

Prediksi Pertandingan Sepak Bola Menggunakan Neuroevolution of Augmenting Topologies dan Backpropagation

Welly Winata, Lily Puspa Dewi², Alvin Nathaniel Tjondrowiguno³
Program Studi Informatika Fakultas Teknologi Industri Universitas Kristen Petra
Jl. Siwalankerto 121 – 131 Surabaya 60236
Telp. (031) – 2983455, Fax. (031) - 8417658
Email: winatawelly16@gmail.com¹, lily@petra.ac.id², alvin.nathaniel@petra.ac.id³

ABSTRAK

Sepak bola merupakan olahraga yang memiliki penggemar paling banyak di dunia. Hal yang membuat sepak bola menjadi sangat populer adalah hasil yang tidak pasti dan sulit ditebak. Ada banyak faktor yang mempengaruhi hasil dari sebuah pertandingan sepak bola, diantaranya *strategy*, *skill*, bahkan sampai keberuntungan. Karena itu, menebak hasil pertandingan sepak bola merupakan masalah yang menarik.

Penelitian dimulai dengan *neuroevolution of augmenting topologies*, yang berfungsi untuk melakukan pencarian struktur dari sebuah *neural network*. Lalu, *network* yang dihasilkan oleh NEAT akan dioptimasi menggunakan *backpropagation*. *Rating* pemain, *rating team*, dan posisi pemain akan digunakan sebagai *features*.

Tingkat akurasi terbaik yang didapat sebesar 81.5% pada akurasi hasil pertandingan, dan 48% pada akurasi skor pertandingan diperoleh melalui proses NEAT yang telah dioptimasi oleh *backpropagation* menggunakan *rating* pemain, *rating team*, dan jumlah masing-masing posisi pada setiap sektor sebagai *features*.

Pada pengujian *real life*, *rating* pemain dan *team* tidak diketahui, sehingga digunakan metode rata-rata untuk menghitung *rating* dari pemain dan *team*. Namun, akurasi yang didapat pada pengujian ini sangat rendah, inkonsistensi dari pemain menyebabkan metode rata-rata yang digunakan tidak mampu bekerja dengan baik.

Kata Kunci: *Machine Learning, Artificial Neural Network, Neuroevolution, Neuroevolution of Augmenting Topologies, Backpropagation*

ABSTRACT

Football, or soccer is the most popular sport in the world. What makes football special is the uncertainty and unpredictable result. There are a lot of factors that can affect the result of a football match, such as *strategy*, *skill*, or even *luck*. Therefore, predicting the outcome of football match can be challenging yet interesting task.

This research started with *neuroevolution of augmenting topologies*, which useful to find the structure of a *neural network*. Then, the *network* produced by NEAT is optimized using *backpropagation*. Player ratings, team ratings, and player position are used as features of *neural network*.

The highest accuracies achieved are 81.5% on the final result predicting, and 48% on score predicting, were obtained through NEAT network that optimized by *backpropagation*, with player ratings, team ratings, and total position from each sectors are used as features.

However, on real life test, the player and team ratings are unknown. To calculate the player and team ratings, averages methods are used. Unfortunately, the network performed poorly causing the accuracies to dropped significantly. Lack of consistency from player ratings are believed to be the main problem on calculating the player and team ratings.

Keywords: *Machine Learning, Artificial Neural Network, Neuroevolution, Neuroevolution of Augmenting Topologies, Backpropagation*

1. PENDAHULUAN

Dalam artikel yang diterbitkan oleh *Bloomberg* pada tahun 2018, 4 dari 10 orang menyatakan bahwa mereka adalah penggemar sepak bola [2]. Ini menjadikan sepakbola sebagai olahraga paling populer di dunia. Ketidakpastian merupakan sifat alami dari sepak bola [8] yang menjadikan ini sebagai salah satu faktor mengapa sepak bola sangat disukai.

Seperti yang dikatakan *The New York Times* dalam artikelnya yang berjudul *Soccer, a Beautiful Game by Chance* banyak sekali komponen yang mempengaruhi hasil akhir dari sebuah pertandingan, seperti *strategy*, *skill*, dan *luck*. Faktor-faktor tersebutlah yang membuat hasil dari setiap pertandingan unik dan sulit diprediksi. Namun, dari setiap pertandingan sepak bola dapat diperoleh data yang dapat digunakan untuk menganalisa bagaimana jalannya pertandingan.

Dengan semakin berkembangnya teknologi, data-data penting yang berkaitan dengan pertandingan sepak bola semakin mudah didapat. Data-data tersebut dapat diolah dan digunakan untuk melakukan prediksi pada pertandingan yang akan datang. Salah satu bidang dalam *Computer Science* yang banyak digunakan untuk melakukan prediksi berdasarkan data adalah *Machine Learning*.

Machine Learning, dalam definisinya, adalah suatu bidang dalam *Computer Science* yang dapat mempelajari pola tertentu dari kumpulan data dan membuat prediksi atau klasifikasi berdasarkan kumpulan data tersebut. Penggunaan *Machine Learning* dalam masalah seperti ini sangat cocok, karena selain banyaknya data yang tersedia, sepak bola juga sulit diprediksi berdasarkan logika, maupun alasan-alasan eksplisit lainnya [9]. Beberapa contoh algoritma *Machine Learning* yang sedang populer saat ini adalah *Artificial Neural Network* (ANN) dan *Support Vector Machine* (SVM).

Pada penelitian sebelumnya [5], SVM dan ANN pernah digunakan untuk melakukan prediksi sepakbola, tetapi hasil yang didapat oleh SVM sangat mengecewakan, akurasi yang didapat hanya sebesar 53.3%, sedangkan ANN secara impresif mampu menghasilkan akurasi diatas 80%. Berkaca dari hasil penelitian tersebut, metode

yang akan digunakan dalam penelitian kali ini adalah *Neuroevolution of Augmenting Topologies* (NEAT).

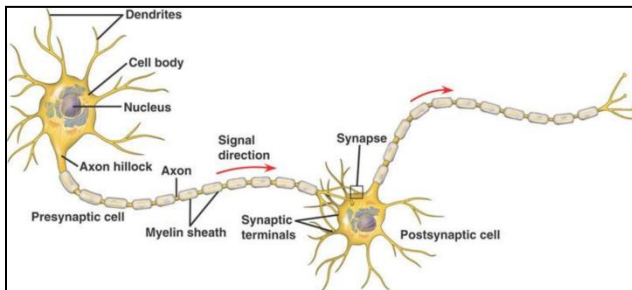
NEAT adalah algoritma penyempurnaan dari *Neuroevolution* (NE) yang berasal dari penggabungan antara ANN dan *Evolutionary Algorithm* (EA). Salah satu kelebihan Perbedaan NE jika dibandingkan dengan ANN tradisional adalah topologi jaringan yang dapat melakukan evolusi seiring berjalannya proses *training* [6]. Tetapi, NE juga memiliki kekurangan, yaitu saat terjadi *crossover* antara 2 jaringan, adanya kemungkinan *offspring* yang dihasilkan memiliki informasi yang tidak lengkap.

Kekurangan yang ada pada NE dapat diselesaikan oleh NEAT. NEAT menyelesaikan masalah ini dengan cara melacak *innovation number* ketika terjadi *crossover*. Setelah proses NEAT selesai, akan dilakukan optimasi menggunakan metode yang biasa digunakan pada ANN pada umumnya, yaitu *backpropagation*. Berbeda dengan penelitian terdahulu, penelitian kali ini akan mencoba melakukan prediksi skor akhir sebuah pertandingan sepak bola, bukan hanya tim mana yang akan memenangkan pertandingan.

2. TINJAUAN PUSTAKA

2.1 Jaringan Syaraf Tiruan

Jaringan saraf tiruan (JST) atau *Artificial Neural Network* (ANN) merupakan cabang dari *machine learning* yang menggambarkan representasi buatan dari otak manusia yang mencoba untuk mensimulasikan proses pembelajaran dari otak manusia [7]. Bentuk representasi ANN adalah berupa jaringan yang terdiri dari kumpulan unit pemroses kecil yang biasa disebut neuron, yang bersifat adaptif karena mampu mengubah struktur unit-unit tersebut untuk menyelesaikan masalah berdasarkan informasi (input) baik informasi eksternal, maupun informasi internal. ANN mampu belajar layaknya otak manusia dengan cara memberi bobot pada tiap neuron. Saat proses pembelajaran sedang berlangsung, neuron akan di *update* berdasarkan *error* yang didapat.



Gambar 1. Ilustrasi neuron pada otak manusia

Setiap neuron pada JST memiliki bobot atau *weight* yang menghubungkan suatu neuron kepada neuron lainnya. Selain menghubungkan tiap neuron, bobot juga berguna untuk melakukan *scaling* terhadap input yang diterima oleh setiap neuron. *Output* dari tiap neuron kemudian akan diteruskan ke neuron selanjutnya hingga mencapai neuron pada lapisan terakhir. Proses ini disebut dengan *feed-forward* atau *forward-pass*.

2.2 Backpropagation

Backpropagation adalah suatu proses pembelajaran bertipe *supervised learning* pada JST [7]. Pada umumnya, terdapat 3 lapisan pada JST, yaitu lapisan input, lapisan tersembunyi, dan lapisan *output*. Berbeda dengan *forward-pass* yang dimulai dari lapisan terdepan atau lapisan input, *backpropagation* dimulai dari lapisan paling akhir atau lapisan *output*.

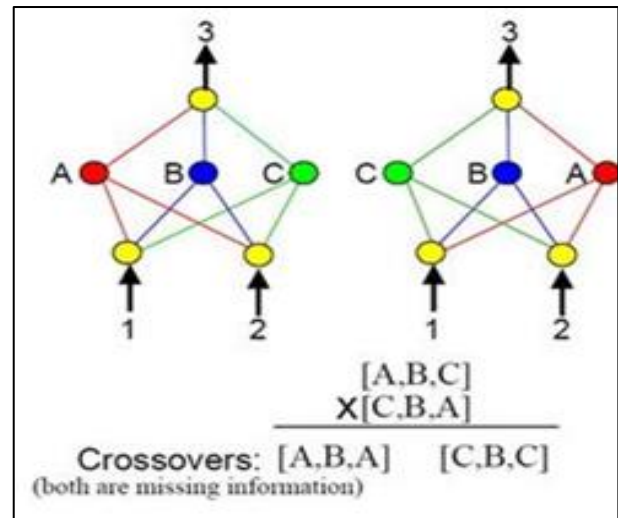
Backpropagation bekerja dengan cara menghitung *error* dari sebuah data, *error* didapat dengan cara $error = (target - prediction)^2$

yang kemudian *error* tersebut akan digunakan untuk menyesuaikan atau meng-*update* bobot-bobot yang ada pada JST, sehingga kesalahan atau *error rate* dapat diperkecil yang kemudian akan menghasilkan prediksi yang lebih akurat [1].

2.3 Neuroevolution of Augmenting Topologies

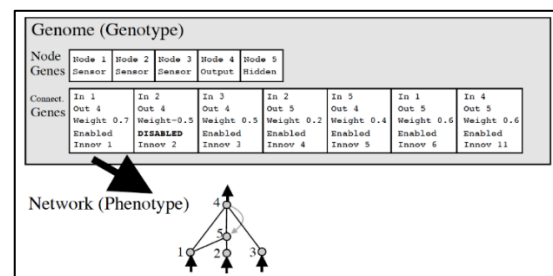
Neuroevolution (NE) merupakan sebuah algoritma penggabungan antara *Genetic Algorithm* (GA) dan *Artificial Neural Network* (ANN). Dalam tradisional NE, akan dilakukan pemilihan topologi dari sebuah ANN sebelum eksperimen dimulai. Biasanya, topologi dari ANN berupa 1 lapisan tersembunyi yang terhubung ke semua lapisan input dan lapisan *output*. Pencarian bobot-bobot atau *weights* kemudian akan dilakukan menggunakan GA, seperti *crossover* dan *mutation*. Maka, tujuan dari tradisional NE atau *Fixed-Topology Neuroevolution* adalah melakukan optimasi pada bobot-bobot sehingga dapat menemukan ANN yang fungsional [10].

Namun, bobot-bobot yang ada pada ANN bukan penentu satu-satunya terhadap *behaviour* dari sebuah ANN. Topologi atau struktur dari ANN itu sendiri juga mempengaruhi bagaimana ANN bekerja [4]. Selain itu, ketika terjadi *crossover* pada tradisional NE, adanya kemungkinan informasi akan hilang, sehingga menciptakan *offspring* yang “cacat”.



Gambar 2. Hilangnya informasi ketika terjadi *crossover* antar ANN

Pada ilustrasi diatas, dapat dilihat bahwa hilangnya informasi pada *offspring* yang dihasilkan oleh hasil *crossover* antara jaringan ABC dan CBA. *Neuroevolution of Augmenting Topologies* (NEAT) mencoba menjawab permasalahan ini dengan memberikan *historical marking* dengan *innovation number* pada setiap *connection genes* yang ada didalam jaringan.

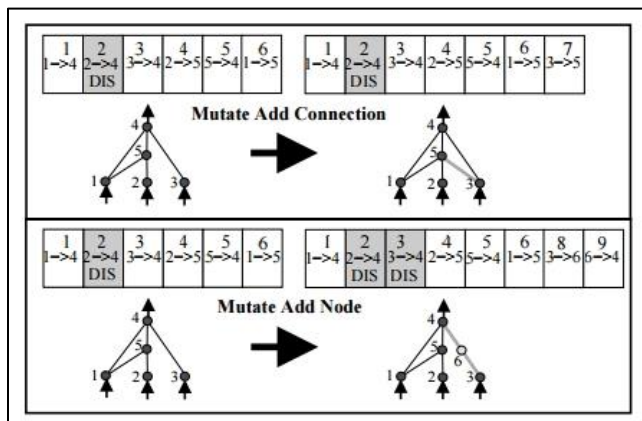


Gambar 3. Encoding dan *innovation number* pada NEAT

2.3.1 Genetic Encoding

Genetic Encoding adalah bentuk representasi linear dari koneksi yang ada pada jaringan [10]. *Genetic Encoding* pada NEAT didesain agar mudah untuk disejajarkan ketika terjadi *crossover*. Setiap *genome* pada NEAT terdiri dari 2 *genes*, yaitu *node genes* dan *connection genes*. *Connection genes* berisi atas kumpulan koneksi pada jaringan, yang mengacu kepada 2 *node genes* yang masing-masing mempunyai informasi tentang *nodes* yang tersedia pada jaringan. *Connection genes* juga berisikan atas informasi tentang *in-node*, *out-node*, bobot dari koneksi, apakah koneksi tersebut aktif, dan *innovation number*, yang nanti akan berguna ketika terjadinya *crossover*.

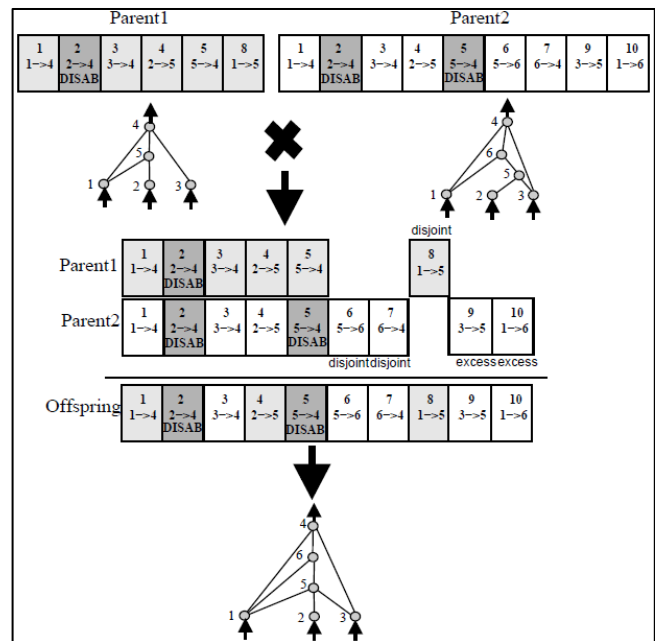
Mutation pada NEAT dapat mengubah bobot koneksi atau struktur dari jaringan itu sendiri. Perubahan bobot yang terjadi karena *mutation* sama dengan *mutation* yang ada pada NE tradisional, yaitu antara terputusnya koneksi atau tidak [10]. Sebaliknya, untuk *structural mutation*, ada 2 kemungkinan yang bisa terjadi, yaitu *add connection mutation* dan *add node mutation*. Pada *add connection mutation*, satu koneksi baru dengan bobot acak akan muncul untuk menyambungkan 2 *node* yang sebelumnya tidak tersambung. Selanjutnya, pada *add node mutation*, koneksi antara 2 *nodes* akan terbagi dan *node* baru akan muncul diantara 2 *nodes* yang bersambungan tersebut. Koneksi lama yang menyambungkan 2 *nodes* awal akan mati, dan 2 koneksi baru akan ditambahkan ke *connection genes*. Koneksi pertama akan menyambungkan *node-in* awal ke *node* baru, kemudian koneksi kedua akan menyambungkan *node* baru kepada *node-out* awal.



Gambar 4. Structural mutation pada NEAT

2.3.2 Historical Marking

Historical marking memberi NEAT suatu kemampuan yang powerful. Dengan *historical marking*, NEAT dapat mengetahui dengan tepat tentang kecocokan suatu *gene* dengan *gene* lainnya. Ketika terjadi *crossover*, *connection genes* akan disejajarkan untuk mencocokkan *innovation number* pada kedua *genes*. *Genes* yang memiliki *innovation number* yang sama disebut sebagai *matching genes*. Sedangkan *genes* yang tidak memiliki kecocokan disebut sebagai *disjoint* atau *excess genes*, tergantung apakah mereka muncul dalam *range innovation number* yang dimiliki oleh *parent* lainnya. *Disjoint* dan *excess genes* disini mewakili struktur yang tidak dimiliki oleh salah satu *genome* saat terjadinya *crossover*. Dalam pembuatan sebuah *offspring*, *matching genes* akan dipilih salah satu secara acak dari kedua *parents genome*. Sedangkan *disjoint* dan *excess genes* akan diturunkan oleh *parent* yang memiliki nilai *fitness* yang lebih baik.



Gambar 5. Crossover pada NEAT

2.3.3 Speciation

Membagi populasi kedalam beberapa spesies memungkinkan suatu individu untuk berkompetisi dengan individu sejenisnya. Dengan cara ini, inovasi yang terbentuk dapat terlindungi dalam sebuah komunitas dimana mereka memiliki waktu untuk berkembang. Ide utama dari *speciation* adalah mengelompokkan individu dalam populasi ke beberapa spesies berdasarkan topologinya. Pengelompokkan berdasarkan topologi dapat diselesaikan menggunakan *historical marking*. Jumlah dari *excess* dan *disjoint genes* dari sepasang *genomes* menjadi tolak ukur apakah kedua *genomes* memiliki kecocokan. Semakin *disjoint* kedua *genomes*, maka semakin sedikit pula mereka memiliki kesamaan sejarah dalam evolusinya, yang berefek pada semakin sedikitnya kecocokan yang mereka miliki.

2.4 Encoding

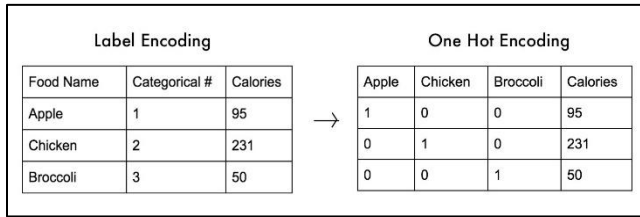
Pre-processing adalah metode untuk menyiapkan data sebelum diberikan kepada suatu model. *Encoding* merupakan bagian dari *pre-processing* yang bertujuan untuk merepresentasikan data agar dapat dimengerti oleh model.

Machine Learning tidak bisa bekerja dengan data bertipe kategori secara langsung, data bertipe kategori harus diubah terlebih dahulu kedalam bentuk angka [3]. Pada penelitian ini, ada data yang berbentuk kategori, yaitu posisi setiap pemain. Karena model tidak dapat melakukan analisa terhadap posisi pemain yang berbentuk kategori, seperti *striker*, *midfielder*, dan *defender*, data ini harus diencode terlebih dahulu.

Ada 2 metode *encoding* yang dapat dilakukan untuk mengubah data berbentuk kategori ke angka, yaitu label *encoding* dan *one hot encoding*.

Label encoding merepresentasikan setiap kategori pada data kedalam suatu *integer* sehingga dapat dipahami oleh model. Namun, karena *output* yang dihasilkan metode ini berupa *integer*, maka *output* yang dihasilkan akan memiliki *natural ordered relationship*, yang berarti komputer secara otomatis akan memberikan bobot yang lebih besar kepada kategori yang memiliki

nilai *categorical* yang lebih besar, ini akan menjadi masalah pada data yang tidak memiliki *ordinal relationship*.



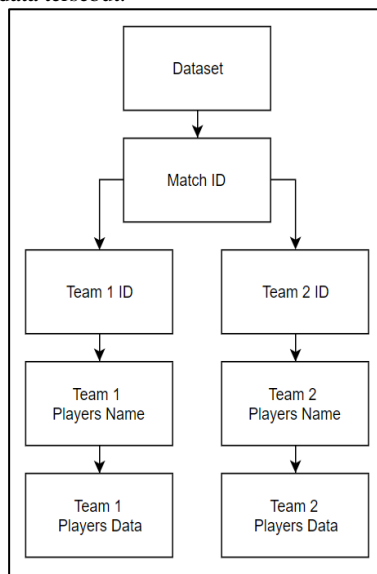
Gambar 6. Label encoding dan one hot encoding

Berbeda dengan *label encoding* yang menghasilkan nilai *categorical* berupa *integer*, *one hot encoding* melakukan *binarization* pada data. Metode ini merupakan metode yang tepat untuk data yang tidak memiliki *ordinal relationship*, seperti posisi pemain pada pertandingan sepak bola. Misalnya, jika menggunakan label encoding, akan ada posisi tertentu yang memiliki bobot lebih tinggi dari posisi lainnya, padahal setiap posisi memiliki perannya masing-masing dan tidak memiliki *ordinal relationship*.

3. DESAIN SISTEM

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

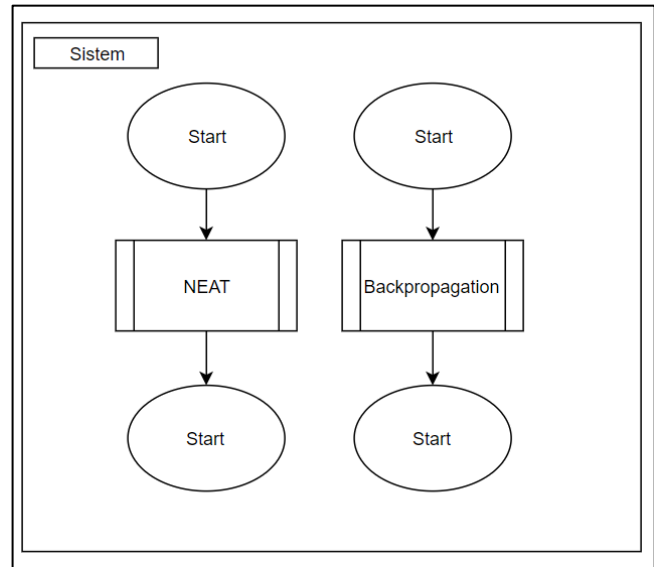


Gambar 7. Struktur dataset yang akan digunakan pada penelitian ini

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. Setelah proses *learning* dilakukan,

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

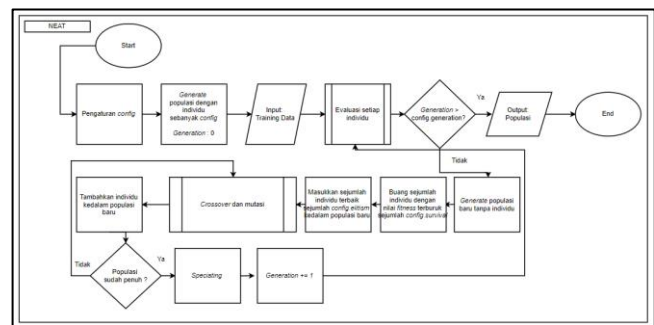
3.2 Desain Sistem



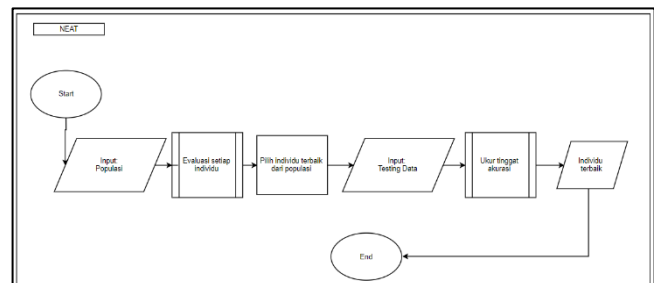
Gambar 8. Flowchart sistem secara umum

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.



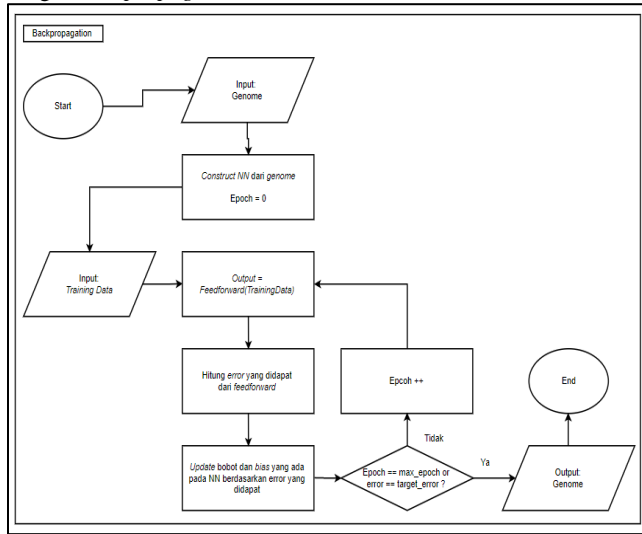
Gambar 9. Flowchart dari proses training pada NEAT



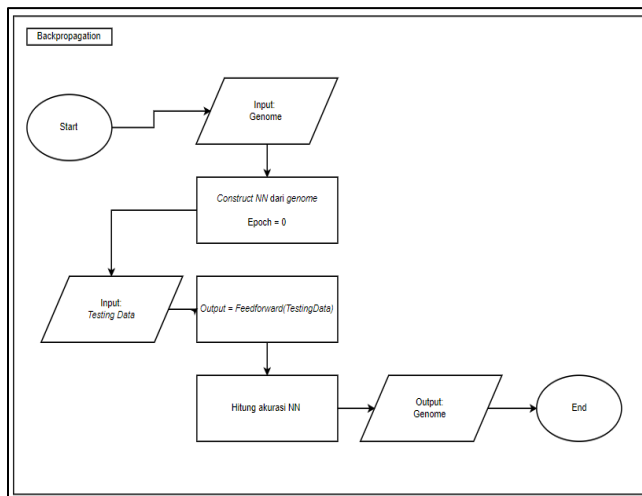
Gambar 10. Flowchart dari proses testing pada NEAT

3.2.1 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*.



Gambar 11. Flowchart dari proses *training backpropagation*



Gambar 12. Flowchart dari proses *testing backpropagation*

4. PENGUJIAN SISTEM

4.1 PENGUJIAN NEAT

Pengujian NEAT pada penelitian ini akan dibagi kedalam 3 tahap berdasarkan *feature* yang digunakan. Pembagian ini bertujuan untuk melihat pengaruh dari *feature* yang digunakan terhadap akurasi dari NEAT. Pada setiap tahap, akan dicoba berbagai konfigurasi untuk mendapatkan hasil yang optimal. Pengujian yang dilakukan meliputi proses *training*, yang bertujuan untuk mencari individu terbaik, dan proses *testing*, untuk mengukur akurasi dari individu yang dihasilkan oleh proses *training*.

Tabel 1. *Features* yang digunakan pada setiap tahap pengujian

Tahap	Features
-------	----------

Tahap 1	<i>Rating</i> pemain
Tahap 2	<i>Rating</i> pemain + <i>rating team</i>
Tahap 3	<i>Rating</i> pemain + <i>rating team</i> + posisi pemain

Tabel 2. Hasil Pengujian NEAT

Pengujian	Akurasi Hasil Pertandingan	Akurasi Skor Pertandingan
Tahap 1 Pengujian 1	107 / 380 (28%)	32 / 380 (8%)
Tahap 1 Pengujian 2	273 / 380 (71%)	85 / 380 (22%)
Tahap 1 Pengujian 3	300 / 380 (78%)	125 / 300 (32%)
Tahap 2 Pengujian 1	302 / 380 (79%)	136 / 380 (35%)
Tahap 2 Pengujian 2	305 / 380 (80%)	136 / 380 (35%)
Tahap 3 Pengujian 1	260 / 380 (68%)	90 / 380 (23%)
Tahap 3 Pengujian 2	194 / 380 (51%)	47 / 380 (12%)
Tahap 3 Pengujian 3	223 / 380 (52 %)	46 / 380 (12%)
Tahap 3 Pengujian 4	299 / 380 (78.6 %)	131 / 380 (34 %)

Selain *feature*, *fitness function* juga memiliki peranan yang sangat penting pada NEAT. Dapat dilihat pada Tahap 1 Pengujian 1 dan Tahap 1 Pengujian 2, terjadi peningkatan akurasi yang signifikan.

Selain *fitness function* yang digunakan, konfigurasi juga memiliki peranan penting ketika proses NEAT berjalan. Pentingnya konfigurasi dapat dilihat dari Tahap 1 Pengujian 2 dan Tahap 1 Pengujian 3. Dengan menggunakan *feature* yang sama, yaitu *rating* pemain, terjadi peningkatan akurasi dari 71% untuk akurasi prediksi hasil pertandingan dan 22% untuk akurasi prediksi skor yang dihasilkan oleh Tahap 1 Pengujian 2, menjadi 78% dan 32% yang dihasilkan oleh Tahap 1 Pengujian 3.

Penambahan 2 *feature* baru, yaitu *rating team* juga ikut meningkatkan tingkat akurasi dari sebuah *network*, walaupun tidak signifikan.

Pada Tahap 3, posisi pemain ikut menjadi *feature* pada proses NEAT. Dari 3 pengujian yang mengaplikasikan *encoding* kepada posisi masing-masing pemain dan 1 pengujian yang menggunakan total dari posisi pemain pada setiap sektor, hasil terbaik diperoleh

pada Tahap 3 Pengujian 4 yang menggunakan total posisi pemain pada tiap sektor. Pengujian ini menghasilkan tingkat akurasi sebesar 78% untuk akurasi hasil pertandingan dan 34% untuk akurasi skor.

Tingkat akurasi terbaik pada pengujian NEAT dihasilkan pada Tahap 2 Pengujian 2. Dengan sedikit melakukan perubahan konfigurasi yang digunakan pada Tahap 2 Pengujian 1, Tahap 2 Pengujian 2 mampu menghasilkan akurasi yang lebih baik, yaitu sebesar 81% untuk akurasi prediksi hasil pertandingan, dan 42% untuk akurasi prediksi skor.

4.2 Pengujian Backpropagation

Setelah proses NEAT selesai dijalankan, proses selanjutnya adalah *backpropagation*. Proses *backpropagation* bertujuan untuk mengoptimasi *weight* dari *network* yang dihasilkan oleh NEAT.

Tabel 3. Hasil pengujian *backpropagation*

Pengujian	Epoch	Akurasi Hasil Pertandingan	Akurasi Skor Pertandingan	Total Waktu
Tahap 1 Pengujian 1	100.000	291 / 380 (76%)	94 / 380 (24%)	264 detik
Tahap 1 Pengujian 2	100.000	289 / 380 (76%)	134 / 380 (35%)	234 detik
Tahap 1 Pengujian 3	44.000	301 / 380 (79%)	155 / 380 (40 %)	181 detik
Tahap 2 Pengujian 1	100.000	303 / 380 (79 %)	167 / 380 (44 %)	355 detik
Tahap 2 Pengujian 2	100.000	308 / 380 (81%)	161 / 380 (42%)	482 detik
Tahap 3 Pengujian 1	100.000	299 / 380 (78%)	153 / 380 (40%)	789 detik
Tahap 3 Pengujian 2	100.000	306 / 380 (80%)	122 / 380 (32%)	2269 detik
Tahap 3 Pengujian 3	100.000	304 / 380 (80%)	162 / 380 (42%)	1199 detik

Tahap 3 Pengujian 4	100.000	310 / 380 (81.5%)	185 / 380 (48%)	656 detik
---------------------	---------	----------------------	--------------------	-----------

Semua pengujian *backpropagation* dijalankan dengan *learning rate* sebesar 0.0001 dan selama 100.000 *epoch*, kecuali untuk Tahap 1 Pengujian 3 karena terjadinya *overfitting* pada *epoch* 44.000 – 45.000 yang menyebabkan proses *backpropagation* terpaksa dihentikan.

Setelah *backpropagation* selesai dijalankan, pengujian yang menghasilkan *network* dengan akurasi terbaik berubah. Tahap 3 Pengujian 4 mengalahkan Tahap 2 Pengujian 2 yang sebelumnya menjadi pengujian dengan akurasi terbaik.

Akurasi yang dihasilkan oleh Tahap 3 Pengujian 4 meningkat dari yang sebelumnya 78% untuk akurasi hasil pertandingan dan 34% akurasi skor pertandingan, menjadi 81% dan 48%. Akurasi yang dihasilkan oleh pengujian ini mengalahkan semua akurasi dari semua pengujian yang telah dilakukan.

Peningkatan paling signifikan pada *network* terjadi pada Tahap 1 Pengujian 1. *Network* yang dihasilkan oleh Tahap 1 Pengujian 1 pada awalnya hanya memiliki akurasi sebesar 28% dan 8%. Namun setelah proses *backpropagation* diaplikasikan pada pengujian tersebut, tingkat akurasi mengalami kenaikan menjadi 76% dan 24%. Hal ini membuktikan bahwa walaupun tingkat akurasi yang dihasilkan oleh *network* dari proses NEAT rendah, struktur yang dimiliki *network* itu sudah cukup baik.

4.3 Pengujian Real Life

Pada data pertandingan *real*, tidak diketahui berapa *rating* pemain yang akan bermain pada pertandingan tersebut. Untuk memprediksi *rating* pemain yang akan bermain, akan dicoba beberapa cara, yaitu rata-rata 5 pertandingan terakhir, rata-rata 10 pertandingan terakhir.

Selain itu, akan diuji juga rata-rata 5 dan 10 pertandingan terakhir yang sudah dinormalisasi. Normalisasi dilakukan dengan cara mengalikan *rating* pemain dengan *rating team* lawan, lalu dibagi dengan *rating team* dari pemain itu sendiri.

Agar tersedianya cukup data, pengujian *real life* dimulai dengan pertandingan dari pertandingan ke-10 untuk rata-rata 5 pertandingan terakhir, dan pertandingan ke-15 untuk rata-rata 10 pertandingan terakhir.

Tabel 4. Hasil pengujian *real life*

Metode Prediksi Rating	Akurasi Hasil Pertandingan	Akurasi Skor Pertandingan
Rata-rata 5 pertandingan terakhir	124 / 280 (44%)	28 / 280 (10%)
Rata-rata 5 pertandingan terakhir dengan normalisasi	78 / 280 (27.8%)	25 / 280 (8.9%)

Rata-rata 10 pertandingan terakhir	92 / 230 (40%)	21 / 230 (9%)
Rata-rata 10 pertandingan terakhir dengan normalisasi	63 / 230 (26.5%)	20 / 230 (8.6%)

Hasil pengujian *real life* sangat memiliki akurasi yang sangat rendah jika dibandingkan dengan hasil pengujian *backpropagation* maupun hasil pengujian NEAT yang menggunakan *rating* yang sebenarnya.

Hal ini disebabkan oleh *rating* pemain yang tidak konsisten pada setiap pertandingan sehingga sulit untuk diprediksi menggunakan rata-rata.

5. KESIMPULAN DAN SARAN

5.1 Kesimpulan

Dari hasil seluruh pengujian yang telah selesai, dapat diambil kesimpulan sebagai berikut, antara lain:

- *Features* terbaik pada proses NEAT adalah *rating* pemain dengan *rating team*.
- Penambahan *rating* team sebagai *features* sedikit meningkatkan tingkat akurasi yang dihasilkan oleh NEAT jika dibandingkan *rating* pemain saja yang menjadi *feature*.
- Akurasi tertinggi yang dihasilkan oleh NEAT didapatkan pada Tahap 2 Pengujian 2 dengan tingkat akurasi mencapai 80% pada prediksi hasil pertandingan dan 35% pada prediksi skor pertandingan.
- Penambahan posisi pemain sebagai *features* membuat proses *training* pada NEAT menjadi jauh lebih lama dan tingkat akurasi yang dihasilkan juga lebih rendah jika dibandingkan dengan *rating* pemain atau *rating* pemain dan *rating team* yang digunakan sebagai *features*.
- Penggunaan *encoding* pada posisi setiap pemain pada Tahap 3 justru memberikan efek negatif, yaitu akurasi yang lebih rendah dan waktu *training* yang lebih lama.
- Penggunaan jumlah posisi pemain pada tiap sektor yang dilakukan pada Tahap 3 Pengujian 4 menghasilkan tingkat akurasi yang lebih baik dan waktu *training* yang lebih singkat daripada melakukan *encoding* pada masing-masing posisi pemain.
- Dalam beberapa kasus, struktur *network* yang dihasilkan NEAT sudah cukup baik, namun *weight* yang tidak optimal menyebabkan tingkat akurasi yang rendah. Hal ini dibuktikan oleh proses *backpropagation*.
- *Backpropagation* mampu meningkatkan akurasi pada semua pengujian yang dilakukan.
- Akurasi tertinggi yang dihasilkan setelah *backpropagation* berasal dari Tahap 3 Pengujian 4, dengan tingkat akurasi sebesar 81.5% pada prediksi hasil pertandingan, dan 48% pada prediksi skor pertandingan.
- Tahap 3 Pengujian 4, yang keluar sebagai pengujian dengan akurasi terbaik, menggunakan *rating team*, *rating* pemain, dan jumlah dari posisi pemain dari setiap sektor (*defender*, *midfielder*, dan *striker*).
- Peningkatan akurasi terbaik setelah proses *backpropagation* selesai terjadi pada Pengujian 1 Tahap

1, dimana akurasi hasil pertandingan meningkat dari 28% menjadi 76%.

- *Rating* pemain yang tidak konsisten menyebabkan pengujian *real life* yang menggunakan metode pencarian *rating* dengan rata-rata menghasilkan tingkat akurasi yang rendah.

5.2 Saran

Saran yang dapat diberikan untuk mengembangkan skripsi ini lebih lanjut antara lain:

- Penambahan dataset dari berbagai liga agar akurasi yang didapat bisa lebih tinggi lagi.
- Gunakan metode lain untuk memprediksi *rating* pemain, seperti *linear regression*.

6. Daftar Referensi

- [1] Aggarwal, C, C. 2018. *Neural Networks and Deep Learning: A Textbook*. Cham: Springer Nature.
- [2] Boudway, I. 2018. Soccer Is the World's Most Popular Sport and Still Growing. URI=<https://www.bloomberg.com/news/articles/2018-06-12/soccer-is-the-world-s-most-popular-sport-and-still-growing>.
- [3] Brownlee, J. 2017. *How to One Hot Encode Sequence Data in Python*. URI=<https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>.
- [4] Chen, D., Giles, C., Sun, G., Chen, H., Lee, Y., & Goudreau, M. 1993. Constructive learning of recurrent neural networks. *1993 IEEE International Conference on Neural Networks*.
- [5] Igiri, C. 2015. Support Vector Machine-Based Prediction System for a Football Match Result. *IOSR Journal of Computer Engineering (IOSR-JCE)*, 21-26.
- [6] Morse, G., & Stanley, K. 2016. Simple Evolutionary Optimization Can Rival Stochastic Gradient Descent in Neural Networks. *2016 Proceedings of the on Genetic and Evolutionary Computation Conference - GECCO '16*.
- [7] Negnevitsky, M. 2005. *Artificial Intelligence A Guide to Intelligent System*. Addison-Wesley Publishing Company, Inc.
- [8] Pappalardo, L., & Cintia, P. 2018. Quantifying the relation between performance and success in soccer. *Advances in Complex Systems*.
- [9] Simeone, O. 2018. A Very Brief Introduction to Machine Learning With Applications to Communication Systems. *IEEE Transactions on Cognitive Communications and Networking*, 648-664.
- [10] Stanley, K., & Miikkulainen, R. 2002. Evolving Neural Networks through Augmenting Topologies. *Evolutionary Computation*.

**Columns on Last Page Should Be Made As Close As
Possible to Equal Length**