

BAB 2

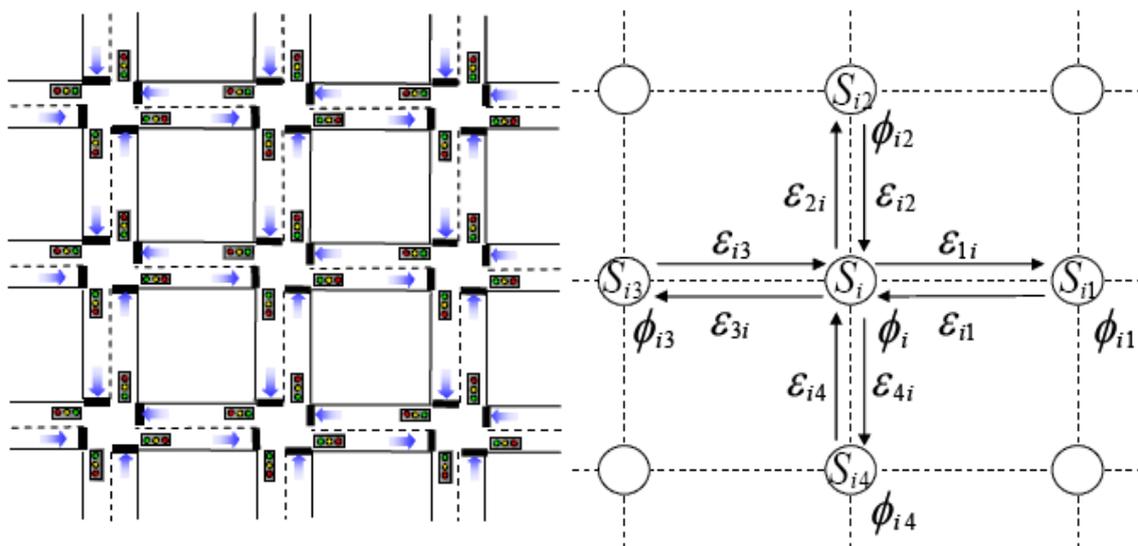
LANDASAN TEORI

Pada bab ini dijelaskan landasan teori dari metode dan perancangan kerja yang digunakan dalam tugas akhir. Pembahasan landasan teori akan menjelaskan konsep *swarm self-organizing map*, sinkronisasi Kuramoto, dan *Open Dynamics Engine* (ODE).

2.1 Swarm-self organizing map

Swarm "robot dan peralatan kontrol" merupakan bidang yang baru dengan fokus utama pada pengendalian sejumlah sistem robot atau mesin yang homogen. Sistem ini berfungsi untuk mengembangkan perilaku ditingkat makro yang merupakan hasil dari sinergi modul-modul sederhana yang berukuran kecil. *Swarm* "robot dan peralatan kontrol" memiliki beberapa keuntungan dibanding dengan robot dan peralatan kontrol individu dengan kemampuan yang sebanding.

Dari penjelasan sebelumnya dapat dianalogikan bahwa setiap persimpangan dalam sistem adalah sebuah robot dan peralatan sistem, dari analogi tersebut *swarm-self organizing map* akan digunakan untuk mengatur lalu lintas secara terdistribusi. Dalam penelitian ini setiap persimpangan memiliki hubungan dengan persimpangan lainnya, untuk itu akan diterapkan *swarm-self organizing map* dalam pengaturan lalu lintas secara terdistribusi, seperti yang terlihat pada gambar berikut ini:



Gambar 2.1 Model matematika

2.2 Sinkronisasi kuramoto

Sinkronisasi adalah sebuah metode atau usaha untuk membuat dua atau lebih objek menjadi sama dalam parameter tertentu. Sinkronisasi bisa terjadi dalam bentuk apapun, beberapa contoh sinkronisasi antara lain kunang-kunang yang melakukan sinkronisasi agar mereka menyala bersamaan atau menyala dengan suatu pola yang sama, tepukan tangan yang dilakukan banyak orang dimana setiap orang berusaha untuk melakukan sinkronisasi agar tercipta suara tepukan tangan yang sama.

Dalam melakukan sinkronisasi dibutuhkan waktu untuk mencapai sinkronisasi yang diinginkan. Para ilmuwan telah banyak meneliti tentang fenomena sinkronisasi dan berusaha memodelkan hal tersebut. Pendekatan yang paling sukses sampai saat ini adalah Model Kuramoto. Pendekatan ini menganalisa sebuah model dari fase *oscillator* yang berjalan dengan frekuensi sembarang dan dilakukan proses *coupling* melewati fungsi *sinus* dari perbedaan fasenya.

Model kuramoto cukup sederhana untuk dianalisa secara matematika, tetapi juga bisa menjadi cukup kompleks untuk dianalisa menjadi model yang bersifat *non trivial*. Pemodelan tersebut cukup baik untuk menampilkan berbagai pola sinkronisasi dan juga fleksibel untuk diterapkan ke permasalahan lainnya.

$$\dot{\theta}_i(t) = \omega_i + \sum_{j=1}^N K_{ij} \sin(\theta_j - \theta_i)$$

Rumus tersebut merupakan persamaan dasar yang digunakan dalam pemodelan sinkronisasi. Setiap *oscillator* pada sistem ini akan di-*phase lock* dengan setiap *oscillator* lain menggunakan fungsi *sinus* dari selisih fase *oscillator*.

K menyatakan konstanta *coupling* yang menunjukkan seberapa kuat *coupling* antara kedua *oscillator*. Sementara ω_i menyatakan frekuensi alamiah dari *oscillator* yang merupakan frekuensi dasar sebelum dipengaruhi oleh *coupling* antar *oscillator*.

2.3 Cycle time, green split, dan offset

Dalam pengaturan lampu lalu lintas terdapat tiga komponen penting yang mempengaruhi keefektifan suatu lampu lalu lintas:

- ***Cycle Time***

Cycle time adalah lamanya satu periode dari siklus lampu lalu lintas. Satu siklus lampu lalu lintas terdiri dari lama lampu merah, lampu kuning, dan lampu hijau.

- ***Green Split***

Lamanya durasi lampu hijau dalam satu periode lampu lalu lintas.

- ***Offset***

Besar waktu perbedaan relatif nyala lampu hijau antara suatu persimpangan dengan persimpangan lainnya.

2.4 Open Dynamics Engine (ODE)

Pengembangan simulasi diperlukan sebelum eksperimen atau pengimplementasian langsung di lapangan karena apabila percobaan dan pengimplementasian langsung dilapangan mengalami kegagalan, akan sangat memakan biaya yang besar. Untuk itu pengembangan simulasi sangatlah penting serta dibutuhkan suatu *platform* simulasi yang merepresentasikan sesuai atau semirip mungkin dengan keadaan dunia nyata. Untuk memenuhi kebutuhan tersebut, pengembangan simulasi ini menggunakan *physics engine Open Dynamics Engine (ODE)*.

Open Dynamics Engine (ODE) adalah *library* untuk mensimulasikan dinamika obyek 3 dimensi. *Open Dynamics Engine (ODE)* memiliki fitur-fitur seperti *joint*, *collision detection*, *error reduction parameter*, dan lain-lain. Dunia dalam *Open Dynamics Engine (ODE)* merepresentasikan dunia nyata dan dapat memiliki beberapa *space* sekaligus. *Open Dynamics Engine (ODE)* sudah banyak digunakan pada banyak aplikasi simulasi dan aplikasi permainan, sehingga sudah tidak diragukan lagi kemampuannya sebagai sebuah *platform* simulasi. Implementasi ODE yang digunakan dalam pengembangan simulasi ini adalah ODE 0.10.1.