3. DESAIN SISTEM

Pada bab ini, akan dibahas desain sistem yang akan digunakan untuk melakukan prediksi skor pertandingan sepakbola dengan menggunakan *Neuroevolution of Augmenting Topologies* dan *Backpropagation*.

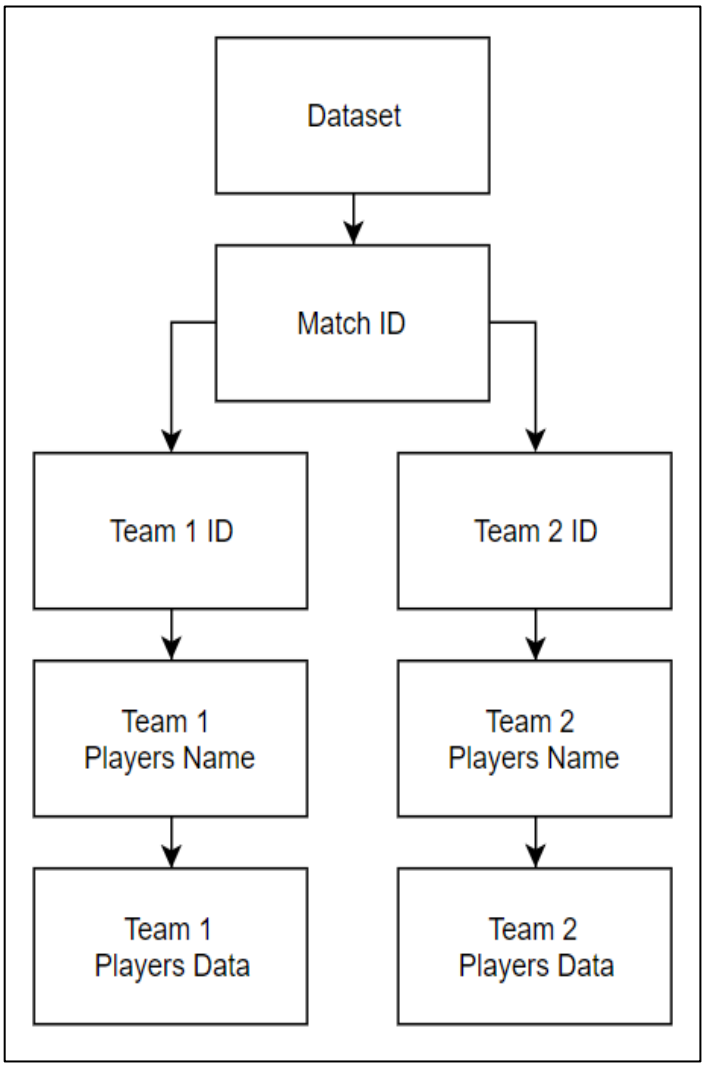
3.1 *Dataset*

Untuk melakukan prediksi, sebuah *machine learning* model membutuhkan data untuk dipelajari terlebih dahulu. Setelah proses *learning* selesai dilakukan, barulah model mampu membuat prediksi berdasarkan data yang yang telah dipelajari.

Dalam penelitian ini, model akan mempelajari data pertandingan sepak bola, dan melakukan prediksi skor pertandingan yang akan datang berdasarkan data tersebut.


```
"Player_stats":{
    "Wojciech Szczesny":{
        "player_details":{
            "player_id":"73379",
            "player_name":"Wojciech Szczesny",
            "player_position_value":"1",
            "player_position_info":"GK",
            "player_rating":"5.81"
        },
        "Match_stats":{
            "touches":"20",
            "saves":"1",
            "total_pass":"13",
            "aerial_won":"1",
            "formation_place":"1",
            "accurate_pass":"11"
        }
    },
    "Calum Chambers":{
        "player_details":{
            "player_id":"124316",
            "player_name":"Calum Chambers",
            "player_position_value":"2",
            "player_position_info":"DC",
            "player_rating":"7.33"
        },
        "Match_stats":{
            "touches":"111",
            "total_tackle":"4",
            "aerial_lost":"2",
            "total_pass":"93",
            "fouls":"1",
            "aerial_won":"2",
            "formation_place":"5",
            "accurate_pass":"86",
            "yellow_card":"1"
        }
    },
    "Mathieu Debuchy":{
        "player_details":{
            "player_id":"11367",
            "player_name":"Mathieu Debuchy",
            "player_position_value":"2",
            "player_position_info":"DR",
```

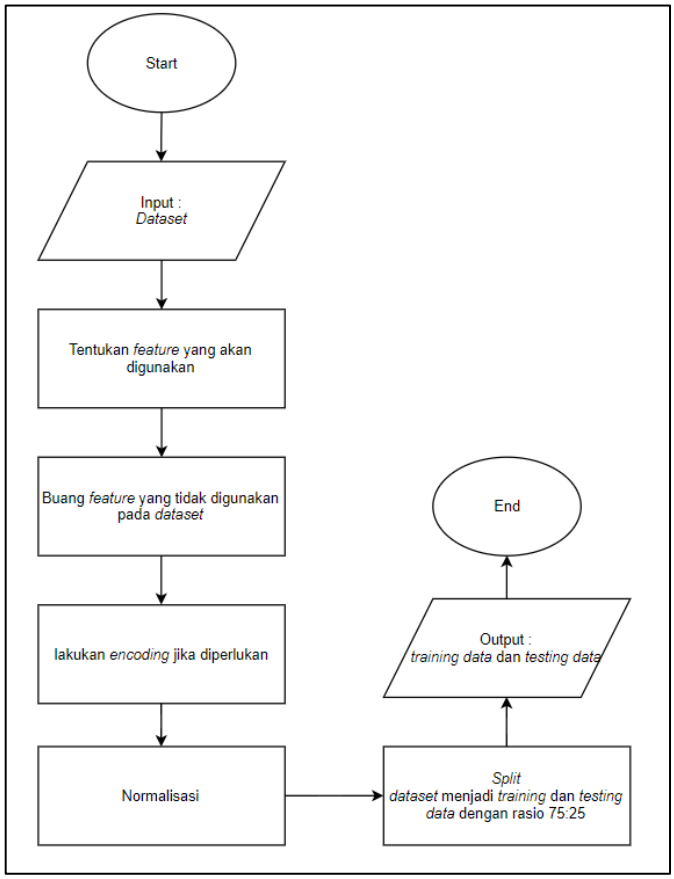

Gambar 3.1 Sebagian dataset yang akan digunakan pada penelitian ini



Gambar 3.2 Struktur dataset yang akan digunakan pada penelitian ini

Dataset yang akan digunakan merupakan data pertandingan dari *English Premier League* yang diperoleh dari situs statistik pertandingan sepak bola, *whoscored.com. Player rating, player position, dan team rating* akan menjadi inputan, sedangkan skor akhir pertandingan akan menjadi *output.* Proses *learning* akan menggunakan data dari 3 musim pertandingan, yaitu dari musim 14/15 sampaimusim 16/17. Seterlah proses *learning* dilakukuan, akan dilakukan validasi

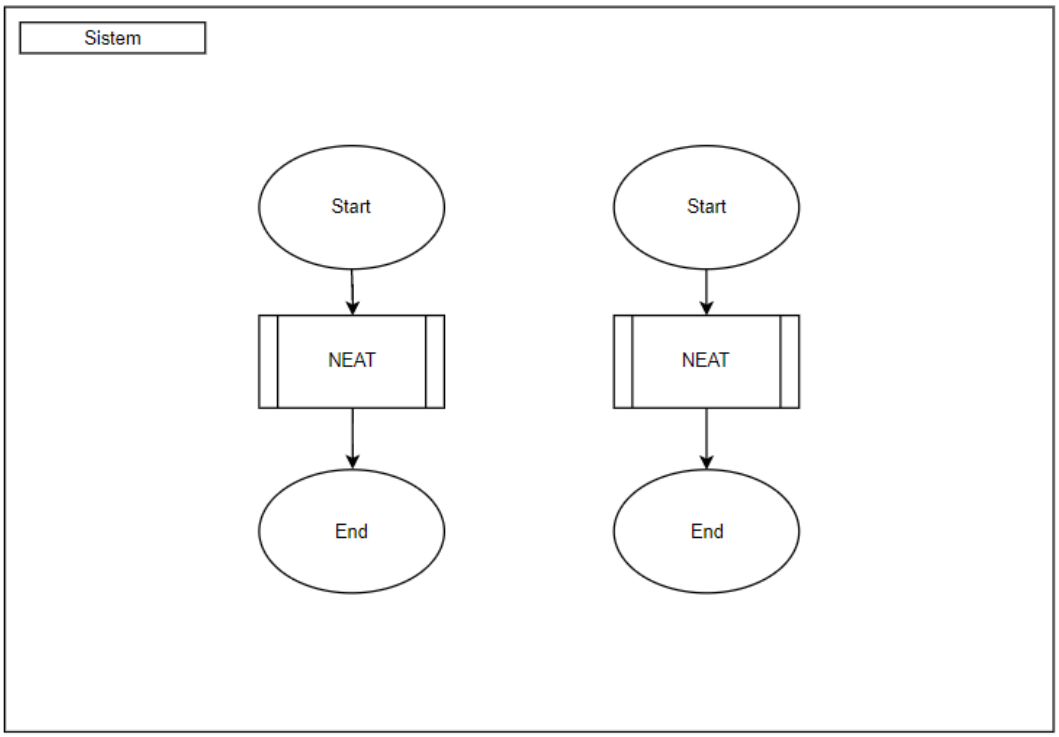
akurasi yang dihasilkan model menggunakan data dari pertandingan pada musim 17/18.



Gambar 3.3 *Flowchart* dari pemrosesan *dataset*

Sebelum data dapat digunakan dalam proses NEAT dan *backpropagation*, akan dilakukan *preprocessing* terlebih dahulu. Seperti yang dapat dilihat pada Gambar 3.1., pada awalnya data yang digunakan berformat *json*. Dari data tersebut kemudian akan ditentukan *feature* apa saja yang akan digunakan untuk melakukan prediksi pertandingan sepak bola. Setelah semua feature yang tidak diperlukan dibuang, selanjutnya akan dilakukan one hot encoding jika ada data yang berbentuk kategori. Kemudian, data akan di*split* menjadi *training* dan *testing* data dengan rasio 75:25. Setelah proses *preprocessing* ini, data siap digunakan.

3.2 Desain Sistem

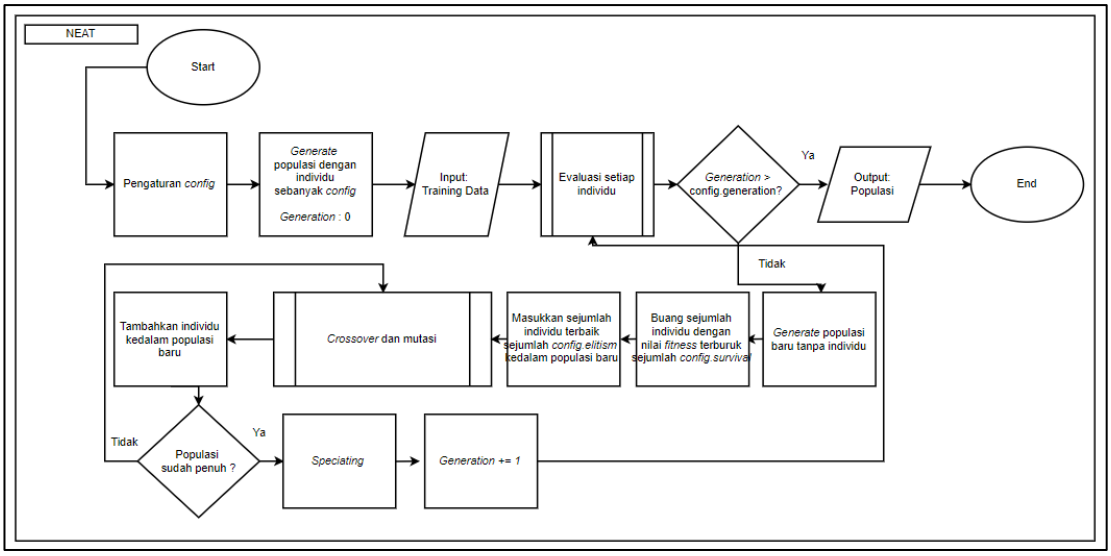


Gambar 3.4 *Flowchart* sistem secara umum

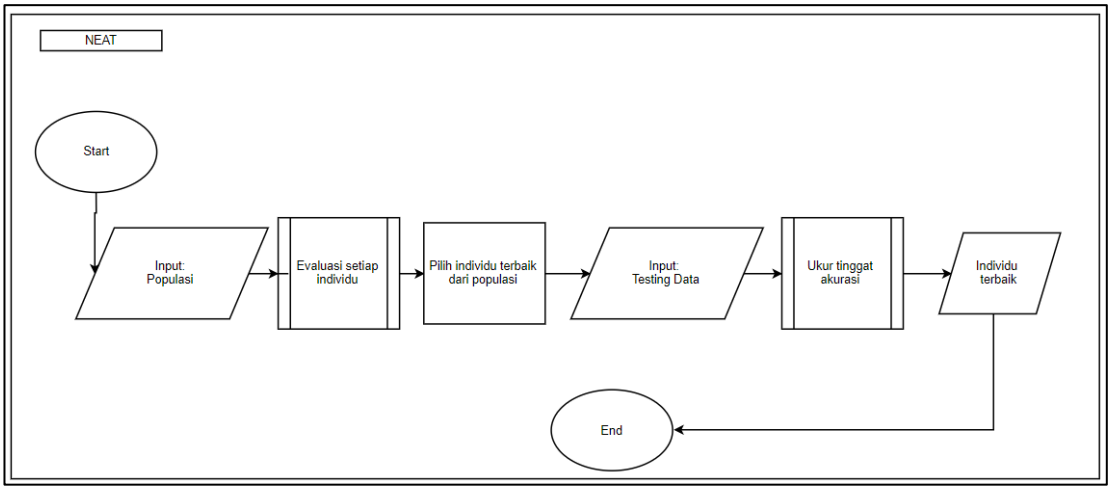
Dapat dilihat pada gambar 3.4., akan ada 2 proses utama pada penelitian ini, yaitu NEAT dan *Backpropagation*, yang masing-masing akan memiliki proses *training* dan *testing*.

Proses NEAT akan mendapat input berupa *training* dan *testing* data. *Training* data digunakan agar model dapat melakukan analisa data sehingga bisa menghasilkan prediksi. *Testing* data, yang berisi data yang tidak dikenali oleh model, akan digunakan untuk mengukur tingkat akurasi dari model. Model yang dihasilkan oleh NEAT kemudian akan dioptimasi oleh *backpropagation* yang bertujuan untuk meningkatkan hasil akurasi dengan menyesuaikan bobot atau *weight* yang ada pada model.

3.2.1 *Neuroevolution of Augmenting Topologies* (NEAT)



Gambar 3.5 *Flowchart* dari proses *training* pada NEAT



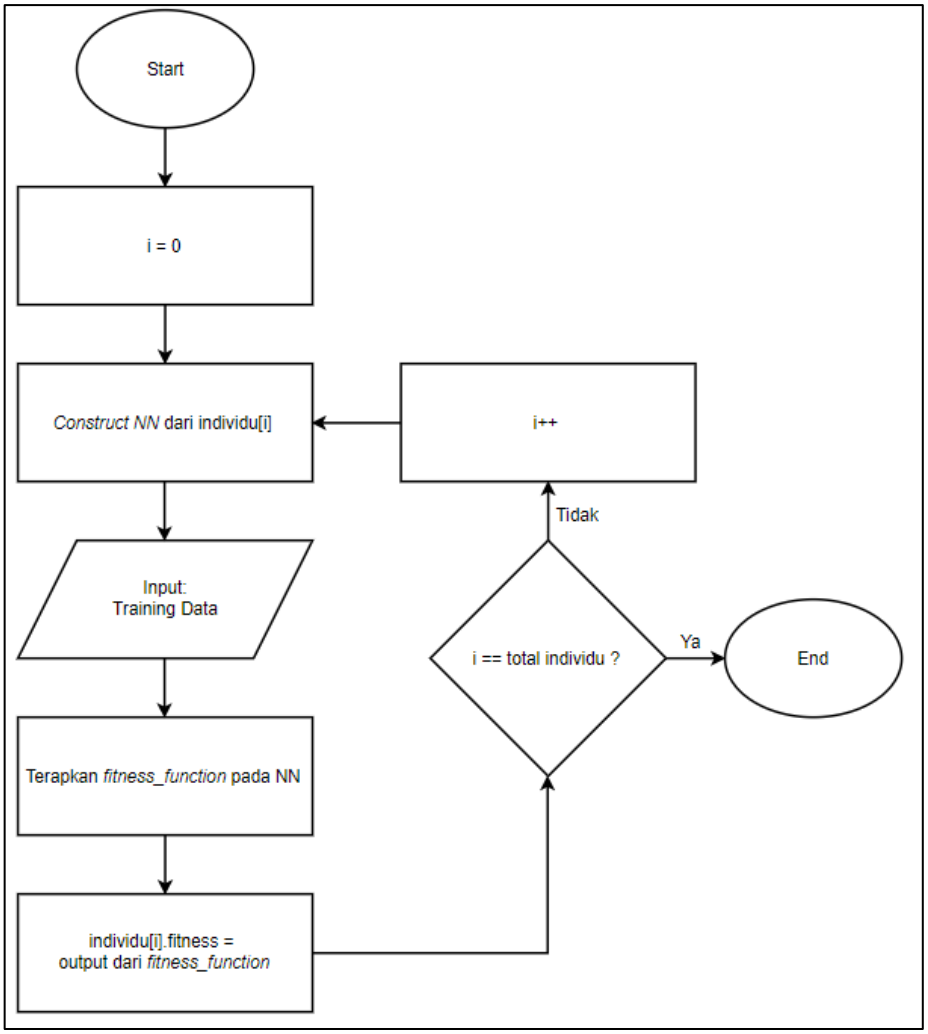
Gambar 3.6 *Flowchart* dari proses *testing* pada NEAT

Proses *training* pada NEAT dimulai dengan melakukan pengaturan *config*, yang merupakan kumpulan dari aturan-aturan yang berlaku selama proses *training* berlangsung, seperti jumlah individu dalam populasi, jumlah generasi yang diinginkan, tingkat kemungkinan terjadinya mutasi, dan lain lain. Pada tahap selanjutnya, NEAT akan membuat sebuah populasi baru dengan jumlah individu yang sudah ditentukan pada *config*. Kemudian, setiap individu akan dievaluasi menggunakan *fitness function* berdasarkan *training data* yang diinputkan untuk menentukan nilai *fitness* dari masing-masing individu. Selanjutnya, akan

dilakukan pengecekan *stopping condition*, yaitu sebuah kondisi untuk menentukan apakah proses NEAT akan berhenti atau tidak. *Stopping condition* disini adalah jumlah generasi. Jika jumlah generasi sudah melebihi jumlah generasi yang ditentukan pada *config*, Maka populasi akan di*export* dan proses *training* berhenti sampai disini. Jika *stopping condition* tidak terpenuhi, NEAT akan membuat sebuah populasi baru yang tidak memiliki individu, populasi baru ini nantinya akan diisi oleh individu-individu baru hasil dari *crossover* dan mutasi. Setelah itu, akan dilakukan seleksi, sejumlah individu pada populasi lama yang memiliki nilai *fitness* terburuk akan dibuang, sehingga ketika terjadinya proses *crossover*, individu-individu tersebut tidak dapat terpilih. Proses selanjutnya pada NEAT adalah *crossover* dan mutasi. Hasil dari proses *crossover* dan mutasi adalah sebuah individu baru yang akan ditambahkan ke dalam populasi baru. Proses *crossover* dan mutasi ini akan terjadi terus menurus hingga populasi baru memiliki individu sebanyak *pop size* pada *config*. Ketika populasi baru sudah penuh, maka akan dilakukan *speciating*, yaitu pembagian individu-individu yang ada pada populasi baru kedalam beberapa spesies. Setelah *speciating*, generasi akan bertambah dan populasi baru akan menggantikan populasi lama. Proses ini akan terus berulang hingga *stopping condition* terpenuhi.

Pada proses *testing*, NEAT akan menerima input berupa populasi yang dihasilkan oleh proses *training*. Setelah itu, akan dilakukan evaluasi yang sama seperti pada proses *training* dan akan dipilih satu individu dengan nilai *fitness* tertinggi. Individu terbaik tersebut kemudian akan diukur tingkat akurasinya dan diekstrak untuk dioptimasi menggunakan *backpropagation* pada proses selanjutnya.

3.2.1.1 Evaluasi pada NEAT



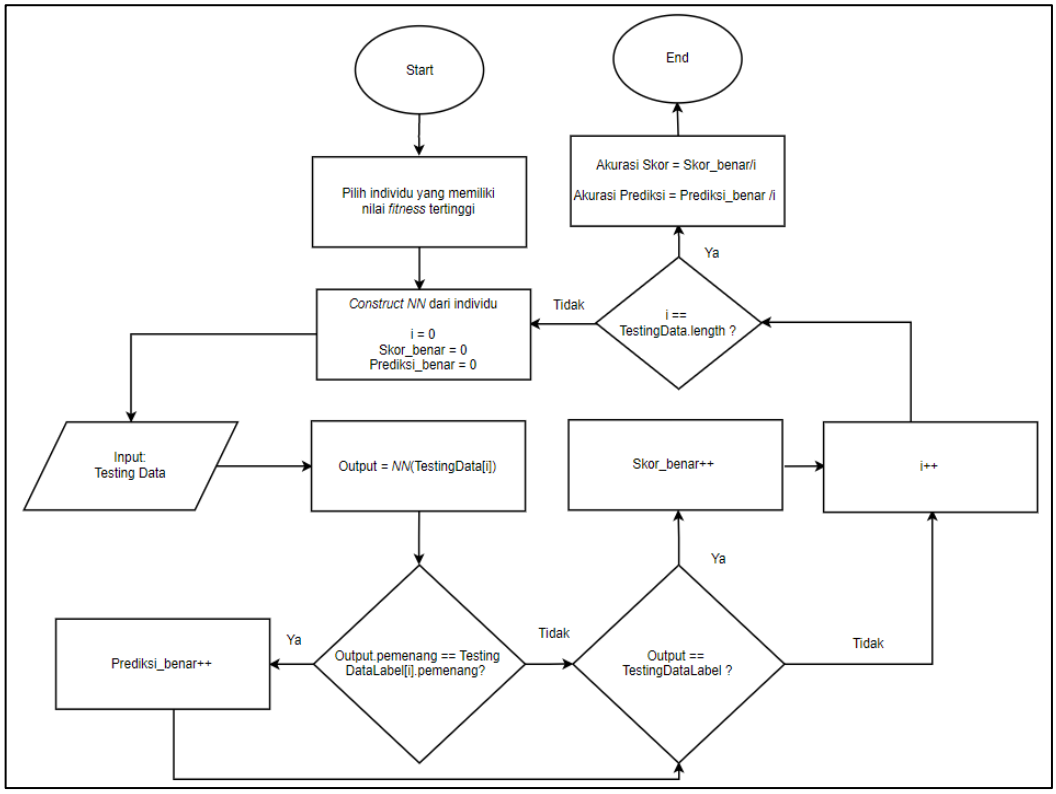
Gambar 3.7 *Flowchart* dari proses evaluasi pada NEAT

Evaluasi pada NEAT merupakan suatu proses yang sangat penting. Tujuan dari proses ini adalah untuk menentukan nilai *fitness*, yang merupakan indikator untuk mengukur seberapa baik sebuah individu dalam populasi.

Langkah pertama dari proses evaluasi adalah memilih individu pada sebuah populasi. Pada setiap individu, terdapat *genome* yang berisikan informasi tentang individu tersebut. Dari informasi yang berasal dari *genome*, dapat dibangun sebuah *Neural Network*, yang merupakan *phenotype* dari individu tersebut. *Neural Network* akan mendapat input berupa *training data*. Kemudian, akan diaplikasikan sebuah *fitness function* kepada *neural network* tersebut. *Output* dari *fitness function* inilah yang akan menjadi nilai *fitness* pada individu tersebut.

Proses Evaluasi ini akan terus dijalankan secara iteratif hingga semua individu pada populasi memiliki nilai *fitness*.

3.2.1.2 Pengukuran Tingkat Akurasi pada NEAT



Gambar 3.8 *Flowchart* dari proses pengukuran akurasi pada NEAT

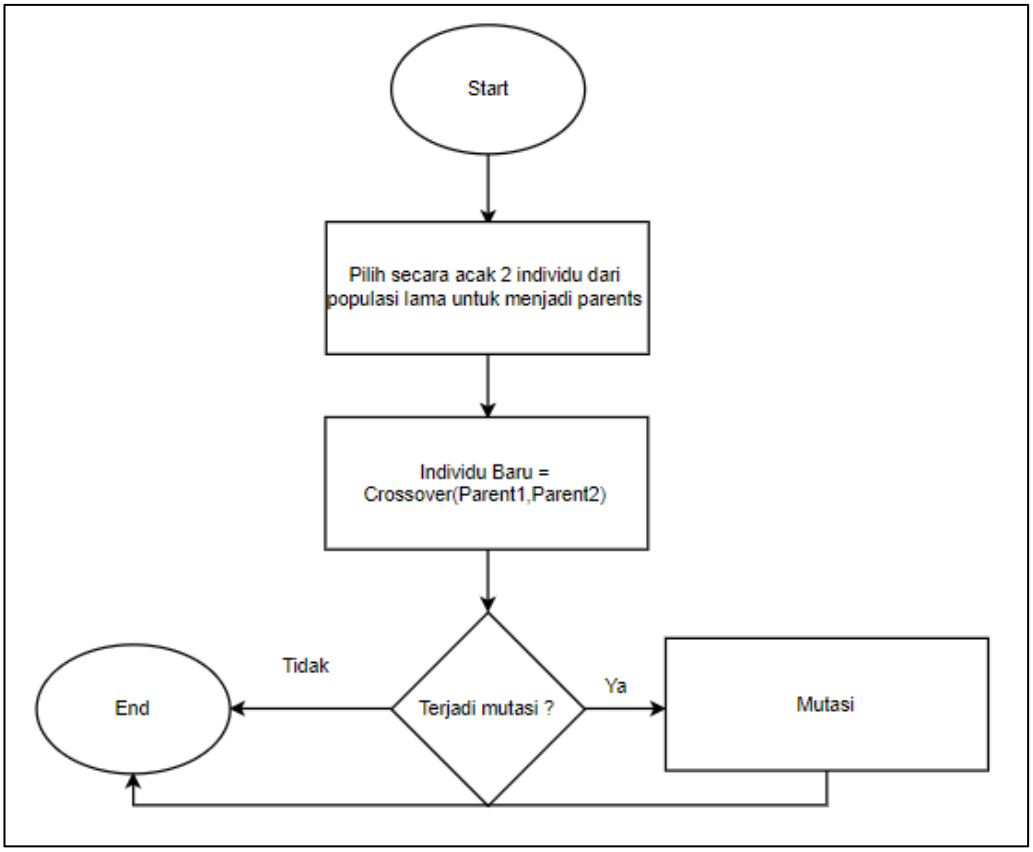
Ketika *stopping condition* pada NEAT terpenuhi, akan dilakukan pengukuran tingkat akurasi pada individu yang memiliki nilai *fitness* terbaik, dan individu terbaik tersebut akan diekstrak untuk dioptimasi menggunakan *backpropagation*. Pengukuran tingkat akurasi pada NEAT bertujuan sebagai pembanding seberapa besar pengaruh *backpropagation* pada individu yang dihasilkan oleh NEAT. Ada 2 jenis akurasi yang dihasilkan pada proses ini, yaitu akurasi skor dan akurasi prediksi. Pada akurasi skor, individu harus bisa melakukan prediksi skor secara tepat untuk dikatakan akurat, sedangkan pada akurasi prediksi, individu hanya perlu menebak tim mana yang keluar sebagai pemenang.

Proses pengukuran dimulai dengan memilih individu dengan nilai *fitness* terbaik dari populasi. Sama seperti proses evaluasi, dari individu tersebut akan dibangun sebuah *neural network*. Tetapi, pada proses ini data yang digunakan

adalah data *testing*, yang merupakan data yang tidak pernah diproses oleh model sebelumnya.

Output dari *neural network* yang berupa skor pertandingan kemudian akan dibandingkan dengan label yang tersedia pada *testing data*. Setelah semua *testing* data diproses, dapat dihitung tingkat akurasi dari individu tersebut.

3.1.2.3 *Crossover* dan Mutasi pada NEAT



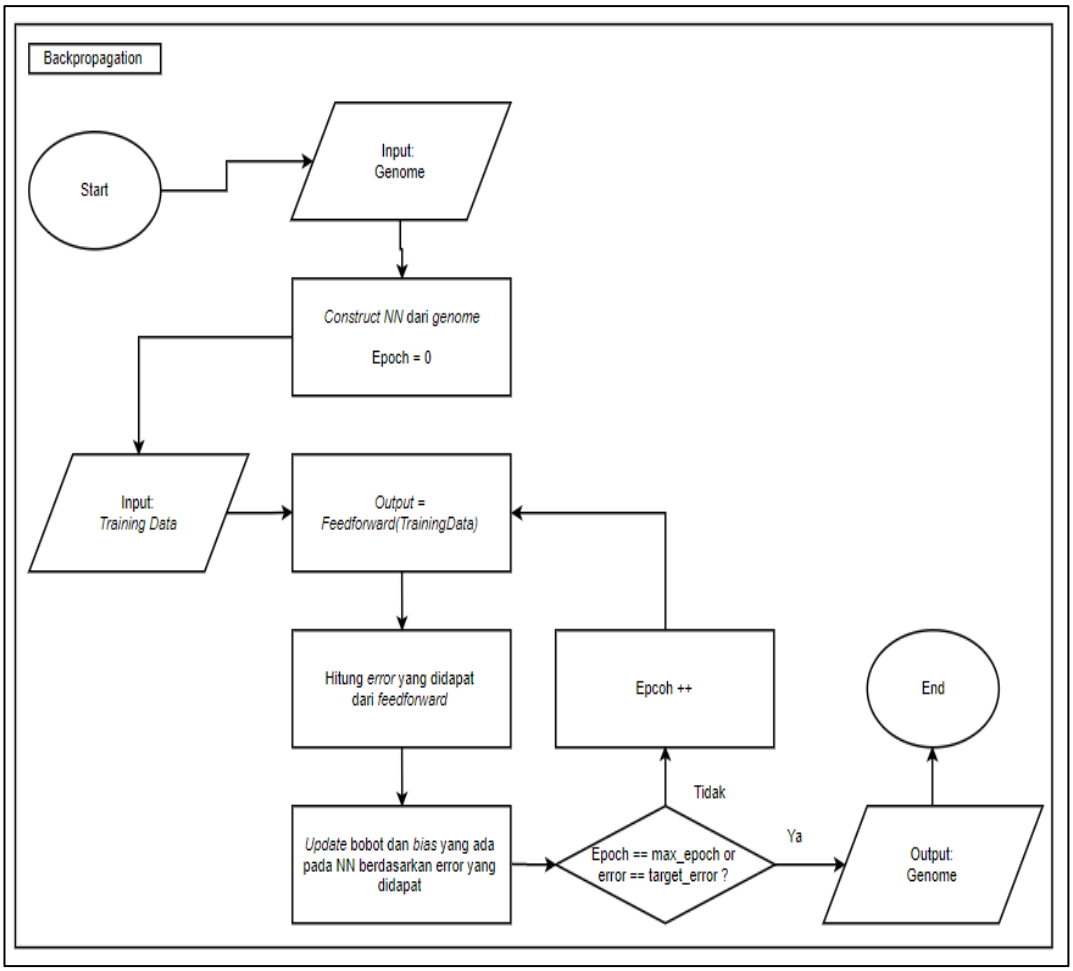
Gambar 3.9 *Flowchart* dari proses *crossover* dan mutasi pada NEAT

Crossover dan mutasi adalah proses terpenting pada NEAT. Proses ini memungkinkan terciptanya suatu inovasi yang memiliki potensi untuk mengungguli individu lama dalam nilai *fitness*.

Sebelum terjadinya *crossover*, 2 individu dalam populasi lama akan dipilih secara acak untuk menjadi *parents*. *Parents* tersebut kemudian akan melakukan *crossover*, yang akan menghasilkan 1 individu baru. Individu baru ini kemudian akan memiliki kemungkinan untuk melakukan mutasi. Kemungkinan terjadinya mutasi ini ditentukan oleh *mutation rate* yang ada pada *config*. Setelah proses

crossover dan mutasi selesai, individu baru tersebut akan ditambahkan kepopulasi baru. Proses *crossover* dan mutasi akan terjadi terus menerus hingga populasi baru terpenuhi.

3.2.2 *Backpropagation*

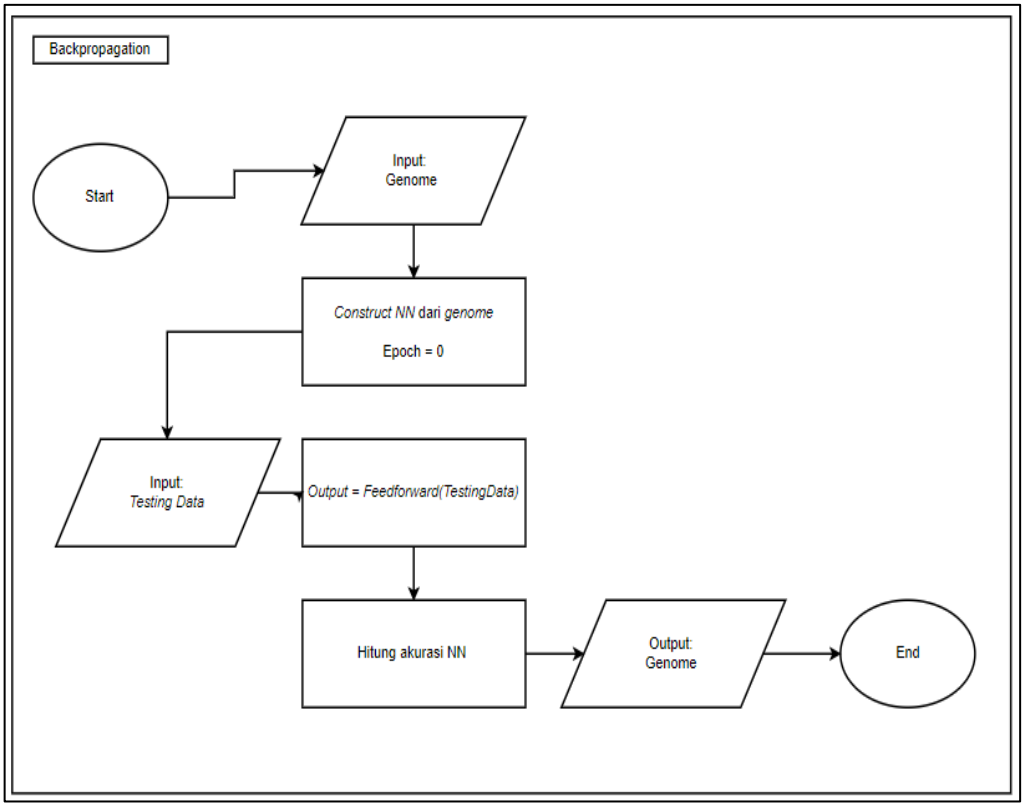


Gambar 3.10 *Flowchart* dari proses *training backpropagation*

Ketika proses NEAT selesai, yang menghasilkan *output* berupa sebuah *genome* dengan nilai *fitness* terbaik dari populasi, proses selanjutnya adalah melakukan optimasi pada *genome* tersebut dengan *backpropagation*.

Proses *training* pada *backpropagation* dimulai dengan membangun sebuah *neural network* dari *genome*. *Neural network* kemudian akan mendapat input berupa *training* data. Berdasarkan *training* data yang diterima, *neural network* akan mengeluarkan output berupa prediksi skor akhir dari sebuah pertandingan. Hasil prediksi yang dihasilkan oleh *neural network* kemudian akan dibandingkan dengan

label yang ada pata *training* data, yang merupakan skor yang benar dari pertandingan tersebut, untuk mendapatkan nilai *error*. Nilai *error* ini akan digunakan untuk menyesuaikan *weight* atau bobot dan *bias* yang ada pada *neural network.*

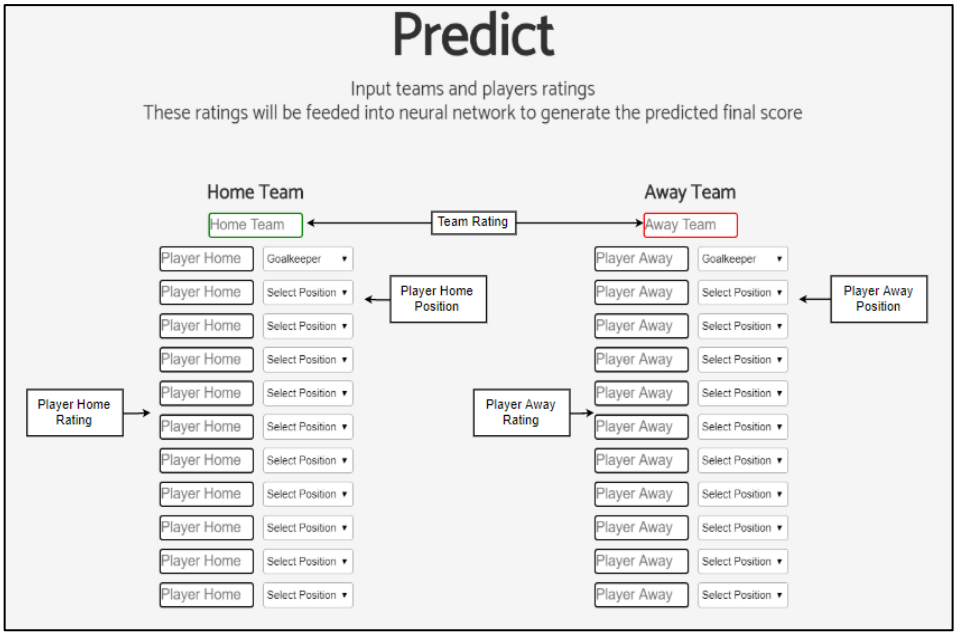


Gambar 3.11 *Flowchart* dari proses *testing backpropagation*

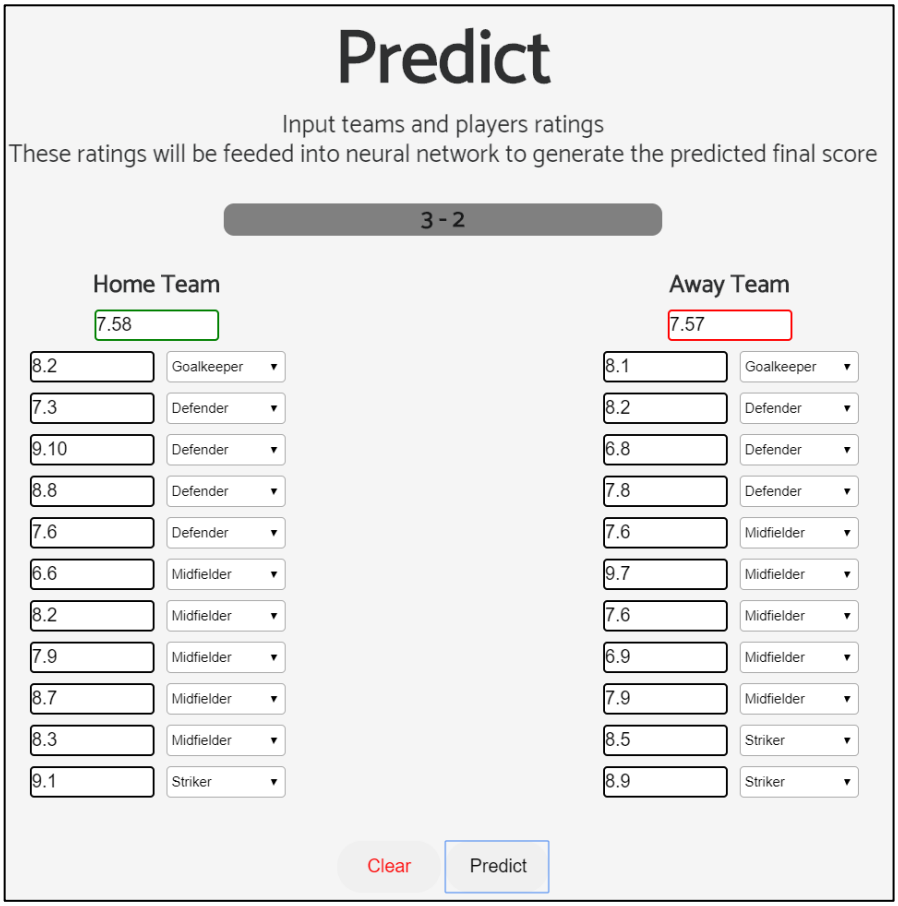
Proses *backpropagation* akan dijalankan terus menurus hingga target *epoch* tercapai, yang ditentukan sebelum proses dimulai, atau jika nilai *error* sudah mencapai target *error*, yang juga ditentukan sebelum proses dimulai.

Setelah proses *backpropagation* selesai, akan dilakukan proses testing dan pengukuran akurasi dari *neural network* yang telah dioptimasi untuk menganalisa seberapa besar pengaruh yang dihasikan oleh *backpropagation* terhadap tingkat akurasi dari sebuah *genome* yang dihasilkan oleh NEAT.

3.3 Desain Aplikasi



Gambar 3.12 Desain halaman utama untuk melakukan prediksi



Gambar 3.13 Halaman untuk menampilkan hasil prediksi

User harus mengisi *rating* dari kedua team pada kolom *Home Team* dan *Away Team*. Setelah itu, *user* juga harus mengisi *rating* dan posisi pemain dari masing-masing *team*. Ada 3 posisi yang dapat dipilih, yaitu *defender*, *midfielder*, dan *striker*. Setelah semua kolom terisi, *user* dapat menekan tombol *predict* untuk melakukan prediksi.

Tombol *predict* akan mengarahkan *user* kehalaman hasil, yaitu halaman yang berfungsi untuk menampilkan hasil prediksi dari data yang telah dimasukkan oleh *user*. Jika *user* menekan tombol *predict again*, user akan diarahkan kehalaman utama untuk melakukan prediksi lagi.