



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk mengubah, memperbaiki, dan membuat ciptaan turunan bukan untuk kepentingan komersial, selama anda mencantumkan nama penulis dan melisensikan ciptaan turunan dengan syarat yang serupa dengan ciptaan asli.

Copyright and reuse:

This license lets you remix, tweak, and build upon work non-commercially, as long as you credit the origin creator and license it on your new creations under the identical terms.

BAB III

METODOLOGI PENELITIAN DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam penelitian ini adalah sebagai berikut.

1. Analisis Kebutuhan

Analisis awal kebutuhan apa saja yang diperlukan untuk mengimplementasikan NEAT pada *karting microgame*. Menentukan *software* dan standard lingkungan yang dibutuhkan untuk penelitian.

2. Telaah Literatur

Memahami dan mempelajari literatur, teori, dan implementasi lain yang memiliki relevansi terhadap perancangan dan pembuatan sistem, seperti game dan simulasi, Unity Karting Microgame dan *self-driving kart*, algoritma genetik, jaringan saraf tiruan, dan *Neuroevolution of Augmenting Topologies*.

3. Perancangan Sistem

Membuat rancangan sistem berdasarkan informasi yang telah didapat dari telaah literatur. Membuat rancangan *use case diagram* untuk simulasi, flowchart untuk implementasi metode NEAT, dan rancangan antarmuka pengguna.

4. Pemrograman dan Implementasi

Mengimplementasikan hasil rancangan yang telah disusun pada tahap sebelumnya dalam bahasa pemrograman C# menggunakan Unity Engine dan

Visual Studio 2019. Selama pemrograman dan implementasi dilakukan pengujian berkala untuk setiap perkembangan untuk memastikan program berjalan sesuai dengan rancangan.

5. Pengujian dan Evaluasi

Pengujian dilakukan dengan melihat keberhasilan individu yang dihasilkan berdasarkan generasi kelahiran, nilai *fitness*, dan waktu yang dibutuhkan untuk menyelesaikan lintasan. Evaluasi dilakukan dengan menguji individu pada lintasan yang berbeda.

6. Penulisan Karya Ilmiah

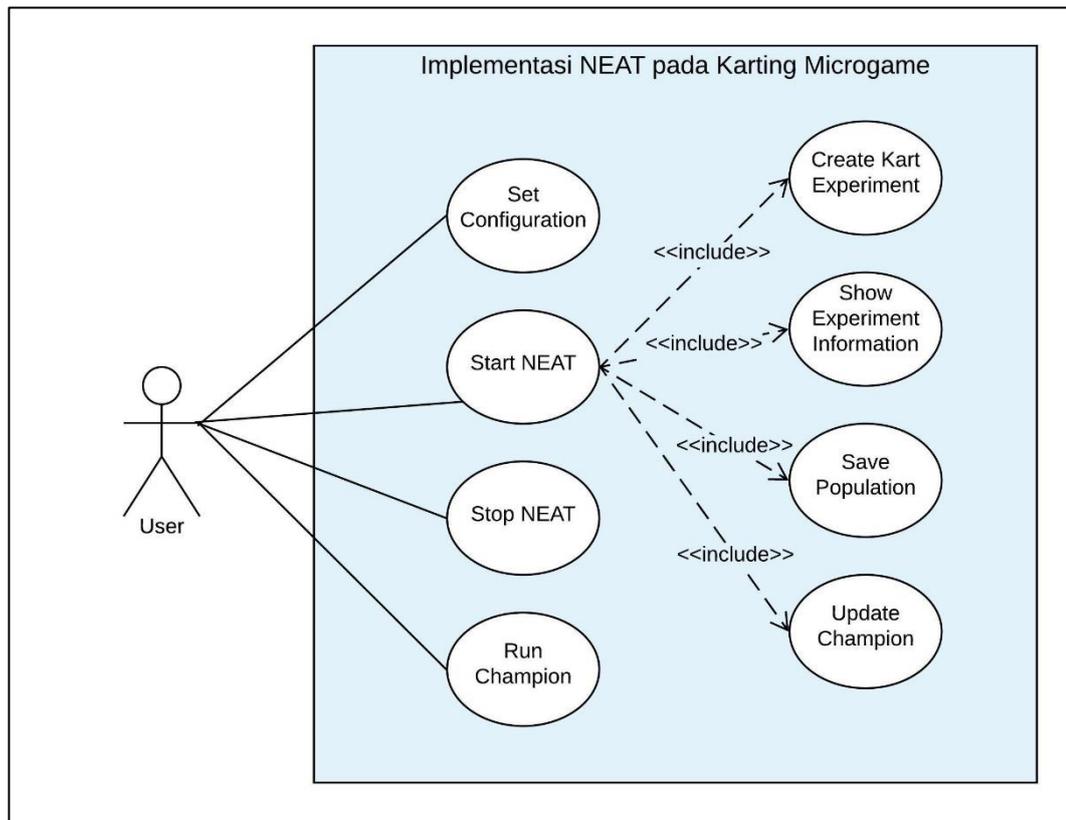
Penulisan karya ilmiah sebagai bentuk dokumentasi penelitian mulai dari awal hingga penulisan kesimpulan dan saran sehingga penelitian ini dapat menjadi sarana ilmu pengetahuan dan referensi untuk penelitian selanjutnya.

3.2 Perancangan Sistem

Sistem dirancang menggunakan *use case diagram*, flowchart, dan rancangan antarmuka pengguna.

3.2.1 Use Case Diagram

Use Case Diagram digunakan untuk mengetahui gambaran umum sistem yang akan dirancang beserta proses utama yang akan dijalankan pada *karting microgame*. Gambar 3.1 merupakan *use case diagram* dari implementasi NEAT pada *karting microgame*. Terdapat empat fungsi utama dalam *use case diagram*, yaitu Set Configuration, Start NEAT, Stop NEAT, dan Run Champion.

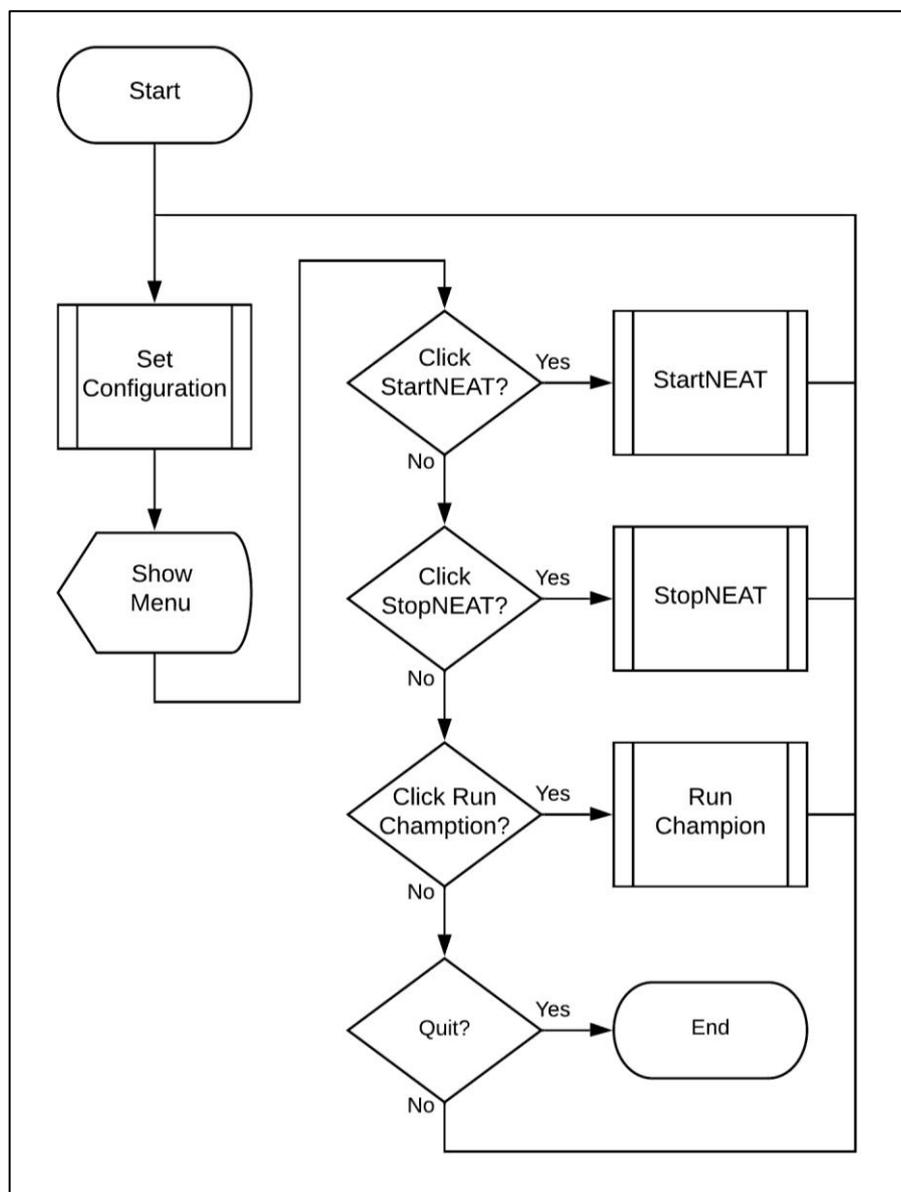


Gambar 3.1 *Use Case Diagram* Implementasi NEAT pada *Karting Microgame*

Set Configuration merupakan proses dimana pengguna menentukan konfigurasi sistem seperti jumlah populasi dan spesies. Start NEAT merupakan proses utama pada sistem yang dapat memanggil proses Create Kart Experiment untuk membuat simulasi dengan NEAT berdasarkan konfigurasi yang diberikan, Show Experiment Information untuk menampilkan generasi dan jumlah populasi yang masih berjalan, Save Population untuk menyimpan populasi, dan Save Champion untuk menyimpan lima individu terbaik dari simulasi. Stop NEAT untuk menghentikan sementara simulasi. Run Champion untuk menampilkan ulang lima individu terbaik sejak simulasi dimulai.

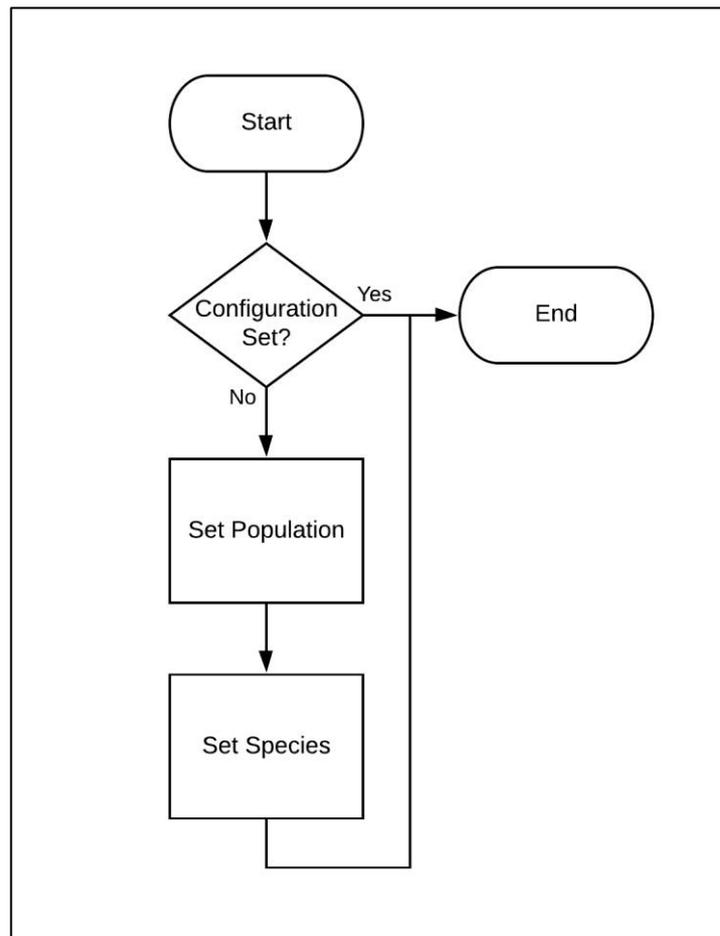
3.2.2 Flowchart

Alur kerja sistem dirancang menggunakan flowchat agar urutan langkah proses dapat lebih mudah dipahami. Berdasarkan *use case diagram*, terdapat empat proses utama, yaitu Set Configuration, Start NEAT, Stop NEAT, dan Run Champion. Flowchart umum implementasi NEAT pada *karting microgame* ditunjukkan pada Gambar 3.2.



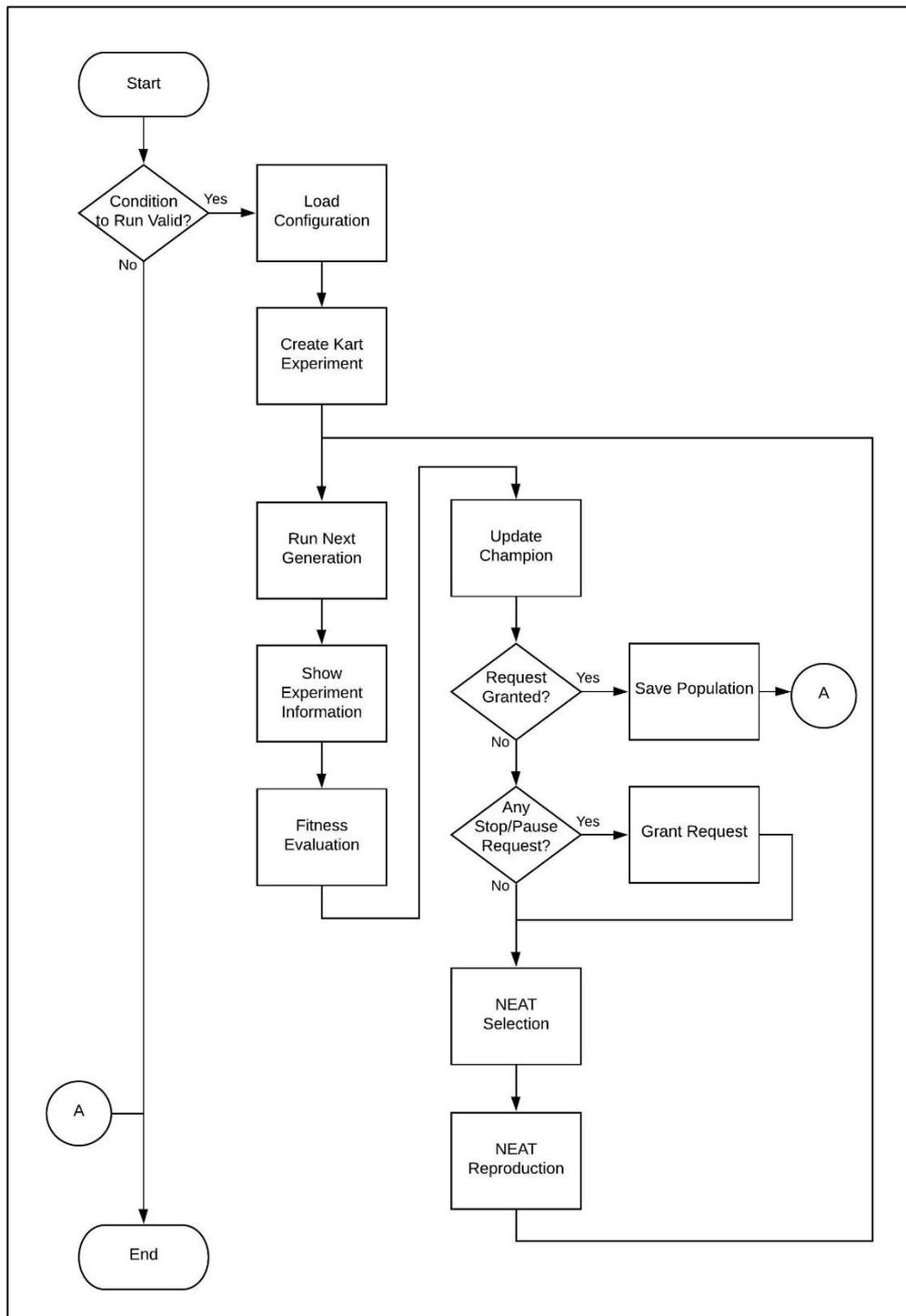
Gambar 3.2 Flowchart Umum Implementasi NEAT pada *Karting Microgame*

Gambar 3.3 menunjukkan flowchart Set Configuration. Sebelum memulai simulasi, pengguna dapat menentukan jumlah populasi dan spesies yang akan digunakan sebelum menjalankan *karting microgame*.



Gambar 3.3 Flowchart Set Configuration

Gambar 3.4 menunjukkan flowchart Start NEAT. Proses ini berfungsi untuk menjalankan simulasi NEAT. Apabila kondisi untuk menjalankan simulasi valid (tidak ada simulasi lain yang sedang berjalan), maka pertama sistem akan memuat konfigurasi, lalu membuat simulasi dengan NEAT.

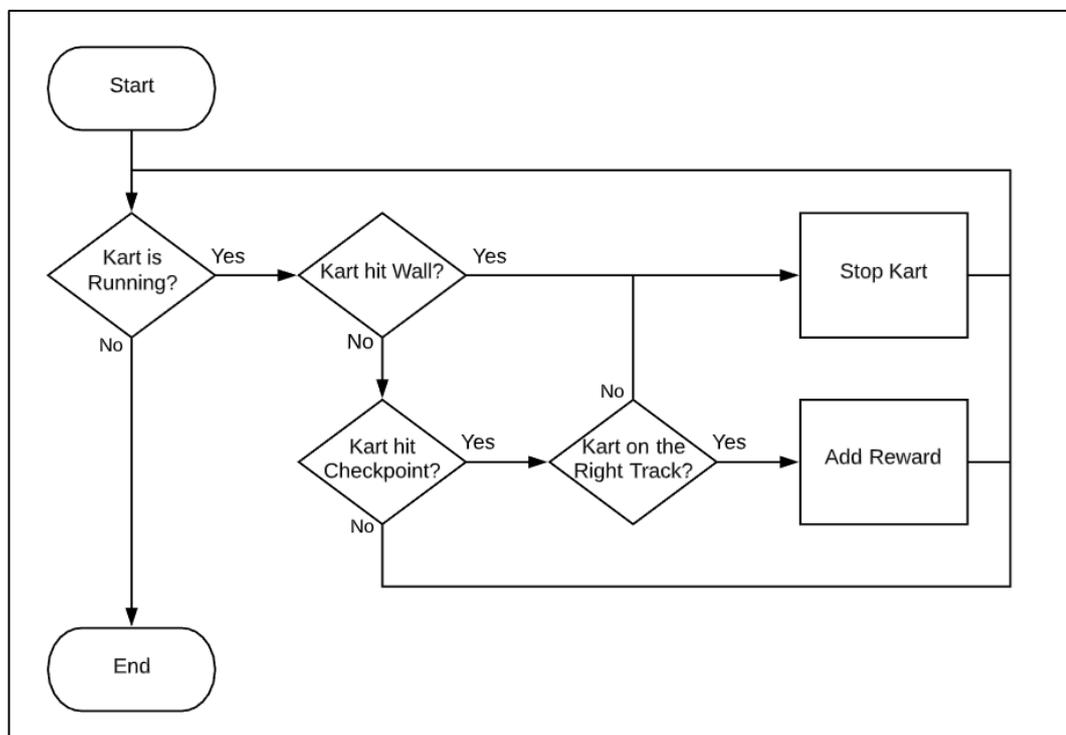


Gambar 3.4 Flowchart Start NEAT

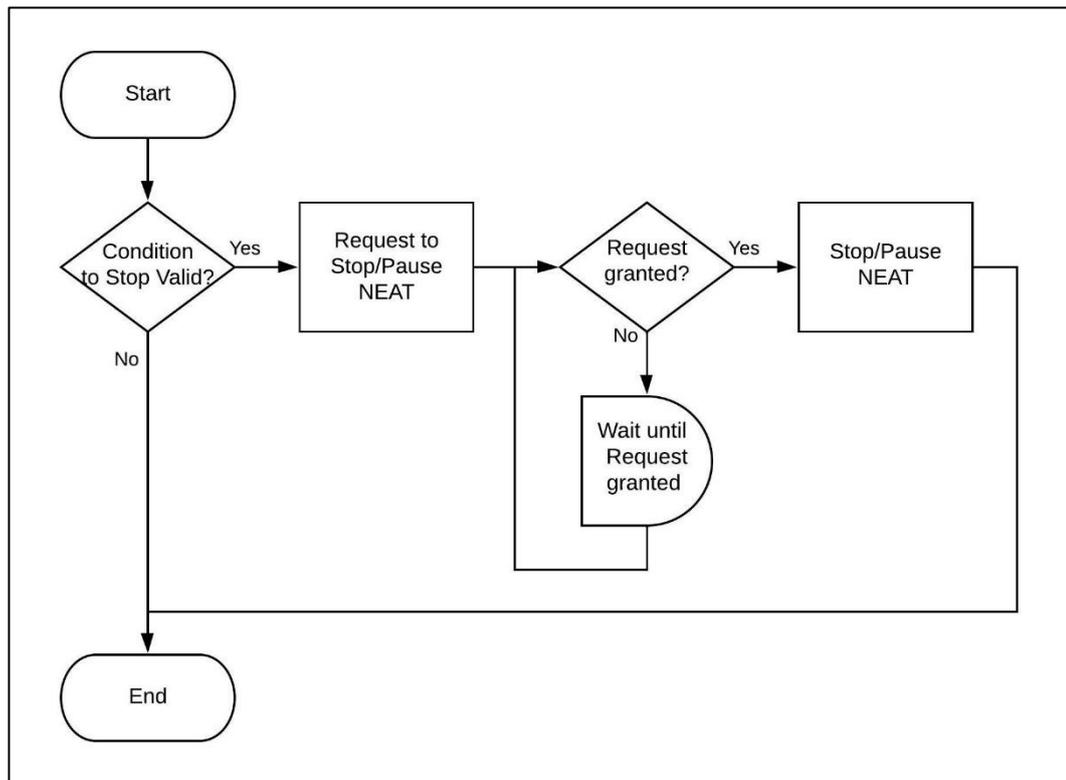
Populasi awal dibuat ketika proses Create Kart Experiment dijalankan. Populasi awal ini akan menjadi generasi selanjutnya yang dijalankan, saat populasi

dijalankan semua individu yang ada di dalam populasi tersebut akan dievaluasi nilai *fitness*-nya. Informasi mengenai populasi yang berjalan akan ditampilkan seiring berjalannya evaluasi nilai *fitness*. Setelah satu generasi selesai dievaluasi nilai *fitness*-nya, champion akan diupdate. Apabila tidak ada permintaan untuk menghentikan sementara simulasi, maka *neuroevolution* oleh NEAT dilakukan (seleksi dan reproduksi). Apabila terdapat permintaan untuk menghentikan sementara dan simulasi sedang berjalan, maka proses akan menunggu hingga populasi terakhir selesai dijalankan, menyimpan populasi tersebut beserta hasil evaluasi *fitness*-nya, lalu mengakhiri proses.

Gambar 3.5 menunjukkan Flowchart Evaluasi Fitness yang dijalankan selama *kart* berada dalam simulasi. Ketika suatu populasi dijalankan, setiap individu dalam populasi akan dievaluasi *fitness*-nya berdasarkan Flowchart Evaluasi Fitness.

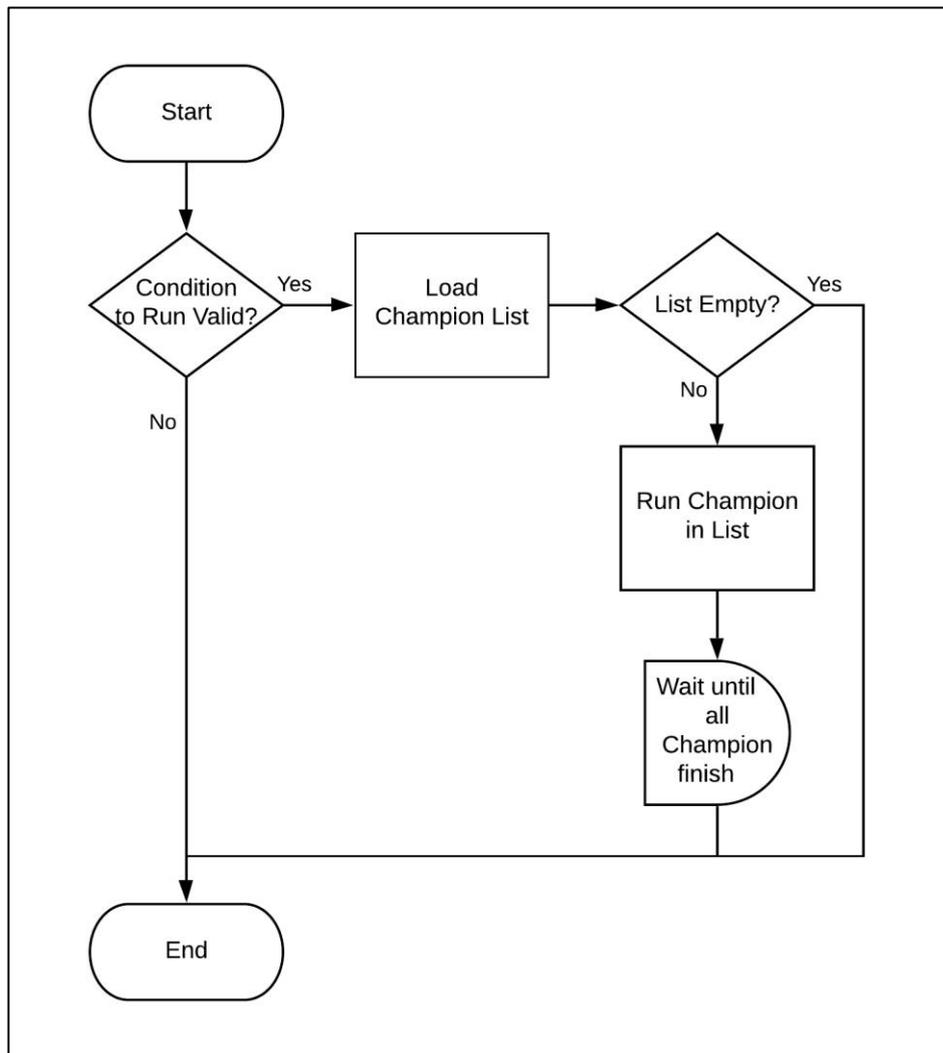


Gambar 3.5 Flowchart Evaluasi Fitness



Gambar 3.6 Flowchart Stop NEAT

Gambar 3.6 menunjukkan flowchart Stop NEAT. Proses ini berfungsi untuk menghentikan sementara simulasi. Apabila kondisi untuk berhenti valid, maka permintaan untuk menghentikan sementara simulasi akan dikirim ke simulasi yang sedang berjalan. Apabila permintaan disetujui, simulasi akan dihentikan sementara. Apabila permintaan belum disetujui, maka sistem akan menunggu hingga diberikan izin untuk menghentikan simulasi sementara.



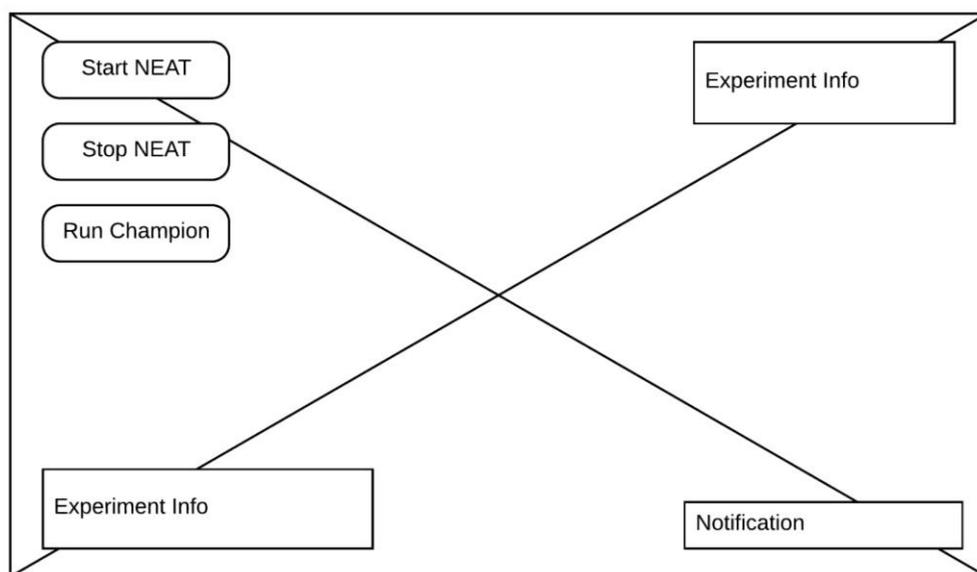
Gambar 3.7 Flowchart Run Champion

Gambar 3.7 menunjukkan flowchart Run Champion. Proses ini berfungsi untuk menampilkan ulang lima individu terbaik sejak simulasi dimulai (champion). Apabila kondisi untuk menjalankan champion valid (tidak ada simulasi lain yang sedang berjalan), maka sistem akan memuat champion list. Apabila list kosong, tidak ada champion yang dapat dijalankan. Apabila list tidak kosong, setiap champion yang ada dalam list akan dijalankan bersamaan sehingga pengguna dapat

melihat dan membandingkan champion. Proses selesai setelah menunggu seluruh champion dalam list selesai dijalankan.

3.2.3 Rancangan Antarmuka Pengguna

Tampilan dirancang menggunakan rancangan antarmuka pengguna untuk memberikan gambaran dari program simulasi yang dihasilkan. Program hanya memiliki satu halaman tampilan. Gambar 3.8 menunjukkan rancangan antarmuka pengguna untuk halaman tampilan tersebut.



Gambar 3.8 Rancangan Antarmuka Pengguna

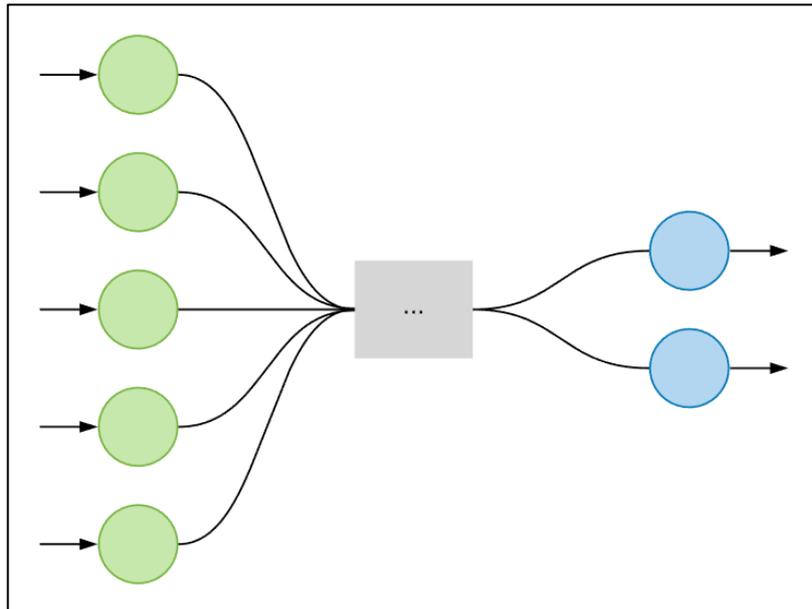
Berdasarkan *use case diagram*, terdapat empat proses utama, yaitu Set Configuration, Start NEAT, Stop NEAT, dan Run Champion. Set Configuration dilakukan di luar program, sehingga tidak di atur oleh antarmuka pengguna. Start NEAT dilakukan sesuai dengan Flowchart Start NEAT ketika button Start NEAT diklik. Stop NEAT dilakukan sesuai dengan Flowchart Stop NEAT ketika button

Stop NEAT diklik. Run Champion dilakukan sesuai dengan Flowchart Run Champion ketika button Run Champion diklik. Experiment Information terdapat pada kanan atas dan kiri bawah layar untuk memberikan informasi terkait populasi yang sedang dijalankan. Notifikasi akan diberikan pada kanan bawah layar untuk menampilkan notifikasi terkait simulasi. Gambar pada latar belakang merupakan tampilan peta simulasi.

3.2.4 Rancangan Simulasi

Simulasi dirancang berdasarkan Unity Karting Microgame yang merupakan *3d racing game* dari Unity dengan komponen utama lintasan dan *kart*. Lintasan pada simulasi dirancang untuk memiliki *checkpoint*. *Checkpoint* adalah titik-titik pengecekan pada lintasan yang berfungsi untuk membantu evaluasi nilai *fitness* berdasarkan *reward system* yang digunakan. Berdasarkan implementasi Unity ML-Agents Toolkit pada Unity Karting Microgame dan penelitian oleh Jallov (2019), *reward system* akan menambah nilai *fitness* apabila *kart* melewati *checkpoint* dengan urutan yang sesuai dengan memperhitungkan kecepatan kart saat melewati *checkpoint* tersebut. Sehingga seperti yang dijelaskan dalam landasan teori, *checkpoint* berpengaruh untuk menentukan nilai *fitness*.

Dalam penelitian ini digunakan dua buah lintasan. Lintasan 1 dirancang berdasarkan *training track* pada Unity Karting Microgame dan digunakan sebagai lintasan untuk melatih jaringan saraf tiruan. Lintasan 2 dirancang dengan menggunakan model *prefab* yang sudah disediakan oleh Unity Karting Microgame dan digunakan sebagai lintasan untuk menguji champion atau individu terbaik.



Gambar 3.9 Rancangan Topologi pada *Self-Driving Kart*

Self-driving kart dirancang berdasarkan *kart* dalam Unity Karting Microgame. Rancangan topologi jaringan saraf tiruan untuk *self-driving kart* ditunjukkan pada Gambar 3.9. Pada Unity Karting Microgame, *kart* dikendalikan oleh pengguna menggunakan *arrow keys* dengan dua jenis pergerakan, yaitu *steering* dan *acceleration*. Sehingga dalam rancangan topologi jaringan saraf tiruan untuk *self-driving kart* diperlukan dua *output nodes* yang masing-masing akan mengembalikan nilai untuk mengatur *steering* dan *acceleration* dan menggantikan nilai yang seharusnya dimasukkan oleh pengguna. Kart diharapkan dapat mengambil tindakan terbaik berdasarkan situasi yang ada dengan mengamati lingkungan melalui sensor. Berdasarkan penelitian oleh Jallo (2019) dan Immanuel (2019), individu yang digunakan dirancang dengan memiliki 5 buah *input sensor*. *Input sensor* diartikan sebagai alat pada individu untuk mengamati lingkungan, dalam simulasi ini *input sensor* digunakan untuk membaca jarak dari

kart ke tepi lintasan. Masing-masing *input sensor* memiliki fungsi yang sama, yaitu membaca jarak dari *kart* ke tepi lintasan namun dengan arah derajat yang berbeda. Mengikuti penelitian sebelumnya oleh Jallov (2019) dan Immanuel (2019), *input sensor* yang digunakan mengarah ke 0° , 45° , -45° , 90° , dan -90° dengan acuan muka *kart* sebagai 0° . Sehingga seperti pada Gambar 3.8, rancangan topologi memiliki 5 buah *input nodes*, 2 buah *output nodes*, dan *hidden nodes* serta *connections* yang akan terdefinisi seiring terjadi *neuroevolution* dalam simulasi. Evaluasi *fitness* pada *self-driving kart* akan mengikuti *reward system* yang dijelaskan pada landasan teori dengan nilai *fitness* yang didapatkan berdasarkan Rumus 2.1 dan cara hitung kompleksitas dari *self-driving kart* yang didapatkan berdasarkan Rumus 2.2.