



Hak cipta dan penggunaan kembali:

Lisensi ini mengizinkan setiap orang untuk menggubah, 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 DAN PERANCANGAN SISTEM

3.1 Metodologi Penelitian

Metodologi penelitian yang digunakan dalam implementasi *evolutionary neural network* pada simulasi *autonomous mobile robot navigation* meliputi beberapa tahap penelitian antara lain.

1. Studi Literatur

Studi literatur dilakukan sebagai pembelajaran terhadap berbagai teori yang memiliki relevansi terhadap perancangan dan pembuatan sistem. Teori-teori yang dimaksud, antara lain simulasi, *autonomous robot*, *neural network*, *genetic algorithm*, dan *evolutionary neural network*.

2. Perancangan Sistem

Tahap perancangan sistem dimulai dengan perancangan *use case diagram* untuk memperlihatkan gambaran umum kerja dari sistem. Kemudian, dilanjutkan dengan perancangan *class diagram* yang berguna untuk menunjukkan hubungan antar *class* yang terdapat pada sistem dan memperjelas *dependencies* antar *class*.

Lalu, perancangan *sequence diagram* juga dibuat. *Sequence diagram* berfungsi untuk menggambarkan kolaborasi dinamis antara sejumlah objek dan menunjukkan rangkaian pesan yang dikirim antara object juga interaksi antara objek. Selain itu, perancangan desain antarmuka dibuat untuk memberikan gambaran visualisasi tampilan sistem. Perancangan desain antarmuka dibuat berdasarkan tampilan dari simulasi.

3. Pembuatan Sistem

Pembuatan sistem implementasi *evolutionary neural network* pada simulasi *autonomous mobile robot navigation* ini menggunakan C# dan platform pembuatan simulasi menggunakan Unity dan Visual Studio 2017. Seluruh data akan disimpan pada file berbentuk *notepad*.

4. Pengujian Sistem

Pengujian sistem dilakukan setelah pembuatan sistem selesai dengan menilai apakah *robot* berhasil mencapai titik tujuan dari sumber yang diberikan. Pengujian sistem ini akan mencatat nilai rata-rata keberhasilan *robot* dari jumlah *robot* yang mencapai titik tujuan dengan jumlah percobaan yang telah ditentukan.

5. Penulisan laporan

Penulisan laporan mencakup seluruh penelitian dari awal pembuatan simulasi hingga hasil dan kesimpulan dari penelitian yang telah dilakukan. Penarikan kesimpulan merupakan ringkasan dari hasil pengolahan data dan analisis yang telah dilakukan.

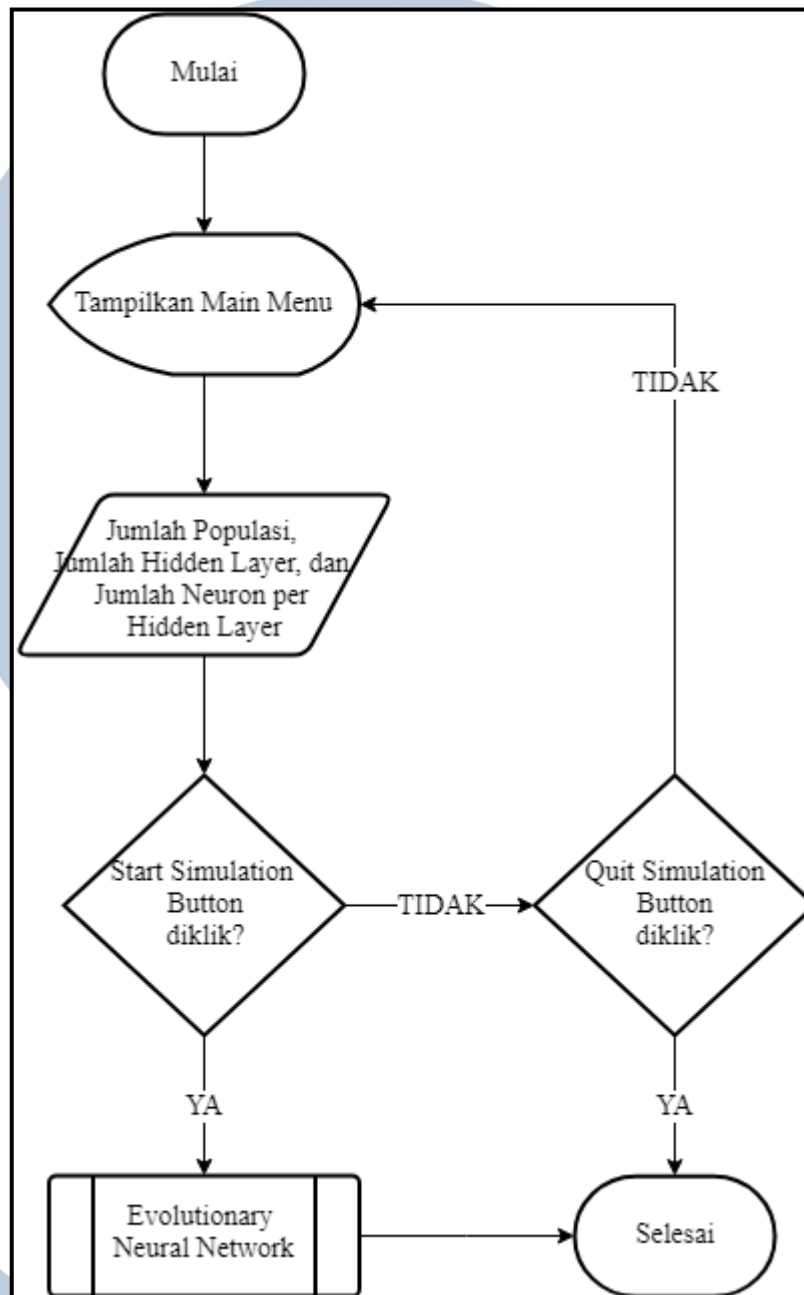
3.2 Perancangan Sistem

Sistem dirancang dengan menggunakan *Flowchart*, *Use Case Diagram*, *Sequence Diagram*, *Class Diagram*, dan Desain antarmuka.

A. Flowchart

Flowchart adalah representasi bergambar yang menggambarkan alur langkah dalam suatu program, orang dalam suatu organisasi, atau halaman dalam presentasi. Berikut merupakan flowchart simulasi *autonomous mobile robot navigation*.

A.1 Flowchart Main Menu

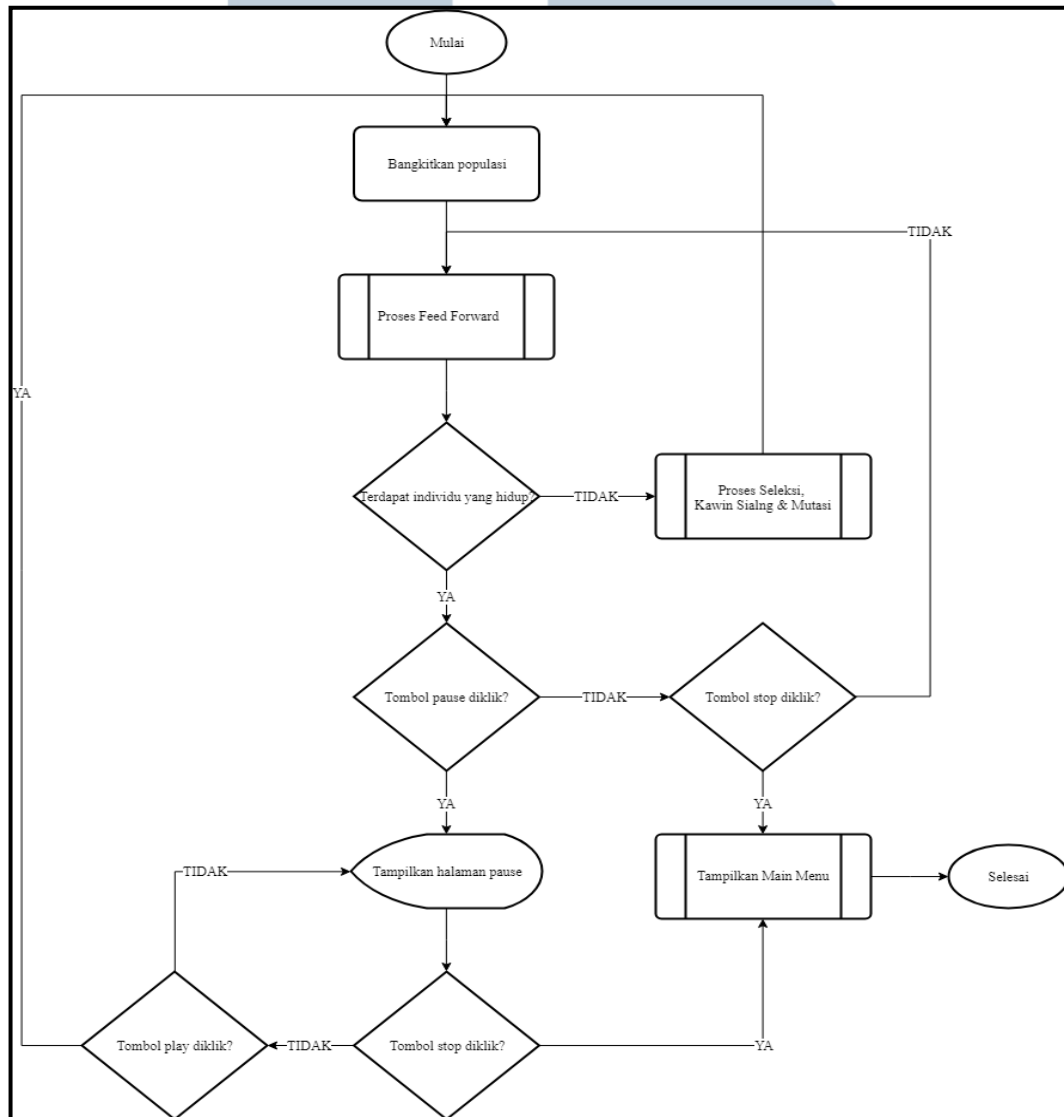


Gambar 3.1 Flowchart Main Menu

Gambar 3.1 merupakan Flowchart *Main Menu* pada simulasi *autonomous mobile robot navigation*. Pada bagian main menu, *user* dapat melakukan interaksi yaitu memasukkan *input* berupa jumlah populasi, jumlah *hidden layer*, dan jumlah *neuron* per *layer*. Ketika tombol *start simulation* diklik oleh *user*, simulasi akan

dimulai dan proses *evolutionary neural network* akan dijalankan. Ketika tombol *quit simulation* diklik *user* akan keluar dari simulasi.

A.2 Flowchart Proses Evolutionary Neural Network

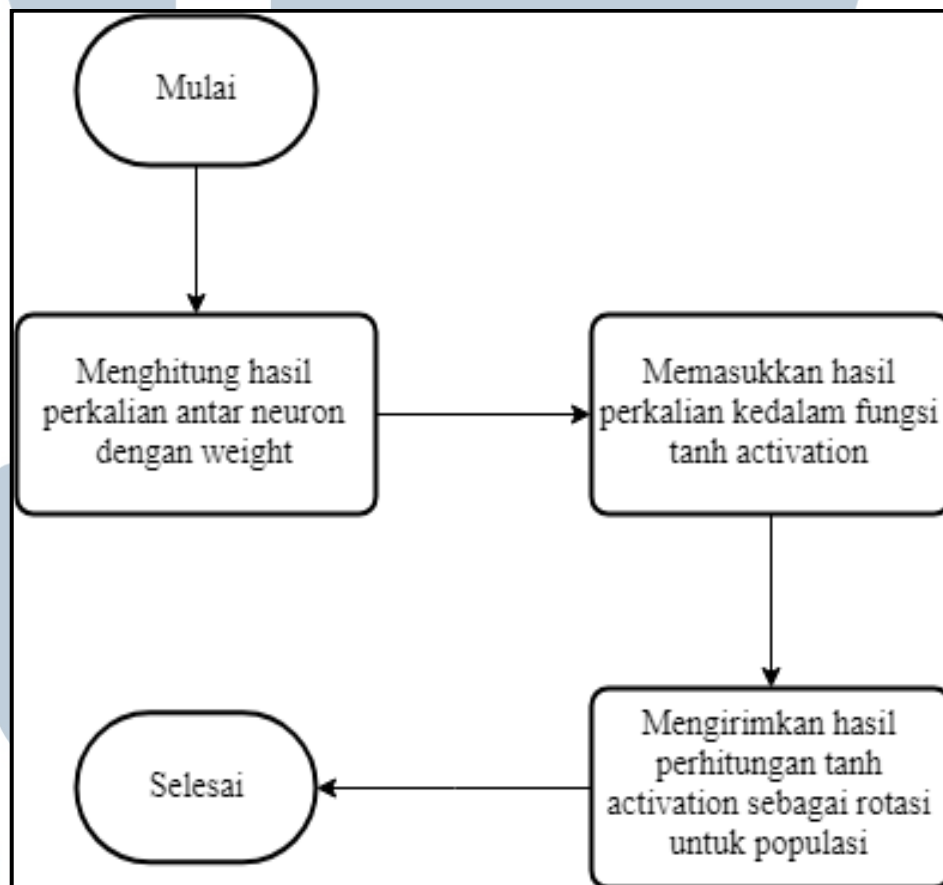


Gambar 3.2 Flowchart proses *Evolutionary Neural Network*

Gambar 3.2 merupakan Flowchart proses *Evolutionary Neural network*. Proses pertama pada *evolutionary neural network* adalah membangkitkan populasi berdasarkan dari *input* yang diberikan oleh *user*, setiap individu dalam populasi tersebut akan melakukan proses *feed forward*. Ketika seluruh individu di dalam populasi tidak ada yang bertahan maka proses *selection*, *crossover*, dan *mutation*

akan dijalankan. Setelah proses *selection*, *crossover*, dan *mutation* dijalankan, populasi yang baru akan dihasilkan berdasarkan dari individu yang dihasilkan pada proses *selection*, *crossover*, dan *mutation*. User dapat melakukan dua interaksi selama proses *evolutionary neural network* dijalankan, yaitu menekan tombol *pause* atau tombol *stop*. Ketika tombol *pause* diklik, simulasi akan dihentikan sementara dan memunculkan tampilan *pause*. Ketika tombol *stop* diklik, simulasi akan dihentikan dan dikembalikan ke tampilan *main menu*.

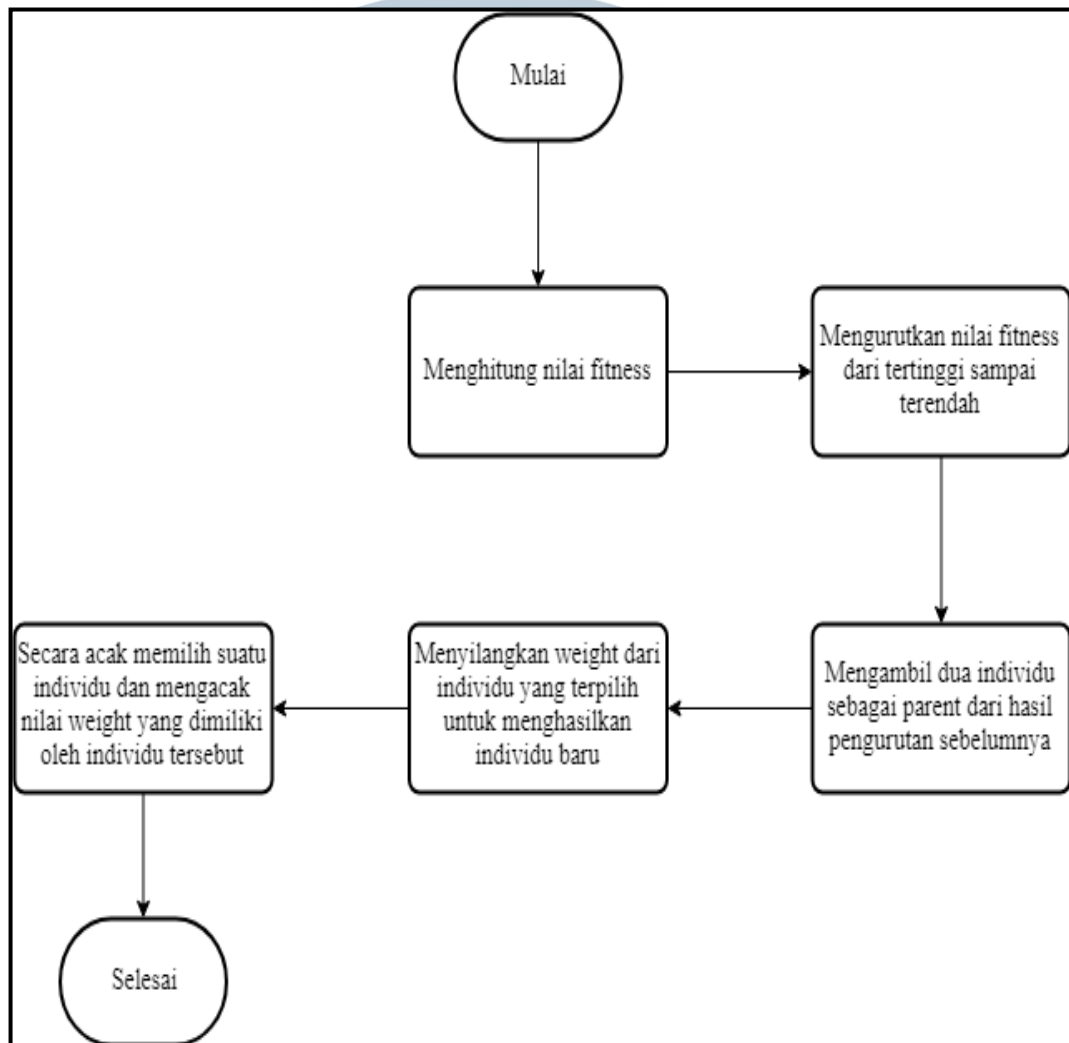
A.3 Flowchart Proses Feed Forward



Gambar 3.3 Flowchart Proses *Feed Forward*

Gambar 3.3 merupakan gambar Flowchart Proses *Feed Forward*. Rumus perhitungan *feed forward* dapat dilihat pada Rumus 2.3.

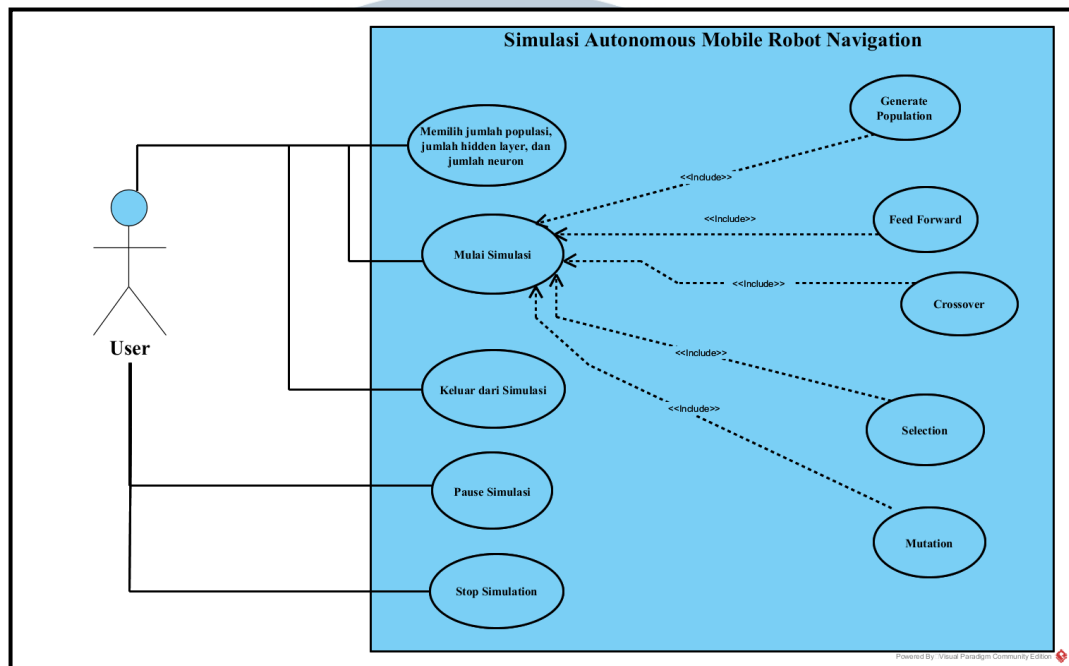
A.4 Flowchart Proses Selection, Crossover, dan Mutation



Gambar 3.4 Flowchart Proses *Selection*, *Crossover*, dan *Mutation*.

Proses *selection* dilakukan dengan cara mengurutkan individu berdasarkan nilai *fitness* tertinggi sampai terendah. Perhitungan nilai *fitness* dapat dilihat pada Rumus 2.1. Setelah diurutkan akan dipilih 2 individu secara acak berdasarkan hasil urutan tersebut. Dua individu yang terpilih akan digunakan untuk proses *crossover*. Setelah dilakukan proses *crossover*, proses selanjutnya yang akan dilakukan adalah proses *mutation*. Pada proses *mutation* akan dipilih individu secara acak dari hasil *crossover* sebelumnya. Flowchart proses *selection*, *crossover*, dan *mutation* dapat dilihat pada Gambar 3.4.

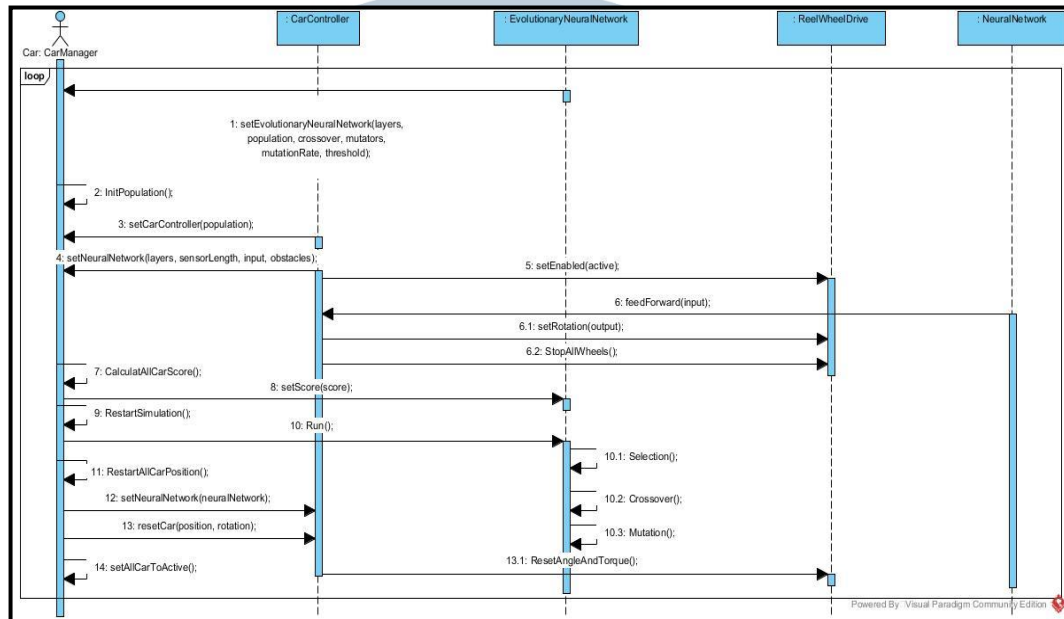
B. Use Case Diagram



Gambar 3.5 Use Case Diagram

Gambar 3.5 merupakan *Use Case Diagram* pada simulasi *autonomous mobile robot navigation*. *Use case diagram* pada simulasi *autonomous mobile robot navigation* mempunyai 5 proses utama, yaitu memilih jumlah populasi, jumlah *hidden layer*, dan jumlah *neuron* per *layer*. Selain itu keempat proses lainnya meliputi, mulai simulasi, keluar dari simulasi, *pause* simulasi, dan *stop* simulasi. Pada proses pemilihan jumlah populasi, jumlah *hidden layer* dan jumlah *neuron*, hasil pemilihan tersebut akan digunakan sebagai *input* untuk simulasi berjalan. Ketika *user* memulai simulasi, terdapat 5 proses yang akan ikut berjalan ketika simulasi dimulai.. Lima proses tersebut yaitu, Proses *Generate Population*, Proses *Feed Forward*, Proses *Crossover*, Proses *Selection*, dan Proses *Mutation*. Proses keluar simulasi akan menutup program simulasi, proses *stop* simulasi akan memberhentikan simulasi yang telah berlangsung, proses *pause* akan memberhentikan sementara simulasi yang telah berlangsung.

C. Sequence Diagram



Gambar 3.6 Sequence Diagram

Gambar 3.2 merupakan *Sequence Diagram* dari simulasi *autonomous mobile robot navigation*. Terdapat 5 kelas di dalam *sequence diagram* tersebut, yaitu CarManager, CarController, EvolutionaryNeuralNetwork, ReelWheelDrive, dan NeuralNetwork. Selain menjadi *class*, CarManager juga berfungsi sebagai aktor. Terdapat 20 proses di dalam *sequence diagram* tersebut, antara lain proses setEvolutionaryNeuralNetwork, proses InitPopulation, proses setCarController, proses setNeuralNetwork, proses setEnabled, proses feedForward, proses setRotation, proses stopAllWheels, proses CalculateAllCarScore, proses setScore, proses restartSimulation, proses Run, proses Selection, proses Crossover, proses Mutation, proses RestartAllCarPosition, proses setNeuralNetwork, proses resetCar, proses resetAngleAndTorque, dan proses setAllCarToActive.

Proses setEvolutionaryNeuralNetwork adalah sebuah proses yang memberikan nilai awal kepada *evolutionary neural network* untuk dapat digunakan selanjutnya. Proses setEvolutionaryNeuralNetwork terjadi ketika simulasi pertama

kali dijalankan. Selanjutnya, proses InitPopulation adalah proses yang melakukan generalisasi kandidat sesuai dengan jumlah populasi yang telah diberikan. Proses setCarController merupakan sebuah proses yang memberikan sebuah kontrol terhadap *robot* yang telah digeneralisasi sebelumnya.

Proses setEnabled merupakan proses yang melakukan pengaturan terhadap *robot* di dalam simulasi untuk membuat setiap *robot* menjadi aktif. Aktif disini memiliki arti *robot* tersebut mulai berjalan dari titik sumber menuju ke titik tujuan. Proses feedForward merupakan proses pembelajaran di dalam sistem simulasi *autonomous mobile robot navigation*, dimana dari masukkan yang diberikan sistem dapat menghasilkan nilai rotasi dari roda *robot*. Setelah mendapatkan nilai rotasi dari proses pembelajaran sebelumnya, selanjutnya proses setRotation akan berjalan. Proses setRotation adalah proses yang memasukkan nilai hasil pembelajaran pada proses feedForward ke dalam nilai rotasi *robot*. Nilai rotasi yang dimasukkan berfungsi untuk *robot* dapat menghindari rintangan yang diberikan.

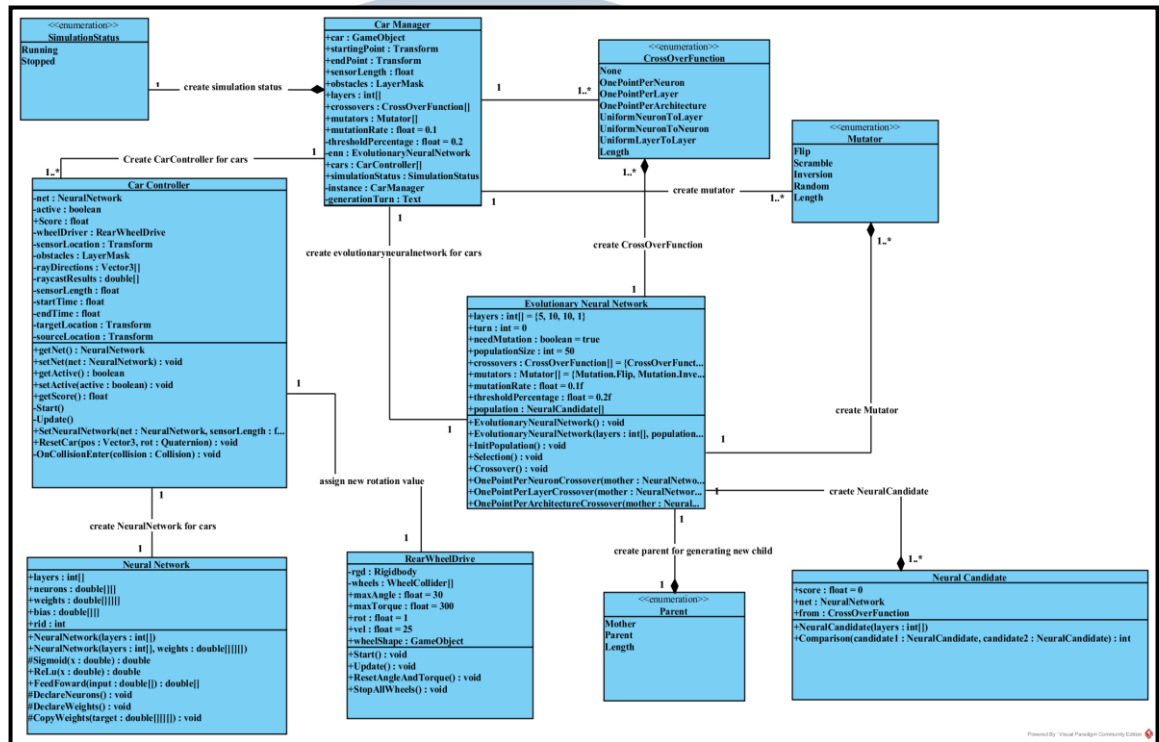
Ketika *robot* menabrak rintangan yang terdapat di dalam simulasi, proses selanjutnya yang akan berjalan adalah proses StopAllWheels. Proses StopAllWheels berfungsi untuk memberhentikan seluruh kerja roda yang terdapat pada *robot*. Selanjutnya, ketika seluruh *robot* telah berhenti proses yang akan berjalan adalah proses CalculateAllCarScore. Proses CalculateAllCarScore berfungsi untuk menghitung *score* yang terdapat pada *robot*, perhitungan *score* berdasarkan dari waktu pertama robot menjalankan simulasi sampai robot tersebut mencapai tujuan atau menabrak rintangan. Selanjutnya, *score* yang telah terhitung diberikan kedalam setiap individu *robot* untuk digunakan pada proses Selection. Selanjutnya setelah *score* telah diberikan ke setiap individu *robot*, proses

RestartSimulation akan berjalan, proses RestartSimulation adalah proses yang mengulang kembali seluruh simulasi yang telah berjalan.

Proses Run adalah proses yang menjalankan proses pada *class evolutionary neural network*, yaitu proses Selection, proses Crossover, dan proses Mutation. Proses Selection adalah proses yang menyeleksi *robot* berdasarkan nilai *score* tertinggi, *robot* dengan *score* yang lebih rendah tidak akan digunakan pada simulasi selanjutnya. Selanjutnya, proses Crossover adalah proses yang menyilangkan 2 *robot* yang memiliki nilai *score* terbaik untuk menghasilkan *robot* baru. Setelah melakukan proses Crossover, setiap individu *robot* baru akan melakukan proses Mutation. Proses Mutation merupakan proses yang menukar nilai gen dengan nilai inversinya, misalnya gennya bernilai 0 akan berubah menjadi 1.

Proses RestartAllCarPosition adalah proses yang mengulang seluruh posisi *robot* yang telah melakukan simulasi ke titik awal. Selanjutnya, proses setNeuralNetwork merupakan proses yang memberikan jaringan syaraf tiruan pada setiap *robot* baru yang telah dihasilkan. Proses ResetCar adalah proses yang mengulang kembali nilai rotasi dan posisi dari *robot* sebelumnya. Setelah melakukan pengulangan nilai rotasi dan posisi dari *robot*, proses ResetAngleAndTorque akan berjalan, proses ResetAngleAndTorque adalah proses yang mengulang nilai *angle* dan *torque* dari *robot* menjadi nilai awal *angle* dan *torque robot* tersebut. Proses setAllCarToActive adalah proses yang memberikan *status active* pada *robot* baru yang telah dihasilkan dari proses sebelumnya sehingga dapat melaksanakan simulasi selanjutnya. Setelah keseluruhan proses telah berjalan, proses tersebut akan terus berulang sampai iterasi yang telah ditentukan atau salah satu *robot* berhasil mencapai titik tujuan yang telah diberikan.

D. Class Diagram



Gambar 3.7 Class Diagram

Gambar 3.7 merupakan *Class Diagram* pada simulasi *autonomous mobile robot navigation*. *Class diagram* diatas menunjukkan hubungan antar *class* yang terdapat pada simulasi. Keterangan pada *class diagram* pada gambar 3.7 akan dijelaskan sebagai berikut.

1. *Class* CarManager mempunyai agregasi dengan *class* CarController dan *class* EvolutionaryNeuralNetwork. Agregasi yang terjadi antara *class* CarManager dengan *class* CarController adalah tipe agregasi satu atau banyak. Sedangkan, agregasi yang terjadi antara *class* CarManager dan *class* EvolutionaryNeuralNetwork adalah tipe agregasi tepat satu.
2. *Class* CarController mempunyai agregasi dengan *class* CarManager, NeuralNetwork, dan RearWheelDrive. Agregasi yang terjadi antara *class* CarController dengan *class* CarController adalah tipe agregasi tepat satu.

Agregasi yang terjadi antara *class* CarController dengan *class* NeuralNetwork adalah tipe agregasi tepat satu. Sedangkan, agregasi yang terjadi antara *class* CarController dengan *class* ReelWheelDrive adalah tipe agregasi tepat satu.

3. *Class* EvolutionaryNeuralNetwork mempunyai agregasi dengan *class* CarManager, dan *class* NeuralCandidate. Agregasi yang terjadi antara *class* EvolutionaryNeuralNetwork dengan *class* CarManager adalah tipe agregasi tepat satu. Sedangkan, agregasi yang terjadi antara *class* EvolutionaryNeuralNetwork dengan *class* NeuralCandidate adalah tipe agregasi satu atau banyak.
4. *Class* NeuralNetwork mempunyai agregasi hanya dengan satu *class* yaitu *class* CarController. Tipe agregasi yang terjadi antara *class* NeuralNetwork dengan *class* CarController adalah agregasi tepat satu.
5. *Class* ReelWheelDrive mempunyai agregasi hanya dengan satu *class* yaitu *class* CarController. Tipe agregasi yang terjadi antara *class* ReelWheelDrive dengan *class* CarController adalah agregasi tepat satu.
6. *Class* NeuralCandidate hanya mempunyai satu agregasi yaitu dengan *class* EvolutionaryNeuralNetwork. Tipe agregasi yang terjadi antara *class* NeuralCandidate dengan *class* EvolutionaryNeuralNetwork adalah tipe agregasi tepat satu.

E. Desain Antarmuka

Dalam implementasi simulasi *autonomous mobile robot navigation* ini, dilakukan rancangan desain antarmuka terlebih dahulu. Rancangan desain antarmuka akan terbagi kedalam 3 bagian yaitu, Rancangan Desain Antarmuka

Main Menu, Rancangan Desain Antarmuka Paused, dan Rancangan Desain Antarmuka Map Simulasi.

E.1 Rancangan Desain Antarmuka Main Menu

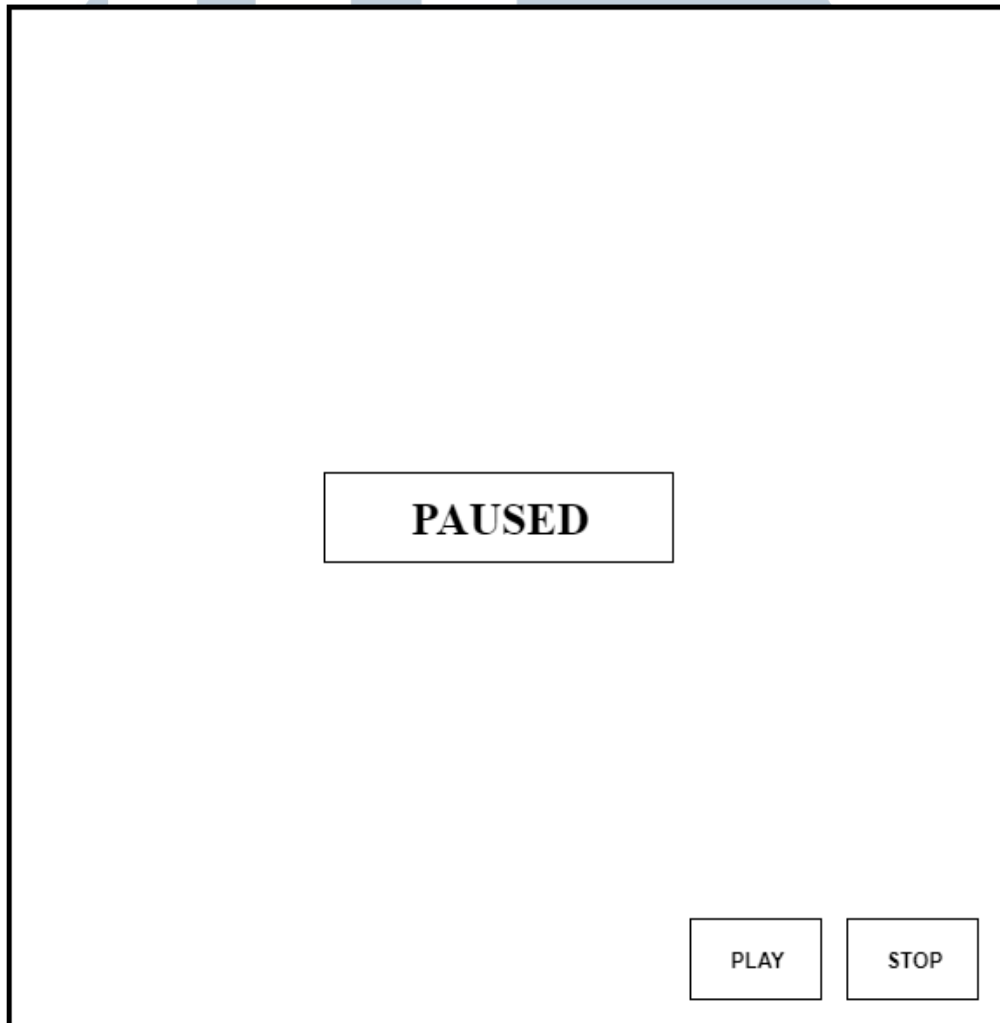
The image shows a software interface for 'ENN SIMULATION'. At the top, there is a title box labeled 'ENN SIMULATION'. Below this, there is a section titled 'ENN INPUT'. Inside this section, there are three rows of input fields: 'Population Size:' with the value '10', 'Total Hidden Layer:' with the value '1', and 'Total Neuron per Layer:' with the value '1'. At the bottom of the 'ENN INPUT' section, there are two buttons: 'QUIT' on the left and 'START SIMULATION' on the right.

Gambar 3.8 Rancangan Desain Antarmuka *Main Menu*

Gambar 3.8 merupakan gambar Rancangan Desain Antarmuka *Main Menu*. Pada *main menu*, user dapat memilih jumlah populasi, total *hidden layer*, dan total *neuron per layer* yang akan digunakan untuk simulasi. Selain itu, terdapat dua tombol pada *main menu* yaitu tombol *quit* dan tombol *start simulation*. Ketika tombol *start simulation* diklik, simulasi akan dimulai dan map simulasi akan

ditampilkan. Sedangkan, ketika *user* menekan tombol *quit* program simulasi akan dihentikan.

E.2 Rancangan Desain Antarmuka Paused



Gambar 3.9 Rancangan Desain Antarmuka *Paused*

Gambar 3.9 merupakan gambar Rancangan Desain Antarmuka *Paused*. Pada tampilan *paused*, terdapat sebuah teks bertuliskan *PAUSED* dan dua tombol dibagian pojok kanan bawah. Ketika tombol *play* diklik oleh *user*, simulasi yang diberhentikan sementara akan dimulai kembali. Sedangkan, ketika tombol *stop* diklik oleh *user*, simulasi akan diberhentikan dan *main menu* akan ditampilkan.

E.3 Rancangan Desain Map Simulasi



Gambar 3.10 Rancangan Desain Antarmuka *Map* Simulasi

Gambar 3.10 merupakan gambar Rancangan Desain Antarmuka *Map* Simulasi. Pada tampilan *map* simulasi, terdapat teks yang berisi jumlah generasi yang telah dilakukan dalam simulasi *autonomous mobile robot navigation*. Selain itu, pada bagian *MAP SIMULATION* berisikan bentuk *map* yang akan digunakan dalam simulasi. Selain itu, terdapat dua button pada tampilan *map* simulasi yaitu tombol *pause* dan tombol *stop*. Ketika tombol *pause* ditekan oleh *user*, tampilan *pause* akan ditampilkan dan simulasi akan diberhentikan sementara. Sedangkan, ketika tombol *stop* ditekan oleh *user*, tampilan *main menu* akan ditampilkan dan simulasi akan diberhentikan secara keseluruhan.