4
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate4, Att_time_in_line`
	`Var_avgtime_gate4 = Var_totaltime_gate4 /`
	`Var_car_processed_gate4`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.4 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.4 pada *Routing* ditunjukan pada gambar 5.7



Gambar 5.7. Model *Routing* Proses ke-3

(Sumber: Hasil pengolahan data)

4. Entitas Car berada di lokasi Queue_gate6

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.5.

Tabel 5.5. Tabel *Process* dan *Routing* Proses ke-4

Entitas	Car
Lokasi	Queue_gate6

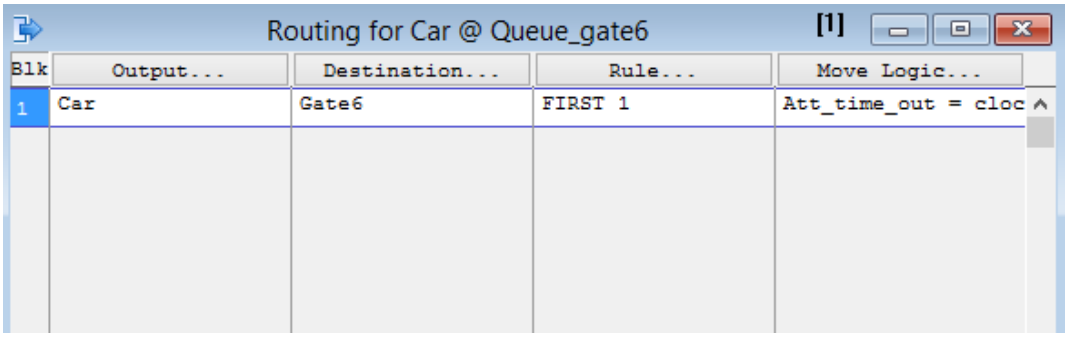
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 6 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate6, 1`
	`Var_qty_queue_gate6 = contents(Queue_gate6)`

Output Entitas	Car
Lokasi Tujuan	Gate6
Aturan Perpindahan	Gate6 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 6
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate6, Att_time_in_line`
	`Var_avgtime_gate6 = Var_totaltime_gate6 /`
	`Var_car_processed_gate6`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.5 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.5 pada *Routing* ditunjukan pada gambar 5.8.



Gambar 5.7. Model *Routing* Proses ke-4

(Sumber: Hasil pengolahan data)

4. Entitas Car berada di lokasi Queue_gate8

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.6.

Tabel 5.6. Tabel *Process* dan *Routing* Proses ke-4

Entitas	Car
Lokasi	Queue_gate8

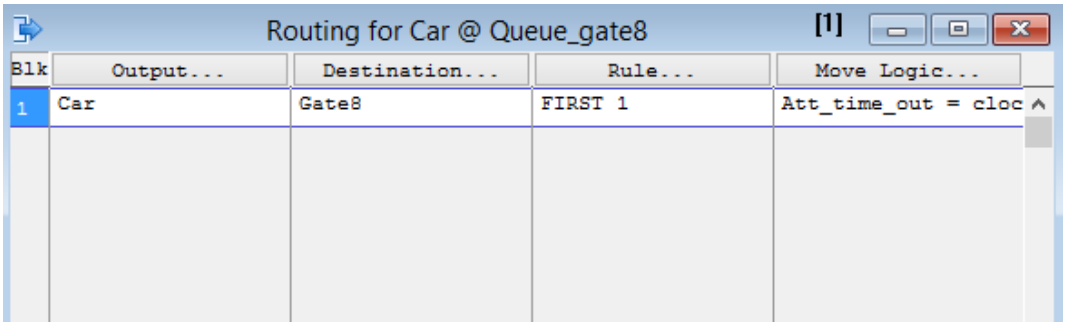
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 8 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate8, 1`
	`Var_qty_queue_gate8 = contents(Queue_gate8)`

Output Entitas	Car
Lokasi Tujuan	Gate8
Aturan Perpindahan	Gate8 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 8
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate8, Att_time_in_line`
	`Var_avgtime_gate8 = Var_totaltime_gate8 /`
	`Var_car_processed_gate8`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.6 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.6 pada *Routing* ditunjukan pada gambar 5.8.



Gambar 5.8. Model *Routing* Proses ke-4

(Sumber: Hasil pengolahan data)

5. Entitas Car berada di lokasi Queue_gate10

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.7.

Tabel 5.7. Tabel *Process* dan *Routing* Proses ke-5

Entitas	Car
Lokasi	Queue_gate10

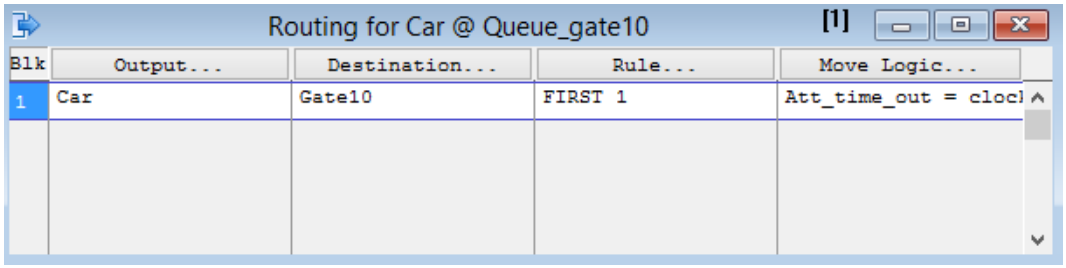
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 10 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate10, 1`
	`Var_qty_queue_gate10 = contents(Queue_gate10)`

Output Entitas	Car
Lokasi Tujuan	Gate10
Aturan Perpindahan	Gate10 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 10
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate10, Att_time_in_line`
	`Var_avgtime_gate10 = Var_totaltime_gate10 /`
	`Var_car_processed_gate10`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.7 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.7 pada *Routing* ditunjukan pada gambar 5.9.



Gambar 5.9. Model *Routing* Proses ke-5

(Sumber: Hasil pengolahan data)

6. Entitas Car berada di lokasi Queue_gate12

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.8.

Tabel 5.8. Tabel *Process* dan *Routing* Proses ke-6

Entitas	Car
Lokasi	Queue_gate12

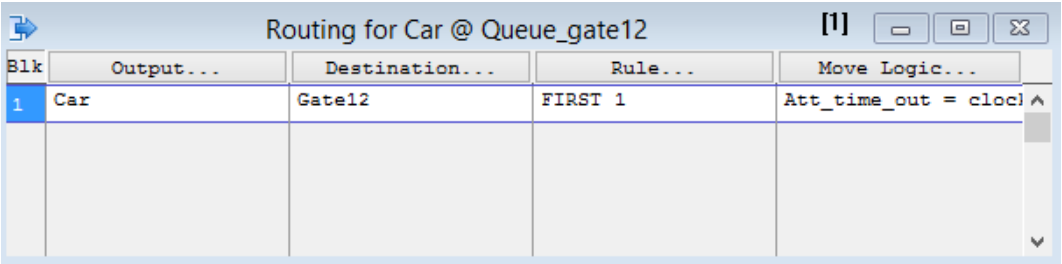
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 12 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate12, 1`
	`Var_qty_queue_gate12 = contents(Queue_gate12)`

Output Entitas	Car
Lokasi Tujuan	Gate12
Aturan Perpindahan	Gate12 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 12
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate12, Att_time_in_line`
	`Var_avgtime_gate12 = Var_totaltime_gate12 /`
	`Var_car_processed_gate12`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.8 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.8 pada *Routing* ditunjukan pada gambar 5.10.



Gambar 5.10. Model *Routing* Proses ke-6

(Sumber: Hasil pengolahan data)

7. Entitas Car berada di lokasi Queue_gate14

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.9.

Tabel 5.9. Tabel *Process* dan *Routing* Proses ke-7

Entitas	Car
Lokasi	Queue_gate14

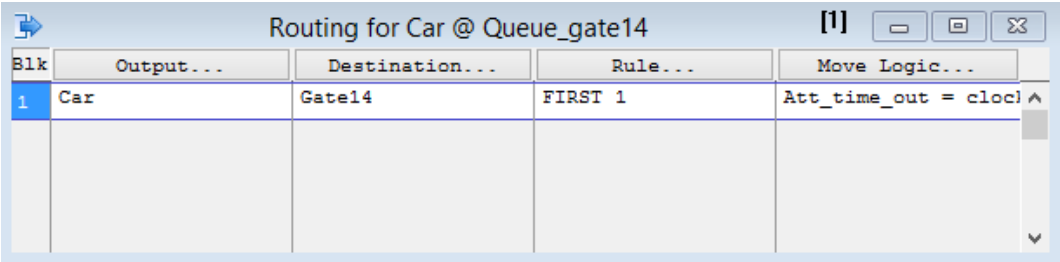
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 14 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate14, 1`
	`Var_qty_queue_gate14 = contents(Queue_gate14)`

Output Entitas	Car
Lokasi Tujuan	Gate14
Aturan Perpindahan	Gate14 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 12
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate14, Att_time_in_line`
	`Var_avgtime_gate14 = Var_totaltime_gate12 /`
	`Var_car_processed_gate14`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.9 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.9 pada *Routing* ditunjukan pada gambar 5.11.



Gambar 5.11. Model *Routing* Proses ke-7

(Sumber: Hasil pengolahan data)

8. Entitas Car berada di lokasi Queue_gate16

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.10.

Tabel 5.10. Tabel *Process* dan *Routing* Proses ke-8

Entitas	Car
Lokasi	Queue_gate16

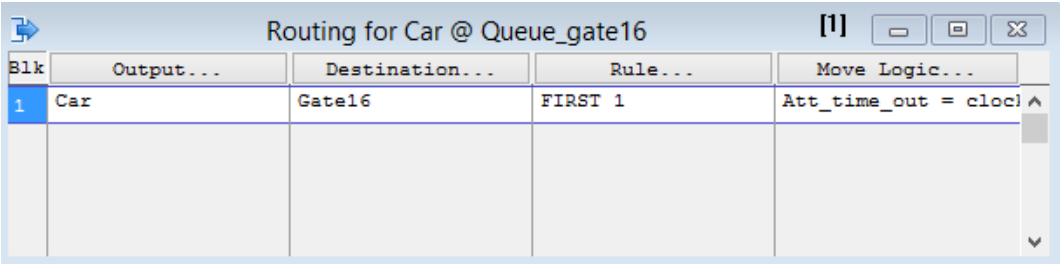
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 16 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate16, 1`
	`Var_qty_queue_gate16 = contents(Queue_gate16)`

Output Entitas	Car
Lokasi Tujuan	Gate16
Aturan Perpindahan	Gate16 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 16
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate16, Att_time_in_line`
	`Var_avgtime_gate16 = Var_totaltime_gate16 /`
	`Var_car_processed_gate16`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.10 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.10 pada *Routing* ditunjukan pada gambar 5.12.



Gambar 5.12. Model *Routing* Proses ke-8

(Sumber: Hasil pengolahan data)

9. Entitas Car berada di lokasi Queue_gate18

   Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.11.

Tabel 5.11. Tabel *Process* dan *Routing* Proses ke-9

Entitas	Car
Lokasi	Queue_gate18

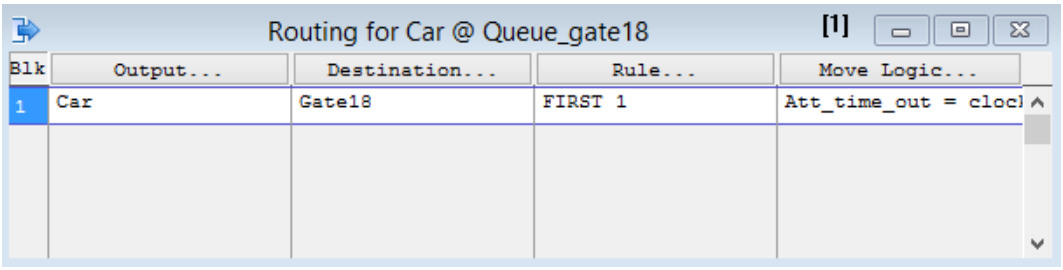
Operasi	Deskripsi:
	1. Catat waktu ketika entitas memasuki lokasi
	2. Tambah data jumlah entitas yang memasuki antrian gardu 18 sebanyak satu unit
	3. Hitung jumlah entitas di dalam lokasi
	
	Operation Logic:
	`Att_time_in = clock(min)`
	`Inc Var_car_processed_gate18, 1`
	`Var_qty_queue_gate18 = contents(Queue_gate18)`

Output Entitas	Car
Lokasi Tujuan	Gate18
Aturan Perpindahan	Gate18 : First 1

Logika Perpindahan	Deskripsi:
	1. Catat waktu ketika entitas keluar dari lokasi
	2. Hitung lama entitas ketika berada di dalam lokasi antrian
	3. Hitung waktu mengantri rata-rata pada gardu 18
	
	Move Logic:
	`Att_time_out = clock(min)`
	`Att_time_in_line = Att_time_out - Att_time_in`
	`inc Var_totaltime_gate18, Att_time_in_line`
	`Var_avgtime_gate18 = Var_totaltime_gate18 /`
	`Var_car_processed_gate18`

(Sumber: Hasil pengolahan data)

Data pada tabel 5.11 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.11 pada *Routing* ditunjukan pada gambar 5.13.



Gambar 5.13. Model *Routing* Proses ke-9

(Sumber: Hasil pengolahan data)

10. Entitas Car berada di lokasi Gate2

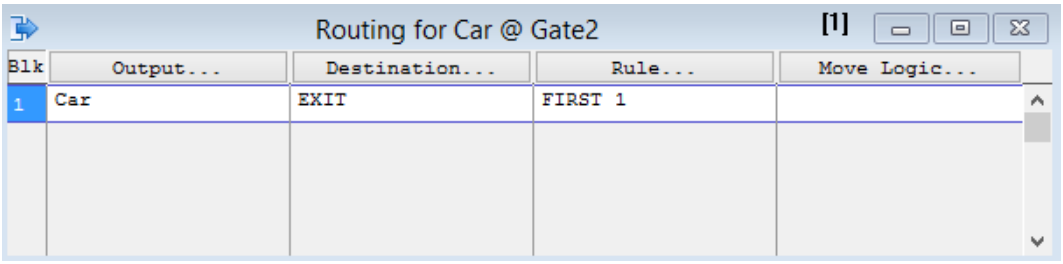
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.12.

Tabel 5.12. Tabel *Process* dan *Routing* Proses ke-10

Entitas	Car
Lokasi	Gate2
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.12 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.12 pada *Routing* ditunjukan pada gambar 5.14.



Gambar 5.14. Model *Routing* Proses ke-10

(Sumber: Hasil pengolahan data)

11. Entitas Car berada di lokasi Gate4

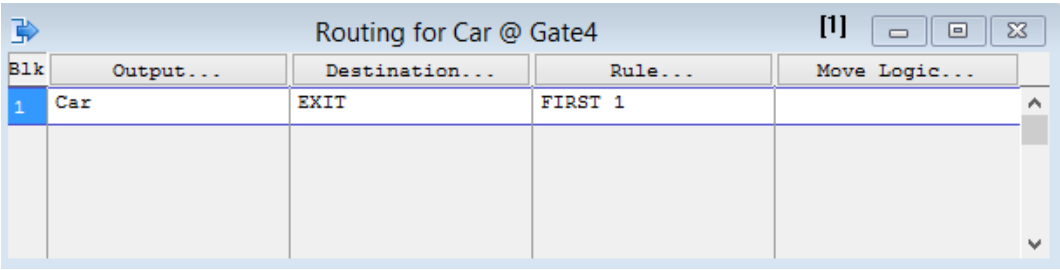
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.13.

Tabel 5.13. Tabel *Process* dan *Routing* Proses ke-11

Entitas	Car
Lokasi	Gate4
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.13 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.13 pada *Routing* ditunjukan pada gambar 5.15.



Gambar 5.15. Model *Routing* Proses ke-11

(Sumber: Hasil pengolahan data)

12. Entitas Car berada di lokasi Gate6

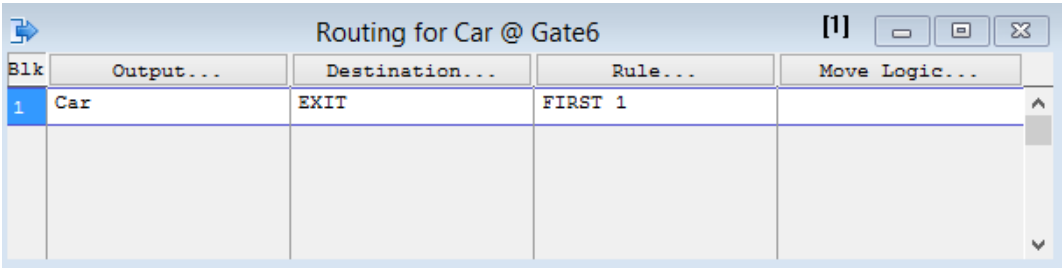
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.14.

Tabel 5.14. Tabel *Process* dan *Routing* Proses ke-12

Entitas	Car
Lokasi	Gate6
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.14 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.14 pada *Routing* ditunjukan pada gambar 5.16.



Gambar 5.16. Model *Routing* Proses ke-12

(Sumber: Hasil pengolahan data)

13. Entitas Car berada di lokasi Gate8

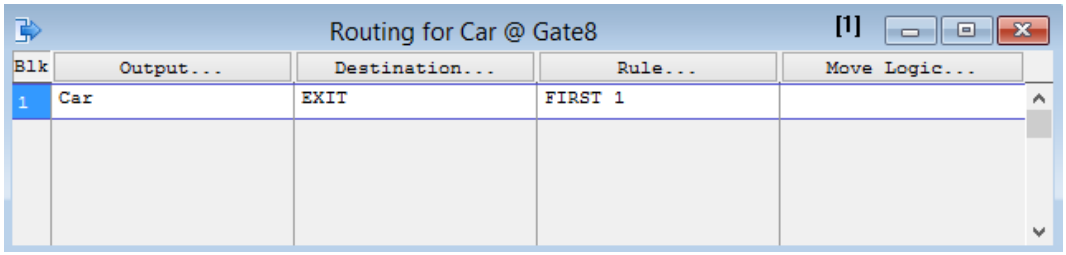
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.15.

Tabel 5.15. Tabel *Process* dan *Routing* Proses ke-13

Entitas	Car
Lokasi	Gate8
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.15 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.15 pada *Routing* ditunjukan pada gambar 5.17.



Gambar 5.17. Model *Routing* Proses ke-13

(Sumber: Hasil pengolahan data)

14. Entitas Car berada di lokasi Gate10

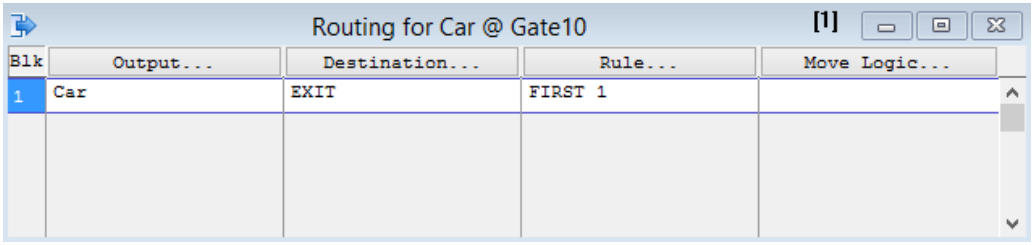
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.16.

Tabel 5.16. Tabel *Process* dan *Routing* Proses ke-14

Entitas	Car
Lokasi	Gate10
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.16 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.16 pada *Routing* ditunjukan pada gambar 5.18.



Gambar 5.18. Model *Routing* Proses ke-14

(Sumber: Hasil pengolahan data)

15. Entitas Car berada di lokasi Gate12

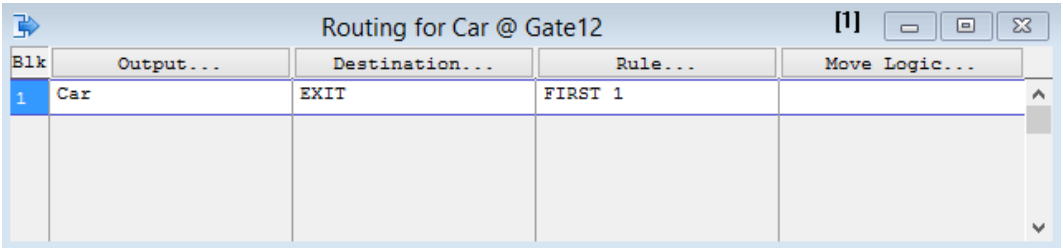
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.17.

Tabel 5.17. Tabel *Process* dan *Routing* Proses ke-15

Entitas	Car
Lokasi	Gate12
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.17 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.17 pada *Routing* ditunjukan pada gambar 5.19.



Gambar 5.19. Model *Routing* Proses ke-15

(Sumber: Hasil pengolahan data)

16. Entitas Car berada di lokasi Gate14

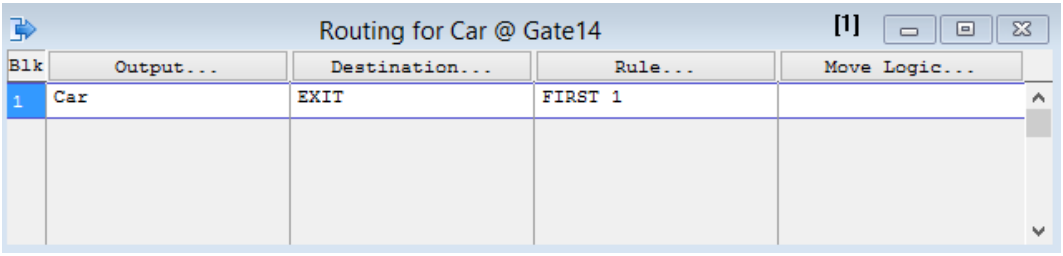
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.18.

Tabel 5.18. Tabel *Process* dan *Routing* Proses ke-16

Entitas	Car
Lokasi	Gate14
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.18 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.18 pada *Routing* ditunjukan pada gambar 5.20.



Gambar 5.20. Model *Routing* Proses ke-16

(Sumber: Hasil pengolahan data)

17. Entitas Car berada di lokasi Gate16

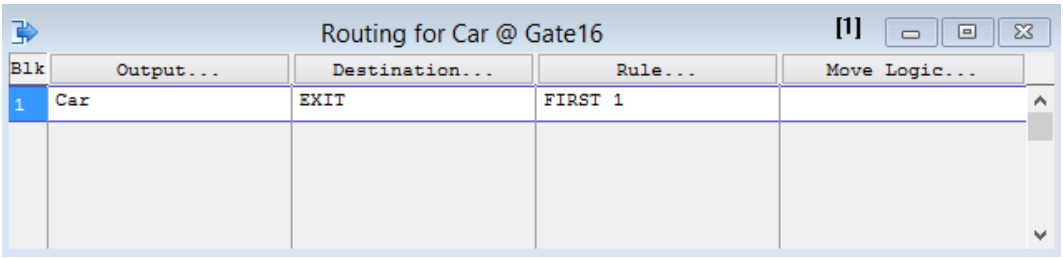
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.19.

Tabel 5.19. Tabel *Process* dan *Routing* Proses ke-17

Entitas	Car
Lokasi	Gate16
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.19 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.19 pada *Routing* ditunjukan pada gambar 5.21.



Gambar 5.21. Model *Routing* Proses ke-17

(Sumber: Hasil pengolahan data)

18. Entitas Car berada di lokasi Gate18

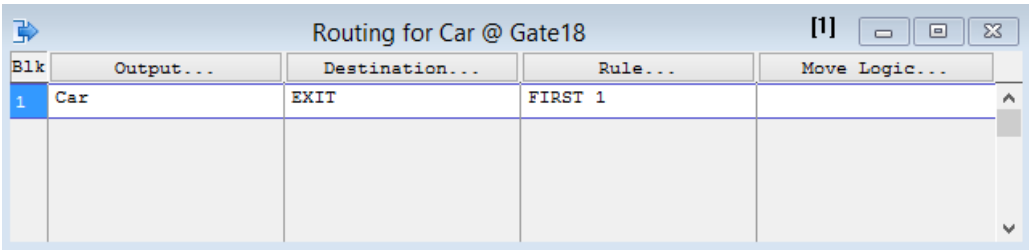
Rincian *Process* dan *Routing* untuk aktivitas ini ditunjukan pada tabel 5.20.

Tabel 5.20. Tabel *Process* dan *Routing* Proses ke-18

Entitas	Car
Lokasi	Gate18
Operasi	Deskripsi: Entitas melakukan aktivitas pembayaran tol *Operation Logic*: `Wait Mcr_process_time sec`
Output Entitas	Car
Lokasi Tujuan	EXIT
Aturan Perpindahan	EXIT : First 1
Logika Perpindahan	-

(Sumber: Hasil pengolahan data)

Data pada tabel 5.20 kemudian dimasukan pada *Routing*. Hasil pemasukan data tabel 5.20 pada *Routing* ditunjukan pada gambar 5.22.

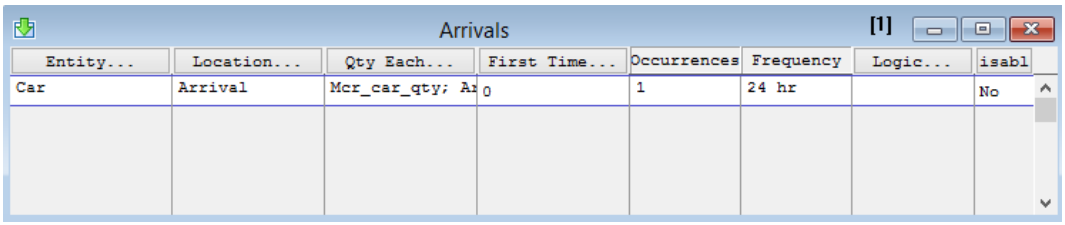


Gambar 5.22. Model *Routing* Proses ke-18

(Sumber: Hasil pengolahan data)

5.1.1.4. *Arrivals*

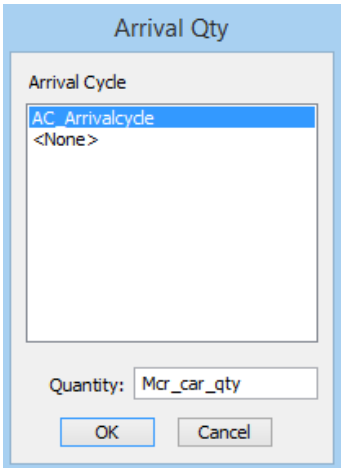
Elemen *Arrivals* menunjukan kedatangan entitas ke dalam sistem. Pada elemen *Arrivals*, data yang digunakan adalah data distribusi kedatangan harian dan data siklus kedatangan. Pengaturan *Arrivals* pada ProModel ditunjukan pada gambar 5.23.



Gambar 5.23. Model *Arrivals*

(Sumber: Hasil pengolahan data)

Pada gambar 5.23, kolom pertama menunjukan entitas yang didatangkan pada sistem, yaitu entitas *Car*. Pada kolom kedua ditunjukan lokasi dimana entitas tersebut didatangkan, yaitu di lokasi *Arrival*. Pada kolom ketiga ditunjukan jumlah entitas pada setiap kedatangan, dalam hal ini ditentukan juga siklus kedatangan dan jumlah entitas yang didatangkan. Pengaturan lebih lanjut ditunjukan pada gambar 5.24, pada gambar tersebut jendela *Arrival Qty* menunjukan siklus kedatangan yang dipilih yaitu AC_Arrivalcycle dengan kuantitas sebesar Mcr_car_qty.



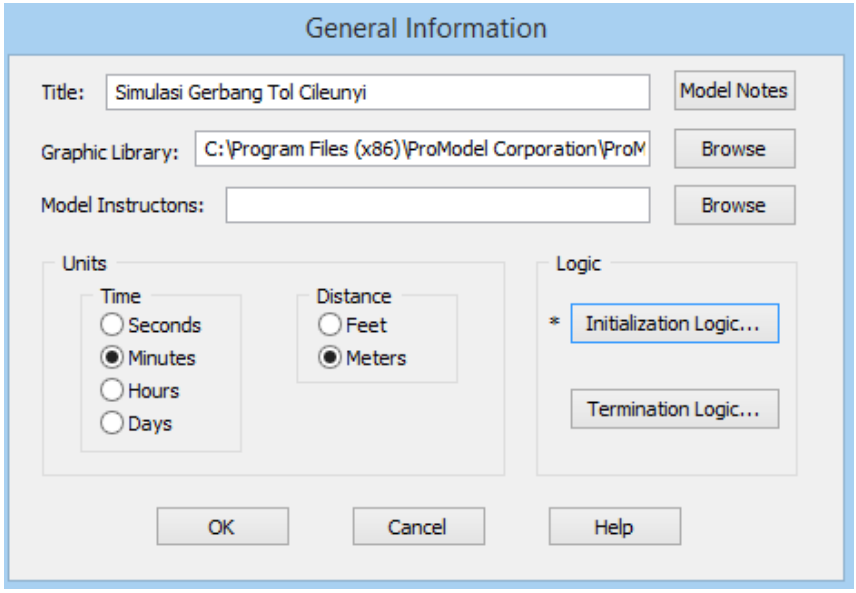
Gambar 5.24. Jendela *Arrival Qty*

(Sumber: Hasil pengolahan data)

Adapun kolom kelima menunjukan berapa kali kedatangan entitas dilakukan, dalam hal ini diatur 1 kali dengan frekuensi kedatangan pada kolom keenam setiap 24 jam. Diatur 1 kali setiap 24 jam karena data yang dimasukkan untuk kuantitas kendaraan adalah kuantitas kendaraan selama satu hari.

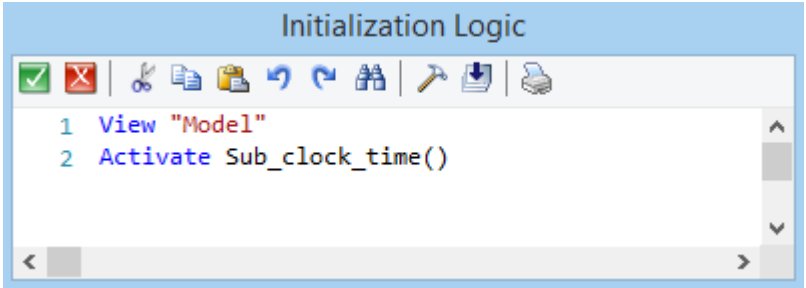
5.1.1.5. *General Information*

General information menunjukan informasi umum seputar model simulasi. General information untuk model simulasi yang dibuat ditunjukan pada gambar 5.25. Sedangkan untuk *initialization logic* model simulasi ditunjukan pada gambar 5.26.



Gambar 5.25. Model *General Information*

(Sumber: Hasil pengolahan data)

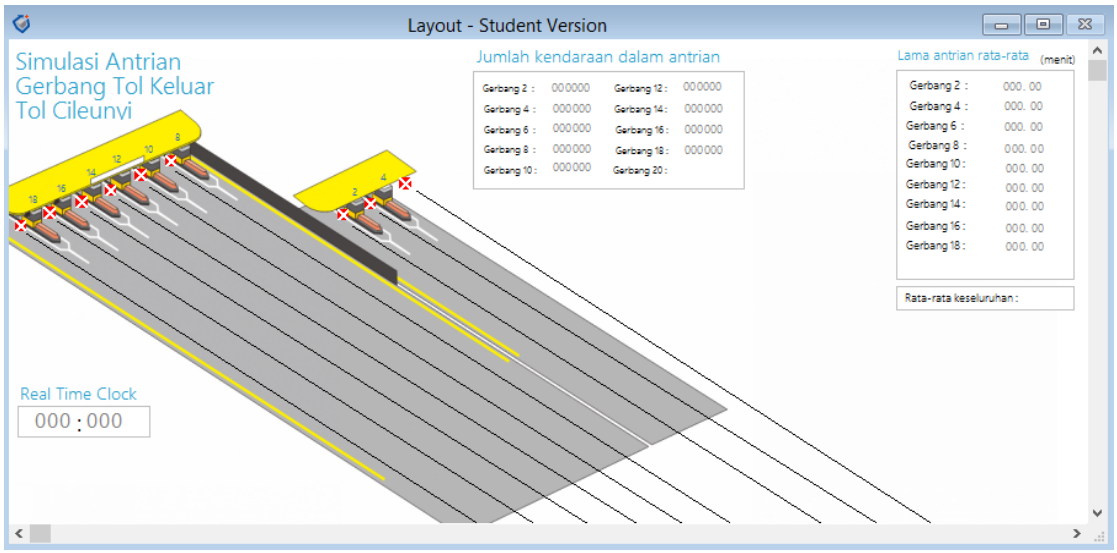


Gambar 5.26. Model *Initialization Logic*

(Sumber: Hasil pengolahan data)

5.1.1.6. *Background Graphic*

Background graphic yang digunakan pada model simulasi yang dibuat ditunjukan pada gambar 5.27.



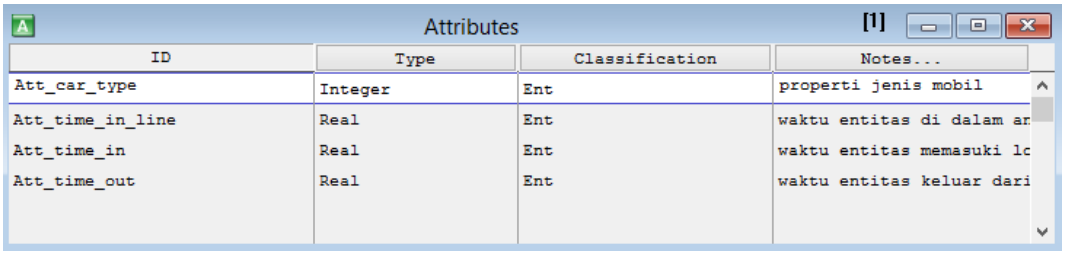
Gambar 5.27. Model *Background Graphic*

(Sumber: Hasil pengolahan data)

5.1.2. Elemen-elemen Lanjutan

5.1.2.1. *Attributes*

Attributes menunjukan properti yang melekat pada suatu entitas atau lokasi. Di dalam model simulasi yang dibuat, *Attributes* digunakan untuk mengetahui waktu ketika entitas memasuki sistem, waktu ketika keluar sistem, klasifikasi entitas, dan waktu ketika entitas berada di dalam antrian. Pengaturan *Attributes* pada ProModel ditunjukan pada gambar 5.28. Pada model simulasi yang dibuat setiap nama elemen atribut diberikan penanda (Att_) untuk memudahkan identifikasi pada saat digunakan di *logic builder*.



Gambar 5.28. Model *Attributes*

(Sumber: Hasil pengolahan data)

Pada gambar 5.28, kolom pertama menunjukan nama ID atribut yang ada, kolom kedua menunjukan tipe data atribut tersebut dan kolom ketiga menunjukan jenis atribut (entitas atau lokasi) sedangkan kolom terakhir menunjukan penjelasan lebih mengenai atribut. Adapun fungsi setiap atribut yang digunakan ditunjukan pada tabel 5.21.

Tabel 5.21. Penjelasan fungsi atribut

ID	Fungsi
Att_car_qty	Digunakan sebagai properti yang menunjukan jenis mobil yang memasuki sistem.
Att_time_in	Digunakan untuk menunjukan waktu entitas ketika memasuki lokasi antrian, diperlukan untuk menentukan lama entitas berada di dalam lokasi tersebut.
Att_time_in	Digunakan untuk menunjukan waktu entitas ketika keluar dari suatu lokasi antrian, diperlukan untuk menentukan lama entitas berada di dalam lokasi tersebut.
Att_time_in_line	Digunakan untuk menunjukan lama suatu entitas berada di dalam antrian, diperlukan untuk mengidentifikasi waktu rata-rata entitas mengantri di dalam lokasi antrian.

5.1.2.2. *Global Variables*

Global variables merupakan suatu wadah untuk menampung perubahan data numeris yang terjadi di dalam simulasi. *Global variables* digunakan ketika model simulasi membutuhkan penyimpanan data kuantitatif yang sifatnya dinamis. *Global variables* yang digunakan di dalam model simulasi ditunjukan pada gambar 5.28 dan 5.29

Variables (global) [1]

Icon	ID	Type	Initial value	Stats	Notes...
Yes	Var_qty_queue_gate2	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate4	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate6	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate8	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate10	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate12	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate14	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate16	Integer	0	Time Seri	menampung data
Yes	Var_qty_queue_gate18	Integer	0	Time Seri	menampung data
Yes	Var_avgtime_gate2	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate4	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate6	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate8	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate10	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate12	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate14	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate16	Real	0	Time Seri	menampung data
Yes	Var_avgtime_gate18	Real	0	Time Seri	menampung data
No	Var_car_processed_gate2	Integer	0	Time Seri	menampung data

Gambar 5.29. Model *Global Variables 1*

(Sumber: Hasil pengolahan data)

Variables (global) [20]

Icon	ID	Type	Initial value	Stats	Notes...
No	Var_car_processed_gate4	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate6	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate8	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate10	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate12	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate14	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate16	Integer	0	Time Seri	menampung data
No	Var_car_processed_gate18	Integer	0	Time Seri	menampung data
No	Var_totaltime_gate2	Real	0	Time Seri	menampung data
No	Var_totaltime_gate4	Real	0	Time Seri	menampung data
No	Var_totaltime_gate6	Real	0	Time Seri	menampung data
No	Var_totaltime_gate8	Real	0	Time Seri	menampung data
No	Var_totaltime_gate10	Real	0	Time Seri	menampung data
No	Var_totaltime_gate12	Real	0	Time Seri	menampung data
No	Var_totaltime_gate14	Real	0	Time Seri	menampung data
No	Var_totaltime_gate16	Real	0	Time Seri	menampung data
No	Var_totaltime_gate18	Real	0	Time Seri	menampung data
Yes	Var_clock_hour	Integer	0	None	menampung data
Yes	Var_clock_min	Integer	0	None	menampung data

Gambar 5.30. Model *Global Variables 2*

(Sumber: Hasil pengolahan data)

Pada gambar 5.29 dan gambar 5.30 kolom pertama *Icon* menunjukan apakah hasil perubahan nilai pada *global variables* ditampilkan pada model atau tidak. Kolom ID menunjukan nama variabel, kolom *type* menunjukan tipe data variabel, kolom *initial value* menunjukan nilai awal dari variabel, kolom *stats* menunjukan bagaimana data akan dikumpulkan, dan kolom *notes* menunjukan penjelasan lebih mengenai variabel. Setiap *global variable* yang digunakan diberikan penanda (Var_) untuk memudahkan identifikasi pada saat digunakan di *logic builder*. Adapun fungsi setiap variabel yang digunakan ditunjukan pada tabel 5.22.

Tabel 5.22. Penjelasan fungsi variabel global

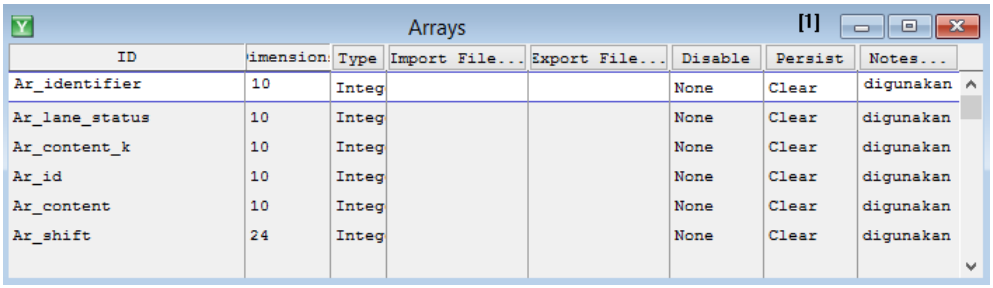
ID	Fungsi
Var_qty_queue_gate2	Untuk menampung data jumlah entitas di dalam antrian gardu 2
Var_qty_queue_gate4	Untuk menampung data jumlah entitas di dalam antrian gardu 4
Var_qty_queue_gate6	Untuk menampung data jumlah entitas di dalam antrian gardu 6
Var_qty_queue_gate8	Untuk menampung data jumlah entitas di dalam antrian gardu 8
Var_qty_queue_gate10	Untuk menampung data jumlah entitas di dalam antrian gardu 10
Var_qty_queue_gate12	Untuk menampung data jumlah entitas di dalam antrian gardu 12
Var_qty_queue_gate14	Untuk menampung data jumlah entitas di dalam antrian gardu 14
Var_qty_queue_gate16	Untuk menampung data jumlah entitas di dalam antrian gardu 16
Var_qty_queue_gate18	Untuk menampung data jumlah entitas di dalam antrian gardu 18
Var_avgtime_gate2	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 2
Var_avgtime_gate4	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 4
Var_avgtime_gate6	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 6
Var_avgtime_gate8	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 8
Var_avgtime_gate10	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 10
Var_avgtime_gate12	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 12
Var_avgtime_gate14	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 14
Var_avgtime_gate16	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 16
Var_avgtime_gate18	Untuk menampung data waktu rata-rata entitas berada di dalam antrian gardu 18
Var_car_processed_gate2	Untuk menampung data total entitas yang telah diproses di gardu 2
Var_car_processed_gate4	Untuk menampung data total entitas yang telah diproses di gardu 4
Var_car_processed_gate6	Untuk menampung data total entitas yang telah diproses di gardu 6
Var_car_processed_gate8	Untuk menampung data total entitas yang telah diproses di gardu 8

Tabel 5.22. Penjelasan fungsi variabel global (lanjutan)

ID	Fungsi
Var_car_processed_gate10	Untuk menampung data total entitas yang telah diproses di gardu 10
Var_car_processed_gate12	Untuk menampung data total entitas yang telah diproses di gardu 12
Var_car_processed_gate14	Untuk menampung data total entitas yang telah diproses di gardu 14
Var_car_processed_gate16	Untuk menampung data total entitas yang telah diproses di gardu 16
Var_car_processed_gate18	Untuk menampung data total entitas yang telah diproses di gardu 18
Var_totaltime_gate2	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 2
Var_totaltime_gate4	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 4
Var_totaltime_gate6	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 6
Var_totaltime_gate8	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 8
Var_totaltime_gate10	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 10
Var_totaltime_gate12	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 12
Var_totaltime_gate14	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 14
Var_totaltime_gate16	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 16
Var_totaltime_gate18	Untuk menampung data total waktu antrian seluruh entitas ketika berada di dalam antrian gardu 18
Var_clock_hour	Untuk menampung data perubahan jam pada simulasi
Var_clock_min	Untuk menampung data perubahan menit pada simulasi

5.1.2.3. *Arrays*

Arrays merupakan matriks sekumpulan baris yang memiliki nilai. Setiap baris *array* memiliki fungsi yang sama dengan variabel. Array yang digunakan di dalam model simulasi ditunjukan pada gambar 5.31.



Gambar 5.31. Model *Arrays*

(Sumber: Hasil pengolahan data)

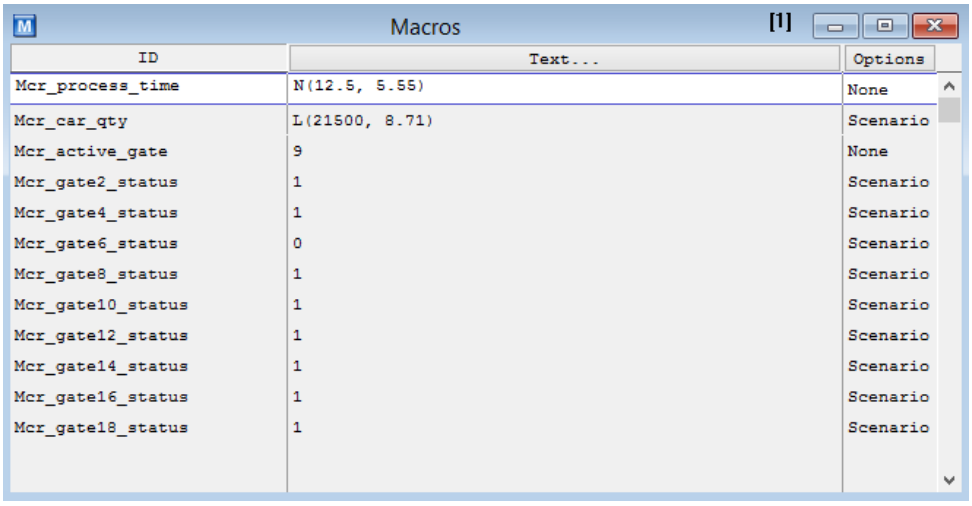
Gambar 5.31 menunjukan pengaturan *Arrays* pada ProModel. Kolom pertama menunjukan nama ID *array*, kolom kedua menunjukan ukuran dan dimensi *array*, kolom ketiga menunjukan tipe data *array*, kolom keempat dan kelima berfungsi untuk *import* dan *export* data array. Kolom *disable* menunjukan penggunaan *array* pada model, kolom *persist* menunjukan pilihan penyimpanan data pada setiap replikasi, dan kolom *notes* menunjukan penjelasan tambahan mengenai *array*. *Arrays* tersebut digunakan di dalam *subroutine*, untuk memudahkan proses pembuatan logika pemrograman. Untuk setiap *array* diberikan penanda Ar_ untuk memudahkan pemanggilan di *logic builder*. Penjelasan fungsi setiap *array* yang digunakan di dalam model ditunjukan pada tabel 5.23.

Tabel 5.23. Penjelasan fungsi *array*

ID	Fungsi
Ar_identifier	Digunakan sebagai identifikasi gardu yang memiliki tingkat antrian terkecil, digunakan di dalam *subroutine* Sub_select_lane
Ar_lane_status	Digunakan untuk mengidentifikasi status setiap gardu yang dibuka pada awal simulasi, digunakan di dalam *subroutine* Sub_select_lane
Ar_content_k	Digunakan untuk mengidentifikasi gardu yang memiliki antrian terkecil yang sama, digunakan di dalam *subroutine* Sub_select_lane
Ar_id	Digunakan untuk menandai gardu yang terbuka, digunakan di dalam *subroutine* Sub_select_lane
Ar_content	Digunakan untuk mengidentifikasi tingkat antrian pada setiap gardu, digunakan di dalam *subroutine* Sub_select_lane
Ar_shift	Digunakan untuk menandai *shift,* digunakan di dalam *subroutine* Sub_shifthour-

5.1.2.4. *Macros*

Macros merupakan elemen yang digunakan untuk menggantikan suatu nilai, persamaan, fungsi, atau teks, berfungsi juga sebagai parameter model yang dapat digunakan dalam percobaan skenario. Digunakan untuk memudahkan pergantian nilai variabel di dalam *logic field*. *Macros* yang digunakan di dalam model simulasi ditunjukan pada gambar 5.31.



Gambar 5.32. Model *Macros*

(Sumber: Hasil pengolahan data)

Pada gambar 5.32, kolom ID menunjukan nama *macro* yang digunakan, kolom *text* merupakan nilai yang akan digantikan oleh ID *macro*, dan kolom *options* menunjukan pilihan untuk menjadikan *macro* sebagai parameter skenario atau tidak. Untuk setiap *macro* diberikan penanda Mcr_ untuk memudahkan pemanggilan di *logic builder*. Adapun penjelasan mengenai fungsi setiap *macro* ditunjukan pada tabel 5.24.

Tabel 5.23. Penjelasan fungsi *macro*

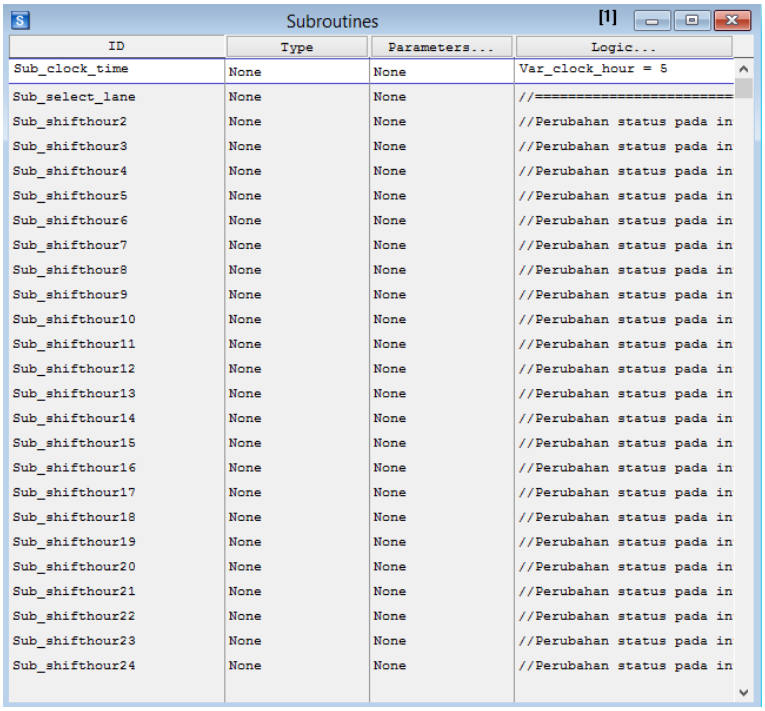
ID	Fungsi
Mcr_process_time	Digunakan sebagai pengganti nilai waktu proses, diisi dengan distribusi waktu proses pembayaran hasil analisis data pada bab sebelumnya
Mcr_car_qty	Digunakan sebagai pengganti nilai jumlah mobil harian, diisi dengan distribusi jumlah mobil harian hasil analisis data pada bab sebelumnya
Mcr_active_gate	Digunakan sebagai pengganti jumlah gardu yang aktif, diisi dengan nilai 9 sesuai dengan jumlah gardu
Mcr_gate2_status	Digunakan sebagai pengganti nilai status gardu 2 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate4_status	Digunakan sebagai pengganti nilai status gardu 4 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate6_status	Digunakan sebagai pengganti nilai status gardu 6 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate8_status	Digunakan sebagai pengganti nilai status gardu 8 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif

Tabel 5.23. Penjelasan fungsi *macro* (lanjutan)

ID	Fungsi
Mcr_gate10_status	Digunakan sebagai pengganti nilai status gardu 10 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate12_status	Digunakan sebagai pengganti nilai status gardu 12 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate14_status	Digunakan sebagai pengganti nilai status gardu 1 pada awal simulasi 4, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate16_status	Digunakan sebagai pengganti nilai status gardu 16 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif
Mcr_gate18_status	Digunakan sebagai pengganti nilai status gardu 18 pada awal simulasi, diisi dengan angka 1 jika gardu aktif, dan angka 0 jika gardu tidak aktif

5.1.2.5. *Subroutines*

Subroutine merupakan sekumplan logika yang dibuat oleh pemodel dalam barisan kode yang berisi proses-proses yang dapat dipanggil di dalam model. *Subroutine* dibuat sebagai alat untuk menyatakan algoritma dari proses-proses logis yang ada pada sistem. Pada setiap *subroutine*, diberi penanda Sub_ untuk memudahkan pemanggilan pada *logic builder*. Adapun *subroutine* yang terlibat di dalam model simulasi yang dibuat ditunjukan pada gambar 5.33.



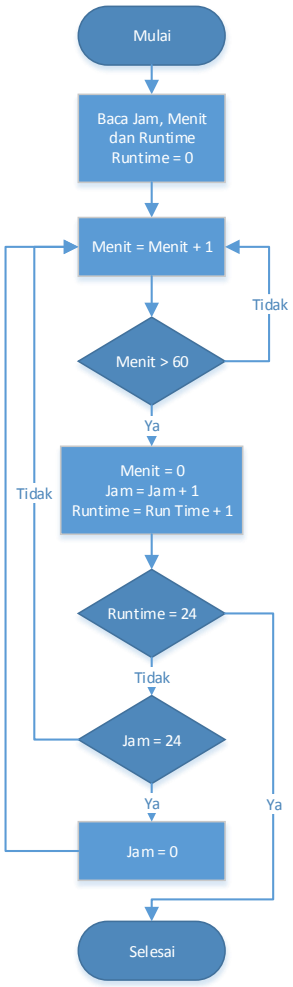
Gambar 5.33. Model *Subroutines*

(Sumber: Hasil pengolahan data)

Pada gambar 5.33, kolom ID menunjukan nama *subroutine* yang digunakan, kolom type menunjukan tipe data jika subroutine menghasilkan suatu nilai, kolom parameter menunjukan variabel yang dijadikan *input* pada *subroutine* dan kolom terakhir menunjukan *logic field* yang digunakan untuk mengisi *subroutine* dengan baris kode. Adapun isi *logic field* dan fungsi dari setiap subroutine adalah sebagai berikut:

1. Sub_clock_time

   Subroutine Sub_clock_time dibuat untuk mengidentifikasi waktu jam pada saat model dijalankan. Algoritma yang digunakan ditunjukan pada gambar 5.34.



Gambar 5.34. *Flowchart* Algoritma *Subroutine* Sub_clock_time

(Sumber: Hasil pengolahan data)

Algoritma tersebut kemudian ditranslasikan menjadi kode yang dimasukan di dalam *logic field*. Adapun hasil translasi algoritma Sub_clock_time ditunjukan pada tabel 5.24.

Tabel 5.24. *Logic Field Subroutine* Sub_clock_time

```
Var_clock_hour = 5                //<============= Jam dan
Var_clock_min = 30                //<============= Menit awal simulasi
Int x = 1
While x > 0 Do
{
    While Var_clock_min < 60 Do
    {
        Wait 1 Min
        Inc Var_clock_min
    }
    Inc Var_clock_hour
    Var_clock_min = 0
    If Var_clock_hour = 24 Then
    {
        Var_clock_hour = 0
    }
}
```

(Sumber: Hasil pengolahan data)

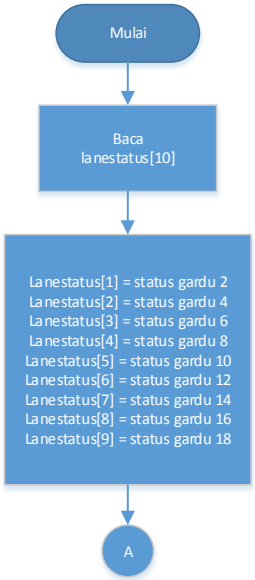
2. Sub_select_lane

   Subroutine Sub_select_lane dibuat untuk merepresentasikan proses pemilihan lajur oleh entitas pada model simulasi, berbeda dengan *subroutine* lainnya, *subroutine* Sub_select_lane terdiri dari beberapa sekumpulan prosedur dalam memilih lajur. Tahapan prosedur pemilihan lajur adalah sebagai berikut:

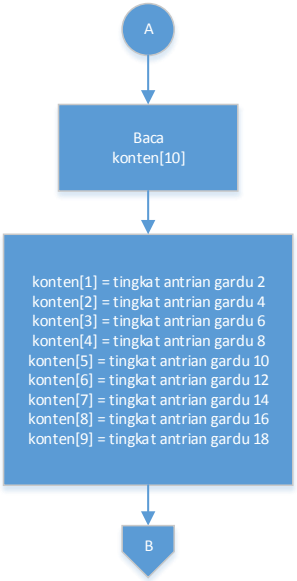
   1. Identifikasi status awal setiap lajur ketika simulasi dimulai
   2. Identifikasi konten setiap gardu tersedia
   3. Identifikasi urutan dan konten gardu yang terbuka
   4. Mengacak jenis mobil yang muncul
   5. Mengurutkan gardu terbuka berdasarkan konten minimum

6. Mengacak rute pemilihan lajur ketika ada gardu dengan konten minimum yang sama

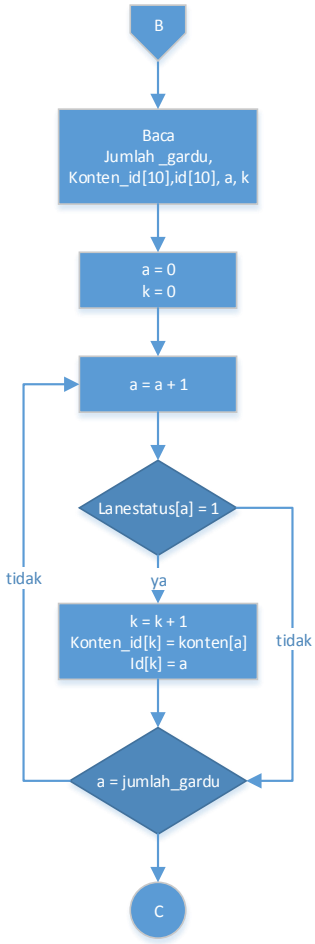
Adapun *flowchart* yang menggambarkan algoritma dari setiap prosedur adalah sebagai berikut:



Gambar 5.35. *Flowchart* Algoritma Prosedur 1 *Subroutine* Sub_select_lane
(Sumber: Hasil pengolahan data)

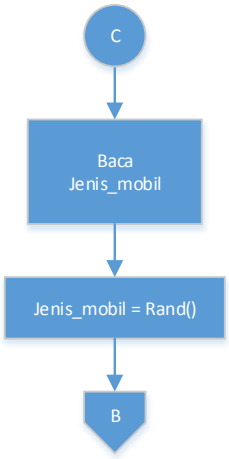


Gambar 5.36. *Flowchart* Algoritma Prosedur 2 *Subroutine* Sub_select_lane
(Sumber: Hasil pengolahan data)



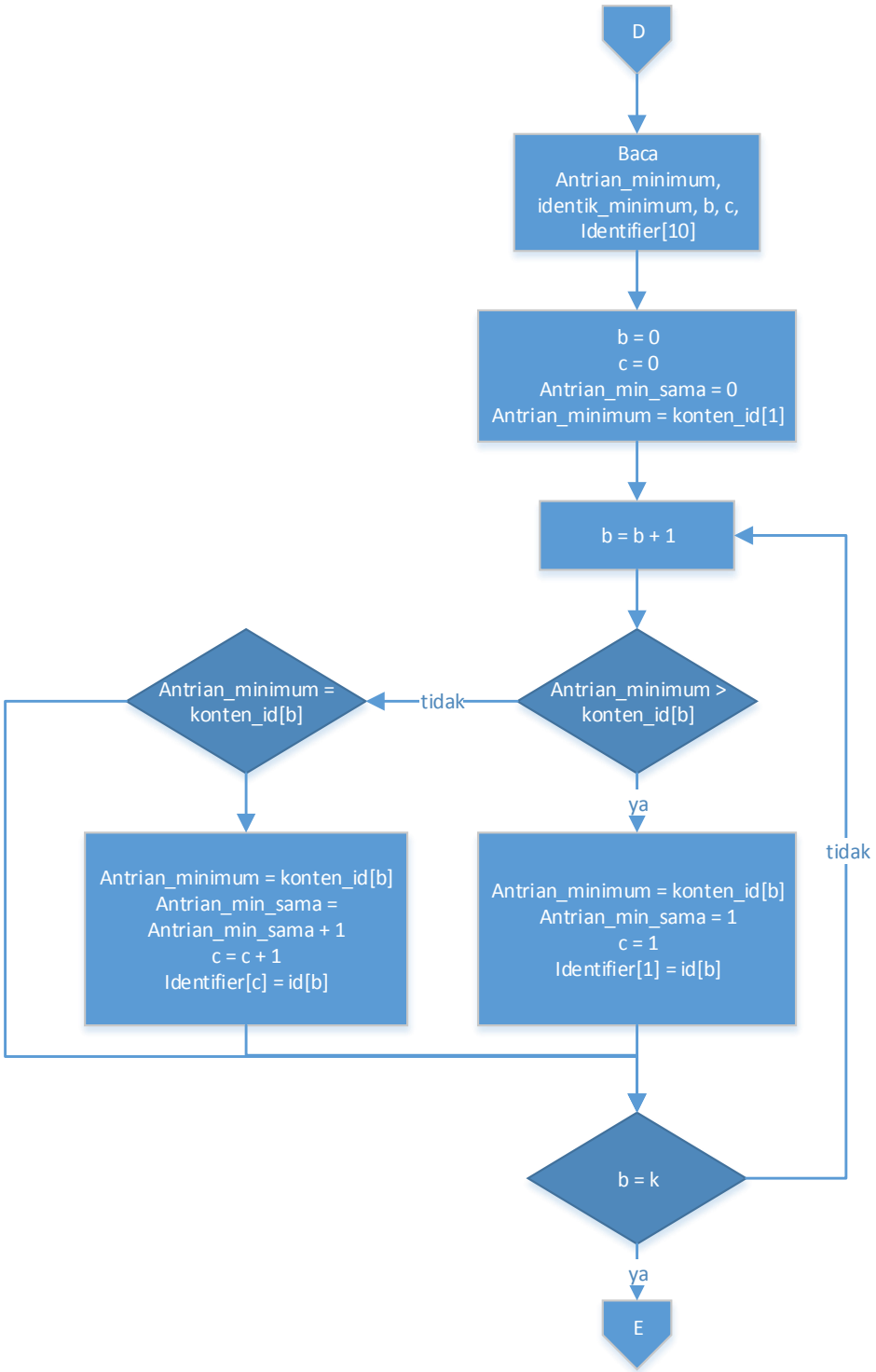
Gambar 5.37. *Flowchart* Algoritma Prosedur 3 *Subroutine* Sub_select_lane

(Sumber: Hasil pengolahan data)



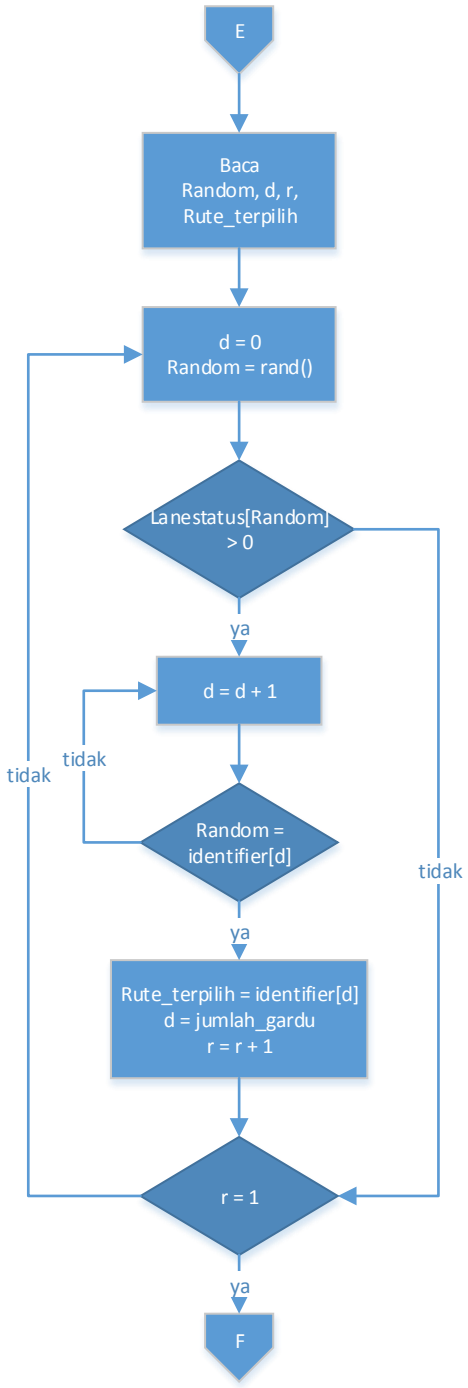
Gambar 5.38. *Flowchart* Algoritma Prosedur 4 *Subroutine* Sub_select_lane

(Sumber: Hasil pengolahan data)



Gambar 5.39. *Flowchart* Algoritma Prosedur 5 *Subroutine* Sub_select_lane

(Sumber: Hasil pengolahan data)



Gambar 5.40. *Flowchart* Algoritma Prosedur 6 *Subroutine* Sub_select_lane

(Sumber: Hasil pengolahan data)

Algoritma tersebut kemudian ditranslasikan menjadi kode yang dimasukan di dalam *logic field*. Adapun hasil translasi algoritma Sub_select_lane ditunjukan pada tabel 5.25.

Tabel 5.25. *Logic Field Subroutine* Sub_select_lane

```
//reset array
Ar_identifier[1] = 0
Ar_identifier[2] = 0
Ar_identifier[3] = 0
Ar_identifier[4] = 0
Ar_identifier[5] = 0
Ar_identifier[6] = 0
Ar_identifier[7] = 0
Ar_identifier[8] = 0
Ar_identifier[9] = 0

//identifikasi status gardu awal
If Var_endloop < 1 Then
{
    Ar_lane_status[1] = Mcr_gate2_status
    Ar_lane_status[2] = Mcr_gate4_status
    Ar_lane_status[3] = Mcr_gate6_status
    Ar_lane_status[4] = Mcr_gate8_status
    Ar_lane_status[5] = Mcr_gate10_status
    Ar_lane_status[6] = Mcr_gate12_status
    Ar_lane_status[7] = Mcr_gate14_status
    Ar_lane_status[8] = Mcr_gate16_status
    Ar_lane_status[9] = Mcr_gate18_status
    Inc Var_endloop, 1
}

//identifikasi konten antrian setiap gardu
Ar_content[1] = contents(Queue_gate2)
Ar_content[2] = contents(Queue_gate4)
Ar_content[3] = contents(Queue_gate6)
Ar_content[4] = contents(Queue_gate8)
Ar_content[5] = contents(Queue_gate10)
Ar_content[6] = contents(Queue_gate12)
Ar_content[7] = contents(Queue_gate14)
Ar_content[8] = contents(Queue_gate16)
Ar_content[9] = contents(Queue_gate18)

//identifikasi gardu yang terbuka
Int local_a
Int local_k
local_a = 0
local_k = 0
```

(Sumber: Hasil pengolahan data)

Tabel 5.25. *Logic Field Subroutine* Sub_select_lane (lanjutan)

```
While local_a < Mcr_active_gate Do
{
    Inc local_a,1
    If Ar_lane_status[local_a] = 1 Then
    {
        Inc local_k,1
        Ar_content_id[local_k] = Ar_content[local_a]
        Ar_id[local_k] = local_a
    }
}

//menentukan jenis mobil yang muncul
Int local_graphic
Att_car_type = UDist_car_prob()

If Att_car_type = 1 Then
{
    local_graphic = rand(7)+2
    Graphic local_cartype_k
}
Else If Att_car_type = 2 then
{
    Graphic 1
}

//======================melakukan pemilihan antrian terpendek==================//
//mengurutkan konten antrian dari gardu terbuka

    int local_b
    Int local_c
    Int identic
    Int minval
    local_b = 0
    local_c = 0
    identic = 0
    minval = Ar_content_id[1]

    While local_b < local_k Do
    {
        Inc local_b,1
        If minval > Ar_content_id[local_b] Then
        {
            minval = Ar_content_id[local_b]
            identic = 1
            local_c = 1
            Ar_identifier[1] = Ar_id[local_b]
        }
```

(Sumber: Hasil pengolahan data)

Tabel 5.25. *Logic Field Subroutine* Sub_select_lane (lanjutan)

```
        Else If minval = Ar_content_id[local_b] Then
        {
            minval = Ar_content_id[local_b]
            identic = identic + 1
            local_c = local_c + 1
            Ar_identifier[local_c] = Ar_id[local_b]
        }
    }

    //melakukan acak kemungkinan pemilihan lane ketika ada konten
      minimum yang sama
    Int localrand
    Int local_d
    Int local_r

    While local_r < 1 Do
    {
        local_d = 0
        localrand = UDist_gate_prob()
        If Ar_lane_status[localrand] > 0 Then
        {
        While local_d < Mcr_active_gate Do
            {
                Inc local_d,1
                If localrand = Ar_identifier[local_d] Then
                {
                    Route Ar_identifier[local_d]
                    local_d = Mcr_active_gate
                    Inc local_r,1
                }
            }
        }
    }

//jalankan subroutine sub_shifthour

//Sub_shifthour2()          //Perubahan status pada interval waktu : 06.30 - 07.30
//Sub_shifthour3()          //Perubahan status pada interval waktu : 07.30 - 08.30
//Sub_shifthour4()          //Perubahan status pada interval waktu : 08.30 - 09.30
//Sub_shifthour5()          //Perubahan status pada interval waktu : 09.30 - 10.30
//Sub_shifthour6()          //Perubahan status pada interval waktu : 10.30 - 11.30
//Sub_shifthour7()          //Perubahan status pada interval waktu : 11.30 - 12.30
//Sub_shifthour8()          //Perubahan status pada interval waktu : 12.30 - 13.30
Sub_shifthour9()            //Perubahan status pada interval waktu : 13.30 - 14.30
//Sub_shifthour10()         //Perubahan status pada interval waktu : 14.30 - 15.30
//Sub_shifthour11()         //Perubahan status pada interval waktu : 15.30 - 16.30
//Sub_shifthour12()         //Perubahan status pada interval waktu : 16.30 - 17.30
//Sub_shifthour13()         //Perubahan status pada interval waktu : 17.30 - 18.30
```

(Sumber: Hasil pengolahan data)

Tabel 5.25. *Logic Field Subroutine* Sub_select_lane (lanjutan)

```
//Sub_shifthour14()          //Perubahan status pada interval waktu : 18.30 - 19.30
//Sub_shifthour15()          //Perubahan status pada interval waktu : 19.30 - 20.30
//Sub_shifthour16()          //Perubahan status pada interval waktu : 20.30 - 21.30
Sub_shifthour17()            //Perubahan status pada interval waktu : 21.30 - 22.30
//Sub_shifthour18()          //Perubahan status pada interval waktu : 22.30 - 23.30
//Sub_shifthour19()          //Perubahan status pada interval waktu : 23.30 - 00.30
//Sub_shifthour20()          //Perubahan status pada interval waktu : 00.30 - 01.30
//Sub_shifthour21()          //Perubahan status pada interval waktu : 01.30 - 02.30
//Sub_shifthour22()          //Perubahan status pada interval waktu : 02.30 - 03.30
//Sub_shifthour23()          //Perubahan status pada interval waktu : 03.30 - 04.30
//Sub_shifthour24()         //Perubahan status pada interval waktu : 04.30 - 05.30
```

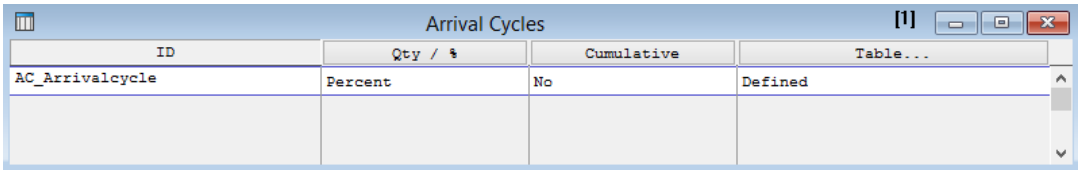
(Sumber: Hasil pengolahan data)

3. Sub_shift_hourN

   Sub_shift_hourN menunjukan perubahan status gardu pada waktu N. Perubahan status gardu dibuat per jam mulai dari jam pertama pada shift 1 pukul 05.30 hingga jam ke-24 pada pukul 04.30. Sehingga terdapat 24 kali kemungkinan perubahan status gardu dalam satu hari.

5.1.2.6. *Arrival Cycles*

Arrival cycles menunjukan siklus kedatangan entitas pada model. Data arrival cycles diisi berdasarkan hasil analisis data pada bab sebelumnya. Data tersebut kemudian dimasukan pada *arrival cycles editor* yang ditunjukan pada gambar 5.41 dan pengisian tabel ditunjukan pada gambar 5.42.



Gambar 5.41. *Arrival Cycles Editor* Model

(Sumber: Hasil pengolahan data)

Table for AC_Arrivalcycle [1]

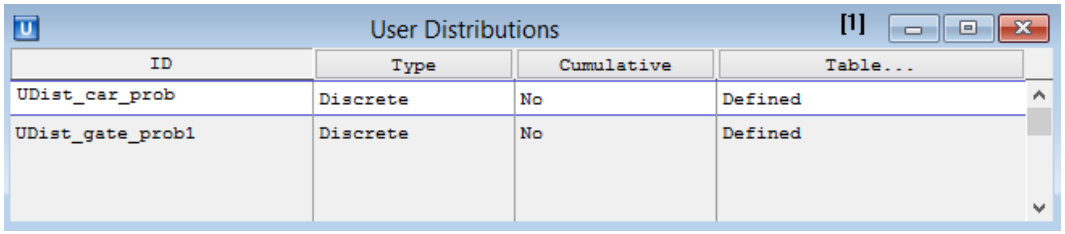
Time (Hours)	Qty / %
1	3.04
2	3.98
3	4.98
4	5.36
5	5.68
6	5.61
7	5.20
8	4.82
9	5.38
10	5.69
11	5.96
12	6.22
13	5.97
14	5.36
15	4.87
16	4.17
17	3.66
18	2.90
19	2.36
20	2.10
21	1.71
22	1.56
23	1.63
24	1.79

Gambar 5.42. Tabel *Arrival Cycles* Model

(Sumber: Hasil pengolahan data)

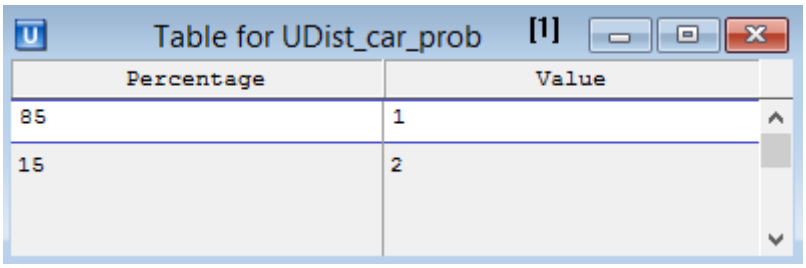
5.1.2.7. *User Distributions*

User distribution menunjukan distribusi probabilitas yang dibuat oleh pemodel. *User distribution* yang dibuat dalam model simulasi adalah probabilitas golongan kendaraan dan probabilitas pemilihan lajur. Pengaturan *user distribution* pada model ditunjukan pada gambar 5.43, sedangkan pengisian tabel distribusi untuk kedua distribusi ditunjukan pada gambar 5.44 dan 5.45.



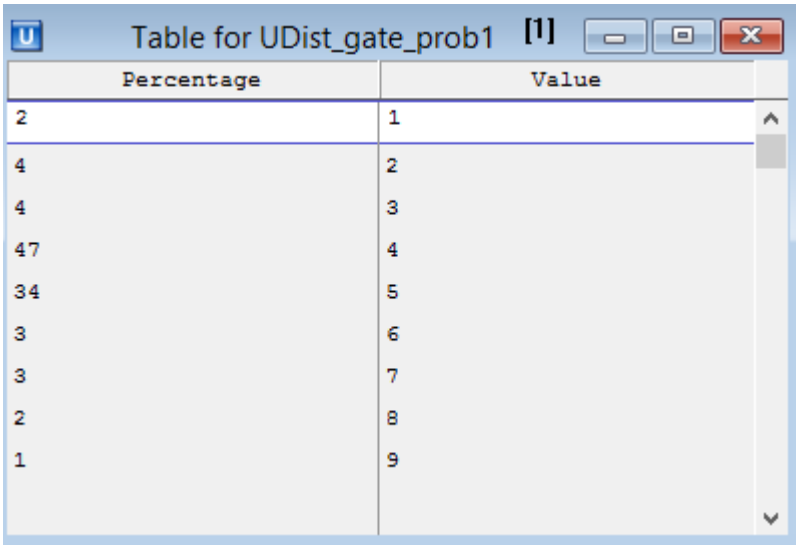
Gambar 5.43. *User Distribution* Model

(Sumber: Hasil pengolahan data)



Gambar 5.44. Tabel *User Distribution* probabilitas jenis kendaraan

(Sumber: Hasil pengolahan data)



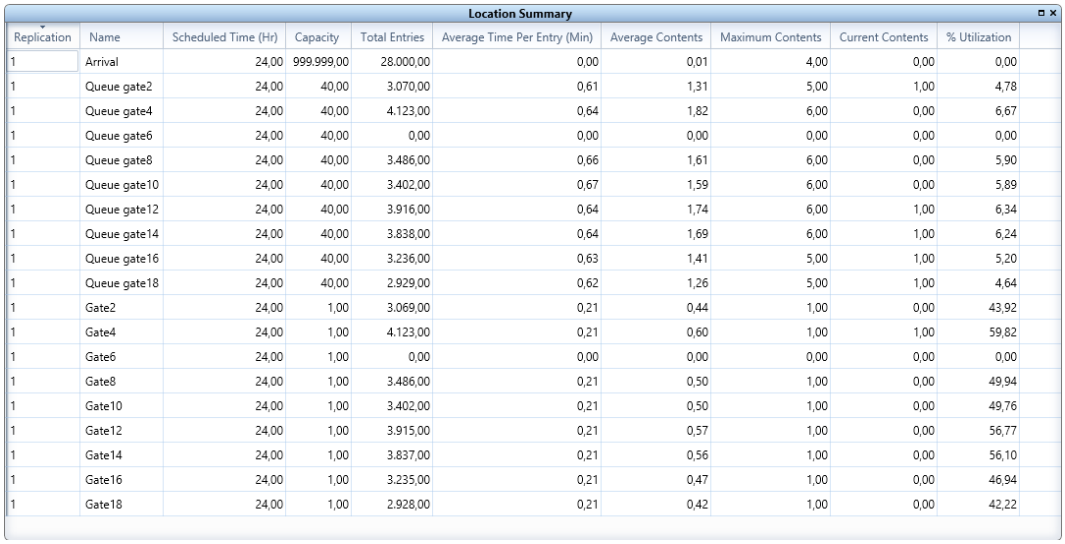
Gambar 5.45. Tabel *User Distribution* probabilitas pemilihan lajur

(Sumber: Hasil pengolahan data)

Tabel 5.44 diisi berdasarkan data golongan kendaraan, sedangkan data pada tabel 5.45 diisi berdasarkan hasil *trial and error* agar model yang dibuat valid.

5.2. Uji Coba Model

Setelah model dibuat langkah selanjutnya adalah menguji coba model untuk mendapatkan nilai statistik awal untuk keperluan eksperimen. Untuk uji coba, model dijalankan selama 24 jam dengan 1 replikasi. Data hasil percobaan ditunjukan pada gambar 5.46.



Location Summary

Replication	Name	Scheduled Time (Hr)	Capacity	Total Entries	Average Time Per Entry (Min)	Average Contents	Maximum Contents	Current Contents	% Utilization
1	Arrival	24,00	999.999,00	28.000,00	0,00	0,01	4,00	0,00	0,00
1	Queue gate2	24,00	40,00	3.070,00	0,61	1,31	5,00	1,00	4,78
1	Queue gate4	24,00	40,00	4.123,00	0,64	1,82	6,00	0,00	6,67
1	Queue gate6	24,00	40,00	0,00	0,00	0,00	0,00	0,00	0,00
1	Queue gate8	24,00	40,00	3.486,00	0,66	1,61	6,00	0,00	5,90
1	Queue gate10	24,00	40,00	3.402,00	0,67	1,59	6,00	0,00	5,89
1	Queue gate12	24,00	40,00	3.916,00	0,64	1,74	6,00	1,00	6,34
1	Queue gate14	24,00	40,00	3.838,00	0,64	1,69	6,00	1,00	6,24
1	Queue gate16	24,00	40,00	3.236,00	0,63	1,41	5,00	1,00	5,20
1	Queue gate18	24,00	40,00	2.929,00	0,62	1,26	5,00	1,00	4,64
1	Gate2	24,00	1,00	3.069,00	0,21	0,44	1,00	0,00	43,92
1	Gate4	24,00	1,00	4.123,00	0,21	0,60	1,00	1,00	59,82
1	Gate6	24,00	1,00	0,00	0,00	0,00	0,00	0,00	0,00
1	Gate8	24,00	1,00	3.486,00	0,21	0,50	1,00	0,00	49,94
1	Gate10	24,00	1,00	3.402,00	0,21	0,50	1,00	0,00	49,76
1	Gate12	24,00	1,00	3.915,00	0,21	0,57	1,00	0,00	56,77
1	Gate14	24,00	1,00	3.837,00	0,21	0,56	1,00	0,00	56,10
1	Gate16	24,00	1,00	3.235,00	0,21	0,47	1,00	0,00	46,94
1	Gate18	24,00	1,00	2.928,00	0,21	0,42	1,00	0,00	42,22

Gambar 5.46. Hasil Uji Coba Model

(Sumber: Hasil pengolahan data)

5.3. Validasi Model

Validasi model dilakukan dengan membandingkan output hasil simulasi dengan *run time* 16 jam dengan data volume lalu lintas pada shift 1 dan 2. Perbandingan dilakukan menggunakan pengujian *t-student* satu sampel dengan hipotesis sebagai berikut:

H_0 : Kedua rata-rata populasi identik

H_1 : Kedua rata-rata populasi tidak identik

Pengujian tersebut dilakukan menggunakan bantuan *software SPSS* dengan kriteria penolakan H_0 apabila nilai probabilitas (sig) < 0.05. Data yang diuji adalah data hasil menjalankan model dengan *run time* 16 jam sebanyak 30 replikasi pada tabel 5.26 dan data volume lalu lintas pada shift 1 dan 2 pada tabel 5.27.

Tabel 5.26. Hasil *output* simulasi 16 jam 30 replikasi

Replikasi	Gardu								
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**
1.00	2348	3001	0	3389	3335	2858	2797	2410	2189
2.00	2226	2922	0	3293	3249	2753	2707	2322	2092
3.00	2741	3352	0	3690	3646	3224	3154	2810	2568
4.00	2781	3435	0	3720	3711	3244	3208	2872	2651
5.00	2302	2986	0	3353	3306	2796	2784	2387	2143
6.00	2607	3252	0	3594	3569	3075	3063	2692	2427
7.00	2466	3130	0	3498	3442	2958	2909	2552	2299
8.00	2487	3161	0	3511	3470	2994	2934	2570	2321
9.00	2921	3485	0	3805	3752	3359	3291	2979	2692
10.00	2339	3003	0	3398	3346	2856	2776	2392	2156
11.00	2593	3220	0	3589	3525	3054	3021	2655	2408
12.00	2214	2898	0	3275	3255	2732	2688	2293	2069
13.00	2143	2814	0	3206	3176	2674	2606	2200	2009
14.00	2517	3140	0	3516	3455	3040	2952	2573	2322
15.00	2575	3208	0	3541	3498	3054	2999	2626	2388
16.00	2276	2941	0	3318	3299	2786	2727	2376	2128
17.00	2555	3187	0	3550	3521	3065	2993	2660	2384
18.00	2316	2991	0	3382	3326	2833	2745	2385	2186
19.00	2505	3164	0	3520	3472	2984	2939	2598	2341
20.00	2591	3216	0	3590	3550	3079	3039	2658	2421
21.00	2400	3058	0	3438	3390	2914	2860	2472	2249
22.00	2251	2918	0	3315	3264	2760	2709	2330	2105
23.00	2300	2966	0	3368	3315	2824	2781	2387	2167
24.00	2201	2888	0	3267	3240	2734	2677	2286	2063
25.00	2401	3045	0	3438	3386	2902	2851	2475	2229
26.00	2398	3047	0	3453	3386	2892	2858	2468	2252
27.00	2625	3251	0	3614	3568	3093	3060	2694	2436
28.00	2602	3230	0	3575	3515	3088	3020	2672	2440
29.00	2533	3176	0	3521	3505	3032	2969	2614	2361
30.00	2632	3248	0	3600	3574	3107	3040	2678	2436

Pada tabel 5.26, nilai yang akan diuji adalah volume kendaraan pada gate2, gate4, gate8, gate10, gate12, gate14, gate16 dan gate18. Nilai tersebut dibandingkan dengan volume rata-rata pada setiap gardu pada tabel 5.27.

Tabel 5.27. Data volume kendaraan shift 1 dan 2 Februari dan Maret 2015

No	Gardu								
	2	**4**	**6**	**8**	**10**	**12**	**14**	**16**	**18**
1	2008	2204	0	3398	3471	2381	2634	2415	1911
2	2527	3209	0	3418	3803	2796	2755	2290	1947
3	2893	3325	0	3676	3613	2768	3242	2691	180
4	2978	3234	0	3634	3621	2765	2826	1644	1718
5	2924	3449	0	3775	3654	2968	3136	2439	518
6	3020	3408	0	3721	3625	3189	2713	2288	2082
7	3163	3027	0	3653	3425	3020	3211	2943	2720
8	3088	2895	0	2770	2538	2867	3223	2640	2076
9	2873	2,991	0	3612	3573	2787	2750	2357	1953
10	2628	2998	0	3669	3619	2247	2618	2216	2276
11	2766	2713	0	3401	3466	2661	2897	2458	2404
12	2079	2713	0	3617	3366	2959	2892	2639	2455
13	1943	3024	0	3671	3445	3047	2691	2808	2141
14	2757	3033	0	3658	3526	3109	3407	2690	2788
15	2279	2842	0	3495	3505	2872	2226	2373	1756
16	2946	3206	0	3139	3709	3247	2762	2160	2084
17	2568	2796	0	3335	2905	2999	2673	2478	2071
18	2902	3244	0	3499	3347	3058	3040	2882	2560
19	2574	2768	0	3617	3051	2837	2828	2451	2316
20	2449	3133	0	3590	3728	2955	2856	2573	2286
21	2746	2925	0	3829	3695	3025	2901	2640	2245
22	1903	2380	0	3504	3592	2398	2496	2186	1697
23	2662	2904	0	3608	3647	3087	2754	2511	2406
24	2427	2883	0	3403	3553	2883	2693	2331	1984
25	2718	2608	0	3801	3644	2470	2677	2530	2402
26	2873	3154	0	3548	2967	2706	3007	2488	2470
27	2824	3333	0	3314	3442	3239	2831	2485	2319
28	2419	3394	0	3548	3412	3473	3339	2923	2731
29	1268	2635	0	3646	3527	2558	2690	2494	1913
30	1954	2992	0	3606	3522	3044	3012	2502	2326
31	1939	3353	0	3478	3422	2593	2707	2558	2291
32	1943	2882	0	3694	3649	2768	2887	2648	2116
33	2198	3203	0	3773	3679	3068	2730	2542	2023

Tabel 5.27. Data volume kendaraan shift 1 dan 2 Februari dan Maret 2015 (lanjutan)

34	2579	3275	0	3549	3597	3174	2958	2509	2473
35	2596	3594	0	3485	3623	3331	3110	2945	2667
36	1637	2919	0	3413	3429	2740	3024	2458	2115
37	1896	2901	0	3765	3551	2849	3171	2803	2331
38	1839	3072	0	3644	3574	3120	2780	2323	2127
39	1788	2908	0	3892	3619	2622	2860	2718	2127
40	2015	2857	0	3583	3357	2909	2776	2655	2240
41	2326	2951	0	3506	3594	3404	2907	2608	2295
42	2460	3597	0	3255	3366	3332	3198	2738	2532
43	1365	2718	0	3464	3121	2573	2634	2624	1942
44	2367	3075	0	2566	3683	2505	2995	2865	2658
45	2157	3263	0	2709	3554	2876	2746	2663	2413
46	2173	3005	0	3370	3458	2625	3068	2507	2551
47	2489	3206	0	3577	3622	2726	2732	2523	2493
48	2760	3387	0	3656	3563	3372	3209	2928	2936
49	2806	3768	0	3498	3403	3563	3471	2653	2697
50	1593	2279	0	3584	3302	2222	2843	2508	2103
51	2422	3037	0	3598	3353	2899	2998	2890	2413
52	1984	3240	0	3742	3160	3203	2491	2390	2166
53	1984	2884	0	3584	3206	2894	2861	2598	2308
54	2153	2983	0	3513	3375	2933	3036	2926	2645
55	2290	2718	0	3738	3381	2457	2890	3150	2832
56	2390	3049	0	3601	3433	2973	3009	3045	2748
57	1147	2684	0	3629	3177	2813	2547	2373	1651
58	2424	2795	0	3802	3359	2471	3122	2578	2791
59	2961	3317	0	3530	3623	3514	2984	2719	2670

Pengujian kemudian dilakukan menggunakan SPSS menggunakan *Independent-Sampel t-Test* dengan taraf signifikansi 5% tingkat keyakinan 95%. Adapun hasil pengujian adalah sebagai ditunjukan pada gambar 5.47.

Independent Samples Test

		Levene's Test for Equality of Variances		t-test for Equality of Means						
									95% Confidence Interval of the Difference	
		F	Sig.	t	df	Sig. (2-tailed)	Mean Difference	Std. Error Difference	Lower	Upper
Gate2	Equal variances assumed	20,530	,000	-1,016	87	,312	-91,36384	89,93085	-270,11115	87,38347
	Equal variances not assumed			-1,296	83,606	,199	-91,36384	70,52118	-231,61259	48,88491
Gate4	Equal variances assumed	6,802	,011	-1,471	87	,145	-88,38814	60,09936	-207,84213	31,06586
	Equal variances not assumed			-1,765	86,621	,081	-88,38814	50,08087	-187,93544	11,15917
Gate8	Equal variances assumed	1,399	,240	1,103	87	,273	53,84011	48,83229	-43,21936	150,89958
	Equal variances not assumed			1,295	85,040	,199	53,84011	41,56011	-28,79196	136,47219
Gate10	Equal variances assumed	1,964	,165	,588	87	,558	26,55706	45,13247	-63,14861	116,26274
	Equal variances not assumed			,674	81,826	,502	26,55706	39,41318	-51,85082	104,96494
Gate12	Equal variances assumed	7,892	,006	-1,008	87	,316	-61,44407	60,95557	-182,59986	59,71172
	Equal variances not assumed			-1,208	86,562	,230	-61,44407	50,85483	-162,53076	39,64263
Gate14	Equal variances assumed	2,622	,109	-,307	87	,759	-14,99605	48,81627	-112,02366	82,03157
	Equal variances not assumed			-,343	77,618	,733	-14,99605	43,76080	-102,12386	72,13177
Gate16	Equal variances assumed	,897	,346	,765	87	,447	39,56271	51,73943	-63,27501	142,40043
	Equal variances not assumed			,841	75,147	,403	39,56271	47,03602	-54,13484	133,26026
Gate18	Equal variances assumed	7,222	,009	-,660	87	,511	-58,93672	89,23143	-236,29385	118,42040
	Equal variances not assumed			-,853	81,426	,396	-58,93672	69,08999	-196,39322	78,51977

Gambar 5.47. Hasil uji Independent Sample t-test SPSS

(Sumber: Hasil pengolahan data)

Pada gambar 5.47 terlihat bahwa nilai signifikansi (sig (2-tailed)) menunjukan nilai diatas 0.05 hal ini menunjukan penolakan H_0 pada seluruh pengujian, menunjukan bahwa tidak ada perbedaan rata-rata populasi antara hasil output model dengan data aktual, menandakan bahwa model simulasi yang dibuat adalah valid.