

## Multilayer Perceptrons Dalam Memprediksi Kemenangan Pertandingan Sepak Bola UEFA EURO 2016

Alfaza Rangana<sup>1\*</sup>, Raymond Chandra Putra<sup>2</sup>, Wahyudin<sup>3</sup>

<sup>2</sup>Universitas Katolik Parahyangan, Indonesia, <sup>1,3</sup>Universitas Pendidikan Indonesia, Indonesia

<sup>1</sup>[alfaza@upi.edu](mailto:alfaza@upi.edu), <sup>2</sup>[raymond.chandra@unpar.ac.id](mailto:raymond.chandra@unpar.ac.id), <sup>3</sup>[wahyudin\\_sanusi@upi.edu](mailto:wahyudin_sanusi@upi.edu)



### Histori Artikel:

Diajukan: 6 November 2023

Disetujui: 10 November 2023

Dipublikasi: 14 November 2023

### Kata Kunci:

Multilayer Perceptron; K-Fold Cross Validation; UEFA EURO; Machine Learning; Tensorflow

### Digital Transformation

*Technology (Digitech) is an*

*Creative Commons License This*

*work is licensed under a*

*Creative Commons Attribution-*

*NonCommercial 4.0 International*

*(CC BY-NC 4.0).*

### Abstrak

Sepak bola merupakan olahraga yang paling populer di dunia, tentunya, para penggemar menginginkan tim idola mereka untuk menang, namun seringkali prediksi mereka tidak didasari oleh data yang akurat. Maka dari itu, penelitian ini berguna dalam memprediksi kemenangan sepak bola terutama pada UEFA EURO 2016 dengan menggunakan data yang akurat, dan prediksi dilakukan menggunakan salah satu metode dalam Machine Learning, yaitu Jaringan saraf Tiruan. Pada Jaringan Saraf Tiruan, model yang digunakan adalah Multilayer Perceptrons (MLP) dengan pembelajaran menggunakan backpropagation, library Tensorflow.Keras digunakan dalam pembuatan model MLP. Data statistik tim pada pertandingan UEFA EURO 2016 dikumpulkan terlebih dahulu melalui situs resmi UEFA dan dilakukan eksplorasi terhadap data tersebut. Prediksi dapat dilakukan dengan Membuat model pertama kali dan dilakukan evaluasi menggunakan K-Fold Cross Validation dan menghasilkan akurasi sebesar 50-60%, setelah dilakukan optimasi terhadap learning rate, jumlah epoch, dan cara mengatasi overfitting, model baru berhasil dibuat. Model baru yang dibuat menghasilkan akurasi sebesar 90-95% setelah di evaluasi menggunakan K-Fold Cross Validation. Setelah mendapatkan model, prediksi dapat dilakukan menggunakan model tersebut dan setelah hasilnya keluar, akurasi prediksi didapat 75%. Model yang digunakan sehingga tercipta akurasi sebesar 75% itu dengan menggunakan model MLP 8-7-3, dimana terdapat delapan unit pada input layer, tujuh unit pada hidden layer, dan tiga unit pada output layer.

## PENDAHULUAN

Sepak bola merupakan salah satu olahraga yang paling diminati oleh banyak orang di dunia (Das, 2021). Sepak bola merupakan olahraga yang dimainkan antara dua tim di lapangan berbentuk persegi panjang yang memiliki tiang gawang atau gawang di setiap ujungnya. Tujuan permainan ini adalah membawa dan memasukan bola melewati garis kotak gawang lawan. Di setiap pertandingan sepak bola, setiap pelatih atau manajer tim sepak bola pasti menginginkan agar timnya selalu menang setiap bertanding dengan tim lawannya. Namun, tidak mudah sebagai pelatih untuk selalu mengatur dan mengubah strategi yang awal dimiliki untuk beradaptasi dengan tim lawan, maka dari itu prediksi menggunakan data statistik tim mereka diperlukan. Jika hasil prediksi tidak memuaskan saat melawan suatu tim, maka pelatih dapat menyesuaikan strategi baru dalam melawan tim tersebut, prediksi ini sudah didasari oleh data yang kuat.

Pada suatu turnamen sepak bola, terdapat data statistik yang dapat dikumpulkan, diantaranya adalah berapa sering bola dikuasai oleh suatu tim, banyak tembakan corner, banyak pelanggaran, banyak free kick, skor menang/kalah/seri tim dan lain-lain. Dari data statistik tersebut, data dapat diolah dan digunakan sebagai dasar memprediksi kemenangan tim menggunakan kecerdasan buatan. Salah satu metode yang dapat digunakan adalah Jaringan saraf Tiruan.

Jaringan Saraf Tiruan merupakan teknik komputasi yang diciptakan untuk meniru kerja otak manusia dalam menyelesaikan suatu masalah. Pada tahun 1943, McCulloch (1990), seorang neurobiologist, dan Pitts, seorang ahli statistik menjelaskan bagaimana otak manusia bekerja dan seberapa sederhana neuron bekerja secara paralel untuk membuat keputusan berdasarkan sinyal masukan yang didapat. Seperti yang dikatakan Amer & Nizar, persamaan yang dimiliki oleh otak manusia dan Jaringan saraf Tiruan adalah kemampuan mereka dalam memproses data untuk mencari solusi berdasarkan hasil training dari data tersebut (Zayegh & Bassam, 2018).

Pada penelitian ini, dibuat model Jaringan saraf Tiruan menggunakan metode Multi-layer Perceptron dengan Backpropagation Learning untuk memprediksi turnamen sepakbola UEFA 2016. Multi-layer Perceptron adalah suatu model pada Jaringan saraf Tiruan Yang dapat digunakan salah satunya untuk prediksi, dimana model sebelum masuk ke lapisan selanjutnya. Lapisan selanjutnya merupakan lapisan tersembunyi, yang dimana pada lapisan ini masukan sebelumnya dapat menentukan hasil keluarannya dengan melihat nilai yang terbesar pada unit di lapisan ini. Nilai yang terbesar pada lapisan tersembunyi kemudian menuju lapisan terakhir yaitu lapisan

keluaran, dimana pada lapisan ini menghasilkan keluaran berdasarkan nilai pada lapisan tersembunyi sebelumnya. Setiap unit pada lapisan saling terhubung dengan semua unit pada lapisan selanjutnya. Tentunya setelah hasil dikeluarkan, hasil dievaluasi untuk memastikan apakah hasil prediksi sama dengan hasil sebenarnya, Backpropagation Learning membantu model untuk membawa hasil evaluasi tersebut sampai ke lapisan masukan lagi, untuk mengubah-ngubah nilai yang ada pada lapisan masukan berdasarkan nilai evaluasinya pada lapisan keluaran.

Turnamen UEFA EURO 2016 terdapat 24 tim sepakbola yang bertanding, dimana masing-masing tim memiliki delapan atribut yang dapat digunakan sebagai prediktor pada model Jaringan Saraf Tiruan yang dibuat. Model yang dibuat terdapat dua eksperimen dalam memilih prediktor yang digunakan sebagai input dalam model, banyaknya node pada hidden layer sehingga model yang dibuat dapat menghasilkan tingkat akurasi yang tinggi dari rancangan model awal dengan kesalahan seminimal mungkin. Model yang dibuat tersebut diharapkan dapat digunakan untuk turnamen sepakbola serupa lainnya. Model dan evaluasinya dibuat dengan bantuan library Tensorflow dan library Keras.

Penelitian serupa dapat dilihat pada publikasi paper oleh Huang, Kou-Yuan dan Chen, Kai-Ju (2011), dimana mereka memprediksi kemenangan turnamen WORLD CUP 2006 dengan menggunakan model MLP juga dengan tingkat akurasi yang dicapai adalah 75%. Sergei Anfilets, Sergei Bezobrazov dan Vladimir Golovko (2020) memprediksi kemenangan turnamen English Premier League dengan akurasi 61,14%. Marios-Christos dan rekan (2022) menggunakan lima model Convolutional Neural Network dan Transfer Learning untuk memprediksi kemenangan turnamen Greek League, dan menggunakan model yang dibuatnya terhadap turnamen lain seperti English Premier League dan Dutch Eredivisie. Penelitian tersebut mendapatkan akurasi 55,86% untuk model DenseNet201, 50,62% untuk model InceptionV3, MobileNetV2 memiliki akurasi 55,86%, 53,86% menggunakan model ResNet101V2 dan terakhir model MobileNetV2 20-layer train dengan akurasi 58,27%. Gabriel, Aline and João (2019) membuat framework dimana dalam framework tersebut menjelaskan dataset dan model seperti apa yang cocok digunakan untuk memprediksi kemenangan dalam pertandingan sepak bola. World Cup 2018 juga diprediksi dalam penelitian oleh Amr Hassan et al. (2020) menggunakan Radial Basis Function (RBF) dan mendapatkan akurasi 72.7% untuk loss dan 83.3% untuk win.

Penelitian ini dilakukan dengan tujuan membangun model MLP dengan menentukan banyaknya *layer* pada model dan *training samples* yang digunakan untuk memprediksi turnamen UEFA EURO 2016. Beberapa fitur dalam Tensorflow digunakan untuk mengevaluasi model MLP yang dibuat. Model yang telah dibuat digunakan kedalam dua eksperimen, dimana pada eksperimen 1, model menggunakan *training samples* yang memperhatikan jumlah kemenangan pada pertandingan sebelumnya. Sedangkan untuk eksperimen 2, prediksi dilakukan menggunakan *training samples* tanpa memperhatikan jumlah kemenangan pada pertandingan sebelumnya.

## STUDI LITERATUR

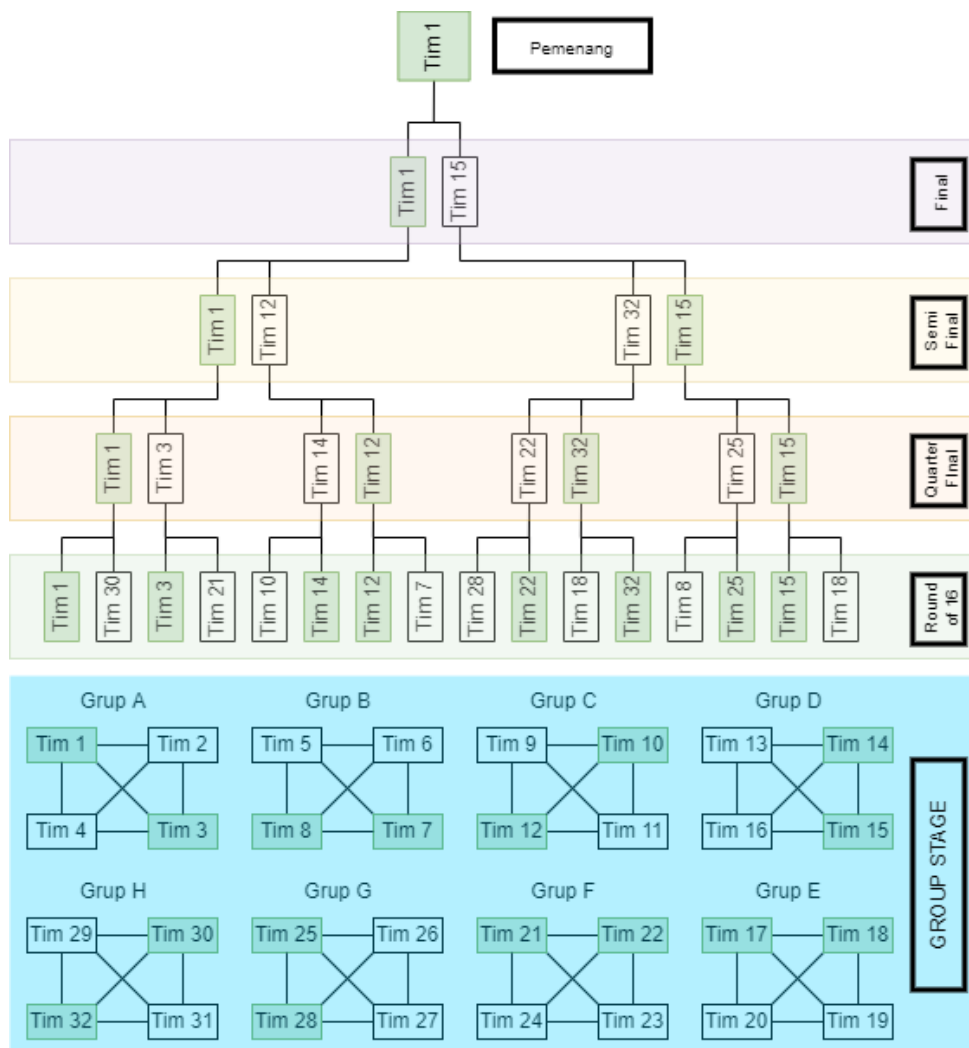
### *Sepak Bola dan Alur Pertandingan Round-robin*

Sepak bola merupakan permainan di mana dua tim yang terdiri dari 11 pemain, menggunakan bagian tubuh mereka kecuali tangan dan lengan mereka untuk memasukan bola ke gawang tim lawan. Hanya penjaga gawang yang diizinkan untuk memegang bola dan hanya boleh melakukannya di dalam area penalti di sekitar gawang. Tim yang mencetak lebih banyak gol merupakan tim yang dinyatakan menang (Walvin, 2000). Sepak bola adalah permainan bola paling populer di dunia dilihat dari jumlah peserta dan penonton. Aturannya yang sederhana membuat olahraga ini dapat dimainkan hampir di mana saja dan kapan saja, mulai dari lapangan resmi (lapangan) hingga gimnasium, jalanan, taman bermain sekolah, taman, atau pantai. Badan pengatur sepak bola dunia, the Fédération Internationale de Football Association (FIFA), memperkirakan bahwa pada pergantian abad ke-21 ada sekitar 250 juta pemain sepak bola dan lebih dari 1,3 miliar orang tertarik pada sepak bola; pada tahun 2010 gabungan penonton televisi lebih dari 26 miliar menonton turnamen utama sepak bola, salah satunya adalah putaran final Piala Dunia empat tahunan yang diselenggarakan selama sebulan. Pertandingan pada EURO dilakukan menggunakan sistem round-robin, di mana peserta yang berupa tim bertanding dalam babak grup terlebih dahulu sebelum akhirnya masuk ke ajang utama.

Alur pertandingan dapat dilihat pada Gambar 1, di mana kotak berwarna hijau menandakan pemenang pada grup atau pasangan di babak tersebut. Penjelasan rinci dijelaskan sebagai berikut :

1. Babak penyisihan grup (Group Stage) dilakukan oleh seluruh peserta tim, di mana satu tim akan melawan semua tim pada grupnya masing-masing. Tergantung seberapa banyak yang mendaftar pada turnamen EURO, pada umumnya babak penyisihan grup dibagi menjadi enam hingga delapan grup. Masing-masing grup akan terdiri dari empat tim yang melawan satu sama lain dan memperoleh poin sebanyak mungkin berdasarkan menang, kalah dan seri dalam setiap pertandingannya. Untuk tim yang menang pada setiap pertandingan di babak ini akan diberikan tiga poin, tim yang seri satu poin dan tim yang kalah nihil. Setelah fase babak penyisihan grup selesai, seluruh poin dalam satu grup untuk masing-masing tim akan dijumlahkan. Dari seluruh grup, dipilih 16 tim dengan perolehan poin tertinggi untuk masuk ke

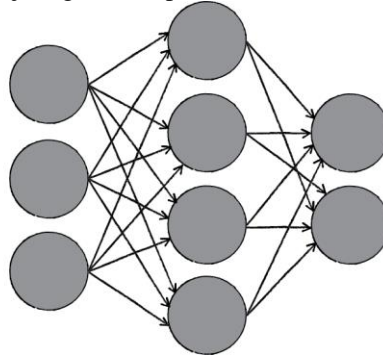
- babak selanjutnya.
2. Babak Round of 16 dilakukan oleh 16 tim yang telah lolos babak penyisihan grup, dan masing-masing grup akan secara acak melawan satu grup lainnya. Terdapat delapan pertandingan pada babak ini, di mana tim yang menang maju ke babak selanjutnya, dan yang kalah gugur. Pertandingan penalti dilakukan jika kedua tim memperoleh skor seri.
  3. Babak Quarter Final dilakukan oleh delapan tim yang telah menang melawan tim yang jadi pasangannya di babak Round of 16, dan masing-masing pemenang dari pasangan tersebut akan melawan tim pada grup pemenang dari pasangan sebelumnya saat babak Round of 16. Terdapat empat pertandingan pada babak ini, di mana tim yang menang maju ke babak selanjutnya, dan tim yang kalah gugur. Pertandingan penalti dilakukan jika kedua tim memperoleh skor seri.
  4. Babak Semifinal dilakukan oleh empat tim yang telah menang melawan tim yang jadi pasangannya di babak Quarter Final, dan masing-masing pemenang dari pasangan tersebut akan melawan tim pada grup pemenang dari pasangan sebelumnya saat babak Quarter Final. Terdapat dua pertandingan pada babak ini, di mana tim yang menang maju ke babak selanjutnya, dan tim yang kalah gugur. Pertandingan penalti dilakukan jika kedua tim memperoleh skor seri.
  5. Babak Final dilakukan oleh dua tim yang telah menang melawan tim yang jadi pasangannya di babak Quarter Final, dan masing-masing pemenang dari pasangan tersebut akan melawan tim pada grup pemenang dari pasangan sebelumnya saat babak Semi Final. Terdapat satu pertandingan pada babak ini, di mana tim yang menang maju ke babak selanjutnya, dan tim yang kalah gugur. Pertandingan penalti dilakukan jika kedua tim memperoleh skor seri, pemenang dinobatkan sebagai pemenang pada ajang pertandingan.



Gambar 1. Skema Round-robin

*Jaringan Saraf Tiruan*

Jaringan saraf adalah sistem komputasi yang pendekatannya untuk komputasi saraf menggunakan prosesor sederhana yang saling berhubungan. Jaringan saraf tidak mengikuti jalur linier, yang artinya informasi diproses secara kolektif, secara paralel di seluruh jaringan node (node, dalam hal ini, adalah neuron) (Shiffman, 2012). Ilmuwan komputer telah lama terinspirasi oleh otak manusia. Pada tahun 1943, Warren S. McCulloch, seorang ahli saraf, dan Walter Pitts, seorang ahli logika, mengembangkan model konseptual pertama dari jaringan saraf tiruan. Dalam penelitian mereka (McCulloch & Pitts, 1990), mereka menggambarkan konsep neuron, sel tunggal yang hidup dalam jaringan sel yang dapat menerima input, memproses input tersebut, dan menghasilkan output. Pekerjaan mereka, dan pekerjaan banyak ilmuwan dan peneliti berikutnya, tidak dimaksudkan untuk menggambarkan secara akurat bagaimana otak biologis bekerja. Sebaliknya, jaringan saraf tiruan dirancang sebagai model komputasi berdasarkan jaringan saraf pada otak manusia untuk memecahkan masalah.



Gambar 2. Jaringan Syaral Tiruan

Salah satu elemen kunci dari jaringan saraf adalah kemampuannya untuk belajar. Jaringan saraf bukan hanya sistem yang kompleks, tetapi sistem adaptif yang kompleks, yang artinya dapat mengubah struktur internalnya berdasarkan informasi yang mengalir melaluinya. Biasanya, ini dicapai melalui penyesuaian bobot. Pada Gambar 2, setiap garis mewakili hubungan antara dua neuron dan menunjukkan jalur aliran informasi. Setiap koneksi memiliki bobot, nomor yang mengontrol sinyal antara dua neuron. Jika jaringan menghasilkan keluaran yang diinginkan (yang didefinisikan nanti), tidak perlu menyesuaikan bobot. Namun, jika jaringan menghasilkan output yang tidak diinginkan atau salah, bisa dikatakan maka sistem akan beradaptasi, mengubah bobot untuk meningkatkan hasil selanjutnya. Shiffman (2012) menjelaskan bahwa jaringan saraf tiruan memiliki kemampuan untuk belajar, terdapat beberapa strategi untuk belajar diantaranya :

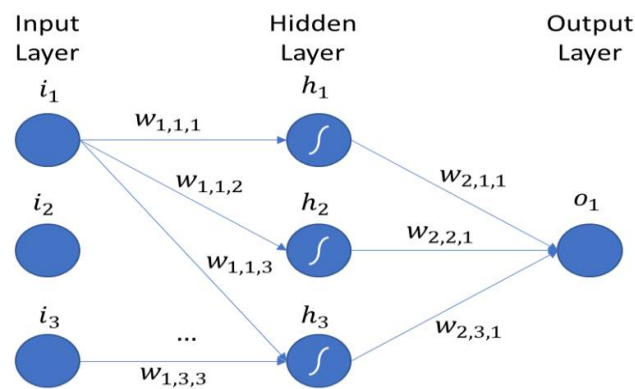
1. **Supervised Learning:** Pada dasarnya, sebuah strategi yang melibatkan seorang pengajar yang lebih cerdas daripada jaringan itu sendiri. Sebagai contoh, mari ambil contoh pengenalan wajah. Pengajar menampilkan sejumlah wajah kepada jaringan, dan pengajar sudah mengetahui nama yang terkait dengan setiap wajah. Jaringan membuat tebakan-tebakannya, kemudian pengajar memberikan jawaban-jawaban kepada jaringan. Jaringan kemudian dapat membandingkan jawabannya dengan yang "benar" yang diketahui dan melakukan penyesuaian berdasarkan kesalahannya. Jaringan saraf pertama kami dalam bagian berikutnya akan mengikuti model ini.
2. **Unsupervised Learning:** Digunakan ketika tidak ada contoh set data dengan jawaban yang diketahui. Bayangkan mencari pola tersembunyi dalam satu set data. Salah satu penggunaannya adalah pengelompokan, yaitu membagi sekelompok elemen ke dalam kelompok-kelompok berdasarkan pola yang tidak diketahui.
3. **Reinforced Learning:** Sebuah strategi berdasarkan pengamatan. Bayangkan seekor tikus kecil yang berlari melalui sebuah labirin. Jika dia belok ke kiri, dia mendapatkan potongan keju; jika dia belok ke kanan, dia mendapatkan sedikit kejutan. Dengan asumsi ini, tikus tersebut akan belajar dari waktu ke waktu untuk berbelok ke kiri. Jaringan sarafnya membuat keputusan dengan hasil (belok kiri atau kanan) dan mengamati lingkungannya (lezat atau sakit). Jika pengamatan adalah negatif, jaringan dapat menyesuaikan bobot-bobotnya untuk membuat keputusan yang berbeda di waktu berikutnya. Pembelajaran dengan penguatan umumnya digunakan dalam robotika.

Jaringan Saraf Tiruan terdiri lebih dari satu jenis, di mana salah satu contohnya adalah Multi-layer Perceptrons (MLP). MLP merupakan teknik pembelajaran mesin yang tergolong sederhana. Contoh MLP dapat dilihat pada Gambar 3 (Han, Kamber & Pei, 2011). Pada Gambar 3, terdapat tiga komponen utama, yaitu input layer, hidden layer, dan output layer dengan penjelasan masing-masing komponen sebagai berikut :

1. **Input Layer:** adalah lapisan yang dapat menerima data dengan format vektor atau tensor, di mana jumlah elemennya sesuai dengan jumlah prediktor yang diproses. Masing-masing elemen vektor ( $x_i$ )

dihubungkan ke setiap prediktor. Jika terdapat delapan atribut prediktor, maka terdapat delapan elemen vektor pada input layer.

2. **Hidden Layer:** adalah lapisan yang dapat terdiri dari satu atau bahkan lebih lapis. Masing-masing lapis berisikan sejumlah perceptron. Pada hidden layer (Gambar 3) input masing-masing neuron terhubung ke elemen vektor ( $x_i$ ) pada input layer, sedangkan output tiap neuron terhubung ke output layer. Masing-masing hubungan dari neuron ke setiap hidden layer memiliki bobotnya tersendiri. Untuk mendapatkan jumlah hidden layer dan jumlah neuron yang tepat, biasanya dilakukan dari berbagai eksperimen. Pada tahap pelatihan, semua bobot pada neuron dihitung dari data pelatihan berulang kali sehingga akhirnya pelatihan menghasilkan MLP dengan nilai bobot-bobot tertentu.
3. **Output Layer:** adalah lapisan yang terdiri dari satu atau lebih neuron. Penentuan jumlah neuron biasanya didasarkan pada eksperimen juga. Contoh jika nilai yang diprediksi terdiri dari dua nilai (0 atau 1), maka pada output layer dapat digunakan satu perceptron yang menghasilkan nilai mendekati 0 atau 1, yang nanti dapat dibulatkan.



Gambar 3. Jaringan Saraf Tiruan - Multilayer Perceptrons

Pada Gambar 3, terdapat jaringan yang saling terhubung antara neuron satu dengan neuron lainnya, itulah MLP. Setiap jaringan tersebut memiliki bobot, yang dimana pada gambar bobot tersebut dijelaskan dengan simbol  $w(x, y, z)$ , dimana  $x$  menandakan bobot berasal dari lapisan berapa pada jaringan,  $y$  menandakan bobot dimiliki oleh neuron berapa pada lapisan tersebut, dan  $z$  menandakan unit tujuan bobot dikirim.

Multi-layer Perceptrons dapat digunakan untuk klasifikasi data ataupun regresi, maka MLP perlu dilatih terlebih dahulu dengan menggunakan data pelatihan atau training. Pada MLP, tujuan dari pelatihan data tersebut untuk menghitung dan memperbaiki bobot yang ada pada hidden layer dan output layer.

#### FeedForward dan BackPropagation

Jaringan saraf tidak bisa prediksi apapun tanpa belajar terlebih dahulu caranya. Untuk melatih jaringan saraf agar menghasilkan output sesuai yang diharapkan, digunakanlah metode pembelajaran yang bersifat supervised learning. Dengan metode ini, jaringan diberikan input yang di mana jawabannya sudah diketahui. Dengan cara ini, jaringan dapat mengetahui apakah prediksi benar atau salah. Jika salah, jaringan dapat belajar dari kesalahannya dan menyesuaikan bobotnya (Shiffman, 2012). Proses dari Feedforward dan BackPropagation dapat dijelaskan sebagai berikut :

1. Input layer menerima masukan sebanyak prediktor yang digunakan untuk diprediksi.
2. Neuron memproses masukan dan menghasilkan output.
3. Output disesuaikan dengan hasil yang sebenarnya untuk dicari kesalahan atau error nya.
4. Perceptron menyesuaikan semua bobot sesuai dengan kesalahan yang ditemukan.
5. Kembali ke langkah pertama hingga output sudah tidak ditemukan kesalahan lagi atau iterasi sudah mencapai batas maksimal yang ditentukan.

Langkah 1 hingga 4 dapat dikemas dalam suatu fungsi. Namun, sebelum dapat dijadikan sebuah fungsi, model harus mendefinisikan kesalahan yang dilakukan oleh perceptron. Kesalahan perceptron dapat didefinisikan sebagai perbedaan antara jawaban yang diinginkan dan hasil prediksinya

$$Loss Function = (y - f(x))^2 \quad (1)$$

Persamaan 1 berfungsi untuk menyesuaikan bobot pada jaringan saraf untuk sampai pada jawaban yang benar. Pada persamaan tersebut, kesalahan dihitung menggunakan Squared Error Loss, dimana  $y$  menandakan hasil sebenarnya,  $f(x)$  merupakan hasil prediksi. Seperti yang dijelaskan pada Tabel 1, jika neuron memprediksi jawaban dengan benar, maka prediksinya sama dengan output yang diinginkan dan errornya adalah 0. Jika

jawaban yang benar adalah -1 dan hasil prediksi +1, maka errornya adalah -4. Jika jawaban yang benar adalah +1 dan hasil prediksi -1, maka kesalahannya adalah +4.

Tabel 1

Tabel contoh perhitungan kesalahan pada output

Label	Prediksi	Kesalahan
-1	-1	0
-1	+1	-4
+1	-1	+4
+1	+1	0

Kesalahan adalah faktor penentu bagaimana bobot pada perceptron harus disesuaikan. Untuk bobot tertentu, apa yang ingin dihitung adalah perubahan bobot, atau sering disebut dengan  $\Delta weight$  (2).

$$weight\ baru = weight + \Delta weight \quad (2)$$

di mana  $\Delta weight$  adalah

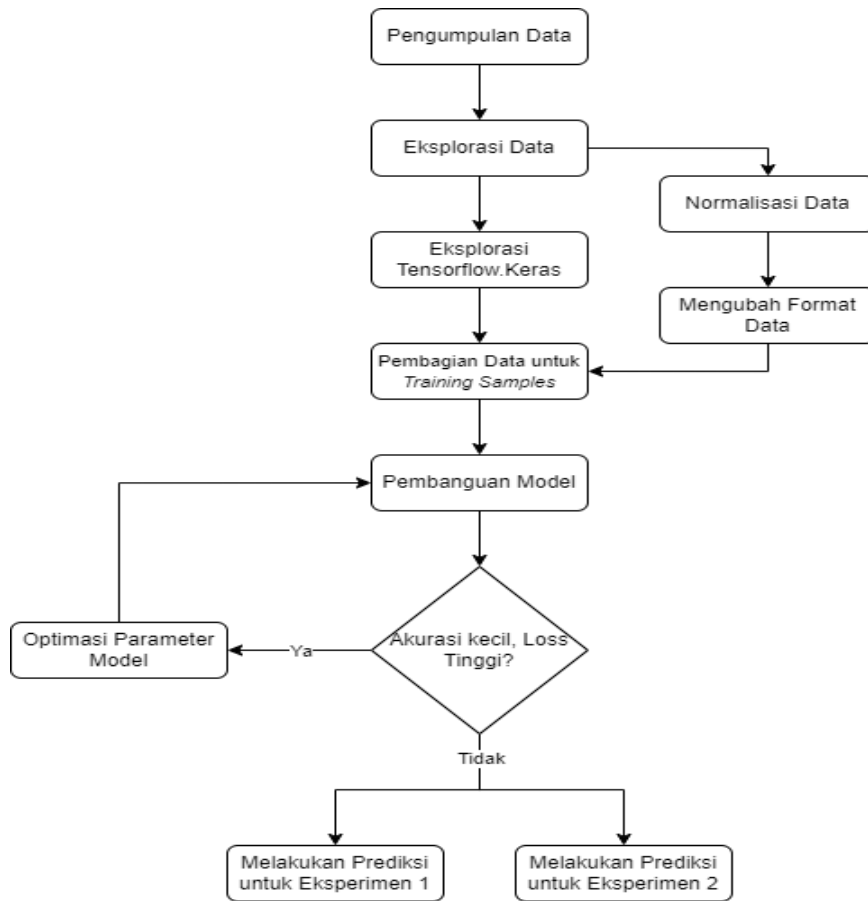
$$\Delta weight = (Loss\ Function) * Input * Learning\ Constant \quad (3)$$

Persamaan 3 terdapat variabel tambahan di mana variabel ini dapat mengendalikan banyaknya perubahan yang terjadi terhadap bobot. Di jaringan saraf, variabel tersebut disebut learning constant. Patut diperhatikan bahwa semakin tinggi learning constant, maka perubahan pada bobot dapat semakin tinggi, sebaliknya jika learning constant bernilai kecil, bobot akan berubah secara perlahan dan waktu training yang dibutuhkan akan cukup lama. Model sudah mempelajari bagaimana dengan supervised learning dapat mengetahui error yang terjadi pada output, namun jika dilihat pada Gambar 3, input layer tidak memiliki hubungan langsung dengan output layer; Input layer harus melalui layer yang bernama hidden layer. Jika perhitungan sedang berada di hidden layer, maka akan mudah mengetahui bobot mana yang mempengaruhi error pada output dikarenakan hubungan langsung yang terjadi antara hidden layer dengan output layer. Maka dari itu, backpropagation diciptakan supaya bisa mengatur bobot yang ada antara input layer dengan hidden layer. MLP mengetahui cara perhitungan error yang ada pada output layer untuk nantinya digunakan untuk mengubah bobot yang ada antara hidden layer dengan output layer. Untuk mengatur bobot antara input layer dengan hidden layer, pada hidden layer perhitungan bobot yang dibawa dari output layer digunakan kembali, itulah yang disebut dengan backpropagation.

### METODE

Pada penelitian ini, model regresi MLP dibuat berdasarkan hasil eksplorasi teknologi pada Tensorflow. Keras terhadap empat argumen yang penting, yaitu fungsi aktivasi, kelas membuat bobot acak, kelas mencari loss, dan kelas optimasi yang digunakan model. Setelah mengetahui argumen-argumen yang digunakan untuk kebutuhan model, maka model siap digunakan. Sebelum model digunakan, data dipersiapkan terlebih dahulu dengan membagi setiap data pelatihan untuk masing-masing babak dalam pertandingan UEFA EURO 2016. Setelah data dibagi, pemilihan prediktor dilakukan berdasarkan analisis yang dilakukan oleh Ulrich untuk eksperimen pertama, sedangkan eksperimen kedua prediktor digunakan semuanya. Setelah data disiapkan dan prediktor telah dipilih untuk kedua eksperimen, data hasil pelatihan digunakan untuk pemodelan MLP awal.

Setelah pemodelan awal dilakukan, model dievaluasi menggunakan K-Fold Cross Validation (teknik evaluasi yang membagi dua bagian pada dataset, yaitu bagian pelatihan dan bagian untuk di uji, dimana setiap kemungkinan dari pembagian tersebut dievaluasi dan dicari yang rata-ratanya berdasarkan banyaknya fold), dengan melihat hasil evaluasi pada model awal, perbaikan pada model dilakukan dengan mencari learning rate (dengan memanfaatkan fungsi callback pada model .fit tensorflow, dimana fungsi ini dapat menaikkan learning rate sedikit demi sedikit sehingga sampai kepada titik dimana akurasi sudah tidak bertambah lagi) dan cara mengatasi model yang overfitting. Model hasil perbaikan dibuat setelah menentukan learning rate terbaik dan cara mengatasi overfitting ditemukan, kemudian model dievaluasi lagi menggunakan K-Fold Cross Validation. Setelah mendapatkan akurasi yang bagus dengan loss yang lebih rendah dari model awal, model terbaru digunakan sebagai model regresi MLP untuk memprediksi kemenangan sepak bola pada pertandingan UEFA EURO 2016. Alur penelitian dapat dilihat pada Gambar 4.



Gambar 4. Flowchart Metode Penelitian

### Penyiapan Data

Pada dataset yang digunakan (*datasets\_uefa\_all.csv*), setiap tim memiliki delapan kolom yang dapat dijadikan prediktor, tentunya belum tentu semua prediktor tersebut dapat digunakan untuk menghasilkan hasil prediksi sesuai dengan hasil yang sebenarnya. Pada hasil wawancara terhadap Ulrich Forstner, salah satu pelatih nasional dalam Science and Education di Amerika yang dilakukan oleh Daniel dan Dominik (2018) dihasilkan prediktor-prediktor yang dapat membantu menilai kualitas suatu tim. Ulrich menjelaskan bahwa atribut yang dapat membantu dalam melihat kualitas suatu tim adalah goals, goal chances, penalty corners, entering the shooting circle and into the attack quarter, passes, dribbling, long corners, free hits, ball gains, dan ball losses. Prediktor-prediktor yang telah dijelaskan oleh Ulrich, diantaranya dapat direpresentasikan dengan atribut dalam dataset sebagai berikut:

1. Untuk prediktor goals dapat menggunakan atribut GoalsFor atau disingkat “gf” dalam dataset.
2. Untuk prediktor goal chances dapat menggunakan atribut ShotsOnTarget atau disingkat “sont” dalam dataset.
3. Untuk prediktor penalty corners dapat menggunakan atribut CornersTaken atau disingkat “ct” dalam dataset.
4. Untuk prediktor ball losses dapat menggunakan dua atribut ShotsOffTarget atau disingkat “sofft” dalam dataset.

Pada penelitian ini, dilakukan dua eksperimen dalam menggunakan Prediktor dan training samples sebagai input dalam model regresi MLP yang dibuat, yang pertama menggunakan hasil analisis yang dilakukan oleh Ulrich dalam pemilihan prediktornya, dan yang kedua menggunakan semua prediktor yang ada dalam dataset yang telah dijelaskan pada tabel 3.1. Training samples yang digunakan pada eksperimen 1 menggunakan pendekatan seperti yang dilakukan oleh Kou-Yuan Huang dan Kai-Ju Chen (2011) yang memilih data pada Data Frame hanya tim-tim yang mewakili tim yang menang, kalah, dan seri. Training samples pada eksperimen 2 menggunakan pendekatan yang hampir sama pada eksperimen 1, namun data yang digunakan tidak memiliki aturan dimana tim yang menang, kalah, dan seri saja yang digunakan sebagai training samples-nya. Nantinya masukan yang diterima oleh model merupakan rata-rata dari banyaknya pertandingan yang telah tim itu lakukan selama pertandingan UEFA berlangsung.

*Training Samples*

Prediktor yang telah dipilih kemudian dilatih sebelum dimasukkan kedalam model, tujuan pelatihan ini supaya model dapat menghasilkan prediksi yang secara umum sesuai untuk data baru. Dataset UEFA 2016, satu barisnya memiliki data statistik untuk dua tim, maka harus dipecah dulu menjadi dua baris berbeda, kemudian digabungkan kembali ke dalam satu DataFrame baru. Setelah data pelatihan siap digunakan kedalam model, perlu dicari terlebih dahulu berapa banyak node dan hidden layer yang digunakan dalam model regresi MLP. Model yang sudah dirancang tersebut kemudian di evaluasi dan jika tidak ada masalah terhadap model tersebut, model dapat digunakan untuk prediksi. Terdapat dua eksperimen pada penelitian ini, yang pertama model menggunakan training samples dengan analisis dari Ulrich, dan yang kedua menggunakan semua prediktor yang ada dalam dataset yang dapat dijadikan prediktor.

Pada penelitian ini, dilakukan dua eksperimen terhadap training samples yang nantinya digunakan sebagai input pada model regresi MLP yang dibuat. Eksperimen tersebut terdiri dari:

1. Eksperimen 1, Training Samples dipilih dengan melihat hasil pertandingan tim pada babak penyisihan grup, dimana pembagiannya dibagi berdasarkan :
  - Tim yang tidak pernah kalah(0 lose) dan menang minimal dua kali(2 win) dipilih sebagai data yang merepresentasikan tim yang menang.
  - Tim yang tidak pernah menang(0 win) dan kalah minimal dua kali(2 lose) dipilih sebagai data yang merepresentasikan tim yang kalah.
  - Tim yang tidak pernah mengalami kekalahan juga kemenangan(0 win dan 0 lose) dipilih sebagai data yang merepresentasikan tim yang seri.
2. Eksperimen 2, Training Samples menggunakan semua data statistik tim yang ada pada setiap babak pada pertandingan UEFA 2016 (tidak ada pemilihan tim untuk merepresentasikan suatu hal).

Pada saat pemilihan data untuk training samples dilakukan, seluruh baris yang ada pada dataset tidak digunakan secara sekaligus, dikarenakan untuk memprediksi babak tertentu tidak menggunakan data statistik yang belum pernah dimainkan. Contohnya untuk memprediksi babak tertentu pada babak round of 16, pemilihan data sebagai training samples hanya berasal dari data statistik babak sebelumnya, yaitu babak penyisihan grup, tidak mungkin menggunakan data statistik babak setelahnya, yaitu babak quarter finals, karena belum pernah terjadi.

Untuk memprediksi suatu babak, digunakan data statistik hasil pertandingan pada babak sebelumnya dan sejarah tim yang bertanding di pertandingan pada turnamen UEFA sebelum-sebelumnya. Maka dari itu, untuk mempermudah eksperimen, DataFrame dibagi lagi menjadi empat DataFrame berbeda, yaitu DataFrame untuk masing-masing babak dalam pertandingan.

*Training Samples untuk Eksperimen 1*

Dalam mencari tim mana saja yang dapat dijadikan training samples sesuai dengan ketentuan yang sudah ditentukan, maka harus dicari terlebih dahulu jumlah kemenangan, kekalahan dan seri masing-masing tim. Training samples dipilih dengan mengambil baris data pada Data Frame sesuai ketentuannya masing-masing yang dimana timnya merupakan hasil dari pencarian sebelumnya. Pencarian tersebut dilakukan dengan cara menghitung berapa banyak kemenangan, kekalahan dan seri setiap tim di babak penyisihan grup, kemudian dimasukkan kedalam variabel baru untuk nantinya dapat dilakukan penyaringan terhadap baris data di Data Frame yang tidak memenuhi ketentuan untuk training samples di eksperimen 1. Pada eksperimen ini, penyaringan menghasilkan 11 tim pada babak penyisihan grup, lima(5) tim pada babak round of 16, tiga(3) tim pada babak quarter finals, dan dua(2) tim pada babak semi finals, hasil dapat dilihat pada Tabel 2.

Tabel 2  
Tabel hasil pencarian tim untuk dijadikan training samples pada eksperimen 1.

Result	Match Type			
	Group Stage	Round of 16	Quarter Finals	Semi Finals
win (w)	France, Germany, Poland, Croatia Romania,	France, Germany	France	France
lose (l)	Russia, Ukraine Croatia - Germany	Croatia	-	Germany

	Czech, Sweden, Austria			
draw (d)	Portugal	Poland	Germany, Poland	-

Data statistik tim pada Tabel 2 kemudian dipecah lagi menjadi dua, data untuk training label prediktor dan data untuk training label target. Masing-masing babak menggunakan training samples yang berbeda untuk memprediksi pertandingan pada babak selanjutnya. Maka dari itu, data statistik yang telah dikumpulkan, dijadikan satu Data Frame baru sesuai dengan kebutuhan masing-masing babak. Setelah data dibagi menjadi dua, data untuk training dan data target untuk testing, pada DataFrame yang menyimpan data target masih tersimpan nilai w, l, dan d. Nilai tersebut masih berbentuk objek, dan tipe data tersebut tidak dapat digunakan sebagai data untuk testing, maka dari itu harus diubah dulu masing-masing nilai label menjadi suatu angka numerik supaya nantinya dapat dijadikan data untuk testing. Perubahan yang dilakukan dengan mengganti nilai w yang artinya menang dengan angka numerik 2, nilai d yang artinya seri dengan angka numerik 1, dan nilai l yang artinya kalah dengan angka numerik 0.

*Training Samples untuk Eksperimen 2*

Pada pemilihan data statistik tim untuk eksperimen kedua ini, caranya hampir mirip dengan yang dilakukan dalam mencari training samples untuk eksperimen 1, yang membedakannya hanyalah tidak dilakukan pencarian data statistik tim dengan ketentuan khusus, karena semua baris pada Data Frame digunakan pada eksperimen ini. Ketentuan untuk memprediksi babak selanjutnya menggunakan Data Frame pada babak sebelum-sebelumnya masih berlaku pada eksperimen kedua ini.

*Pembangunan Model MLP*

Setelah pengumpulan data dilakukan, model dapat mulai dirancang untuk digunakan sebagai model prediksi pertandingan UEFA 2016. Yang pertama, tentunya menentukan berapa banyak unit pada hidden layer. Pada artikel yang dipublikasikan Mirchandani dan Cao (1989), mereka membuat suatu teorema dalam menentukan banyaknya unit dalam hidden layer, yaitu banyaknya daerah yang terpisah (M) adalah suatu fungsi dari banyaknya node atau unit pada hidden layers (H), dan dimensi pada input (d), dimana dalam pembuktiannya fungsi  $M = 2^H$ .

$$M(H, d) = \sum C(H, k), \text{ dimana } C(H, k) = 0, H < k \quad (4)$$

Dapat dilihat bahwa maksimal data untuk training samples pada babak final dapat terdiri dari 288 baris data, dengan dimensi input d pada eksperimen 1 adalah 4 dikarenakan hanya terdapat 4 prediktor yang digunakan, sedangkan untuk eksperimen 2 adalah 8. Ketika jaringan pada MLP mempunyai dimensi input d sebanyak 4, maka  $M(H, d) = M(7, 4)$  dimana 27 dapat memiliki maksimal 128 daerah yang terpisah  $M(4)$ . Maka pada eksperimen 1 dan eksperimen 2 dapat menggunakan model regresi MLP dengan tujuh(7) node pada hidden layer nya atau disebut juga model regresi MLP 4-7-3 untuk eksperimen 1, dan model regresi MLP 8-7-3 untuk eksperimen 2.

Model regresi MLP dibuat berdasarkan eksplorasi teknologi di Tensorflow, dengan menerapkan 4-7-3(4 neuron pada lapisan masukan, 7 neuron pada lapisan tersembunyi dan 3 neuron) pada eksperimen 1, dan 8-7-3(8 neuron pada lapisan masukan, 7 neuron pada lapisan tersembunyi dan 3 neuron) pada eksperimen 2, model pun dibuat dengan menggunakan library Tensorflow.Keras. Data yang sudah dikumpulkan di training terlebih dahulu untuk dimasukkan kedalam model regresi MLPnya. Digunakan data training dan test untuk memprediksi round of 16 permainan sepak bola, dikarenakan setiap model memiliki data training dan test yang berbeda-beda, maka hanya ditampilkan yang memprediksi round of 16 saja, tetapi cara dan pengerjaannya sama dengan yang lain. model regresi MLP dibuat dengan menggunakan fungsi aktivasi ReLU hidden layernya, sedangkan output layer menggunakan aktivasi softmax.

*Evaluasi Model MLP*

Mengevaluasi model yang telah dibuat dapat menggunakan banyak cara yang ada, namun pada penelitian ini digunakan evaluasi dengan K-Fold Cross Validation, dimana K-Fold Cross Validation adalah salah satu teknik evaluasi yang membagi dataset menjadi dua bagian, bagian pertama untuk pelatihan dan bagian kedua untuk uji. Data yang digunakan untuk uji tidak dimasukkan ke tahap pelatihan supaya nantinya data yang dilatih dapat sekiranya memprediksi menggunakan data asli yang tidak digunakan pelatihan untuk pengujiannya.

```

Score per fold
-----
> Fold 1 - Loss: 1.020896553993225 - Accuracy: 46.666666865348816%
-----
> Fold 2 - Loss: 1.0266274213790894 - Accuracy: 50.0%
-----
> Fold 3 - Loss: 1.0461080074310303 - Accuracy: 42.85714328289032%
-----
> Fold 4 - Loss: 0.9627370834350586 - Accuracy: 50.0%
-----
> Fold 5 - Loss: 0.7668225169181824 - Accuracy: 78.57142686843872%
-----
> Fold 6 - Loss: 0.9099477529525757 - Accuracy: 71.42857313156128%
-----
> Fold 7 - Loss: 0.6334841847419739 - Accuracy: 92.85714030265808%
-----
> Fold 8 - Loss: 0.8786260485649109 - Accuracy: 57.14285969734192%
-----
> Fold 9 - Loss: 0.9556271433830261 - Accuracy: 64.28571343421936%
-----
> Fold 10 - Loss: 1.1345516443252563 - Accuracy: 21.42857164144516%
-----
Average scores for all folds:
> Accuracy: 57.523809522390366 (+- 19.182616194563014)
> Loss: 0.9335428357124329
    
```

Gambar 5. Evaluasi model MLP menggunakan K-Fold

Pengujian dilakukan sebanyak K, dimana K menandakan berapa banyak evaluasi dilakukan sehingga nanti dapat dicari rata-rata akurasi. Hasil evaluasi model dengan menggunakan K = 10 dan pembagian data pelatihan 80% dan data uji 20% dapat dilihat pada Gambar 5.

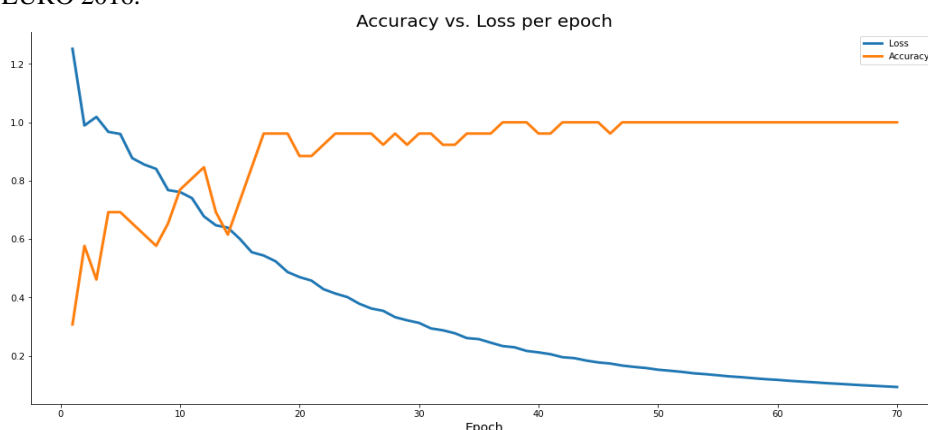
Dapat dilihat pada Gambar 5, bahwa akurasi pada model awal sangat buruk, berada pada 57% dengan loss sebesar 0.9. Model dapat diperbaiki dengan cara mencari learning rate yang bagus untuk digunakan dalam model. Model dapat dibuang overfitting ketika model memiliki validation loss yang terus meningkat, overfitting ini dapat ditangani dengan menggunakan fungsi Dropout pada Tensorflow.Keras.

#### Perbaikan Model MLP

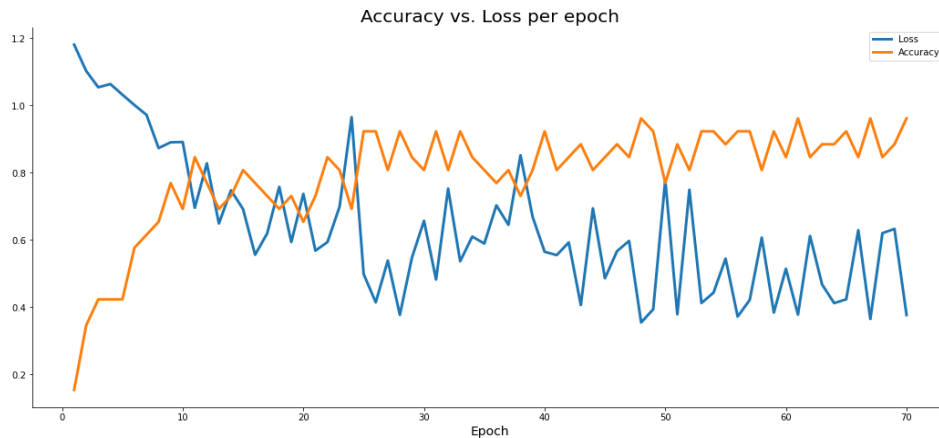
Dalam pencarian learning rate terbaik untuk model regresi MLP ini, dapat dicari dengan menggunakan salah satu parameter fungsi yang bernama callbacks pada method .fit punya Tensorflow.Keras. Fungsi callbacks menunjukkan accuracy dan loss dari suatu model dengan perlahan-lahan menaikkan learning rate-nya, dengan ini accuracy dan loss dapat divisualisasikan, hasil visualisasi penggunaan fungsi callbacks dapat dilihat pada Gambar 6 dan Gambar 7. Fungsi Dropout digunakan untuk menghindari *overfitting*, setelah fungsi Dropout digunakan, akurasi sudah tidak lagi konstan >= 1, dikarenakan Dropout menghilangkan beberapa neuron pada lapisan tersembunyi sehingga model memiliki kombinasi banyaknya neuron pada lapisan tersembunyi. Dapat dilihat pada Gambar 7, bahwa loss terendah yang dicapai berada pada epoch sekitar 60 hingga 70-an dengan learning rate sekitar 0.1359 .. Maka dari itu, model regresi MLP yang digunakan untuk memprediksi, learning rate-nya menggunakan 0.1359. Grafik akurasi dan loss pada model yang telah dioptimalisasi dan diimplementasi Dropout.

#### Evaluasi Perbaikan Model MLP

Model terbaru kemudian dievaluasi kembali menggunakan K-Fold Cross Validation. Hasil dapat dilihat pada Gambar 8, akurasi model tercatat sebesar 94% dengan loss sebesar 0.32. Dengan akurasi yang besar dan loss yang cukup kecil dari model sebelumnya, model hasil perbaikan dapat digunakan untuk prediksi kemenangan sepak bola UEFA EURO 2016.



Gambar 6. Grafik akurasi dan loss model awal



Gambar 7. Grafik akurasi dan loss pada model yang telah diimplementasikan Dropout

### HASIL

Setelah didapat model regresi MLP yang digunakan dan learning rate-nya, dengan menggunakan training samples yang sudah disiapkan sebelumnya, prediksi menggunakan model regresi MLP tersebut dilakukan. Prediksi dilakukan menggunakan dua eksperimen pada training samples. Data yang digunakan sebagai input adalah rata-rata data tim yang diprediksi untuk setiap atribut yang digunakannya. Hasil prediksi terdiri dari tiga node atau tiga kelas, dimana kelas 0 berartikan kalah, kelas 1 berartikan seri, dan kelas 2 berartikan menang. Angka yang ada pada masing-masing kelas nantinya merupakan seberapa besar kemungkinan tim tersebut menang, kalah atau seri pada pertandingannya.

Eksperimen 1 dengan menggunakan model regresi MLP 4-7-3, dimana model terdiri dari satu input layer, satu hidden layer, dan satu output layer. Untuk input layer memiliki empat node, tujuh node pada hidden layer, dan tiga node pada output layer. Epoch yang digunakan sebanyak 70 epoch, menghasilkan hasil prediksi dengan tingkat akurasi sebesar 50%. Sedangkan pada eksperimen 2, dengan menggunakan model regresi MLP 8-7-3 dengan banyak layer dan node untuk masing-masing layer yang sama dengan eksperimen 1, menghasilkan akurasi sebesar 75%. Hasil prediksi ini membandingkan hasil prediksi yang telah dilakukan oleh model regresi MLP pada masing-masing eksperimen dengan hasil sebenarnya, disebut juga sebagai supervised learning.

```

Score per fold
-----
> Fold 1 - Loss: 0.23428888618946075 - Accuracy: 100.0%
> Fold 2 - Loss: 0.49417707324028015 - Accuracy: 64.28571343421936%
> Fold 3 - Loss: 0.08077783137559891 - Accuracy: 100.0%
> Fold 4 - Loss: 0.2085922658443451 - Accuracy: 100.0%
> Fold 5 - Loss: 0.1585369110107422 - Accuracy: 92.85714030265808%
> Fold 6 - Loss: 0.6471205949783325 - Accuracy: 92.85714030265808%
> Fold 7 - Loss: 0.06671161949634552 - Accuracy: 100.0%
> Fold 8 - Loss: 1.1990946531295776 - Accuracy: 92.85714030265808%
> Fold 9 - Loss: 0.06519971042871475 - Accuracy: 100.0%
> Fold 10 - Loss: 0.14309795200824738 - Accuracy: 100.0%
-----
Average scores for all folds:
> Accuracy: 94.28571343421936 (+- 10.497813530977881)
> Loss: 0.3297597497701645
-----
Model sudah berhasil dibangun untuk digunakan prediksi
    
```

Gambar 8. Evaluasi model MLP setelah perbaikan menggunakan K-Fold

Pada penelitian ini, pertandingan dengan skor akhir bernilai seri tidak dimasukkan kedalam perhitungan akurasi. Setelah dilakukan banyak sekali iterasi, tidak pernah ditemukan kasus dimana kemungkinan menang kedua tim sama atau setidaknya selisihnya sangat kecil sehingga dapat dianggap pertandingan seri. Hasil prediksi dapat dilihat lebih rinci pada Tabel 3 untuk eksperimen 1 dan Tabel 4 untuk eksperimen 2. Simbol “w” mengartikan bahwa tim menang, simbol “l” artinya kalah, dan “d” artinya seri, sedangkan kolom “Benar?” mengartikan bahwa hasil perbandingan antara prediksi dan hasil sebenarnya sudah benar apa belum, simbol “-” mengartikan bahwa pertandingan tidak diperhitungkan.

Untuk eksperimen 1, dapat dilihat pada Tabel 3 bahwa pada pertandingan round of 16 akurasinya adalah 50%, untuk pertandingan quarter finals akurasinya adalah 50%, untuk pertandingan semi finals akurasinya adalah 100%, dan final akurasinya adalah 0%. Sedangkan untuk eksperimen 2, dapat dilihat bahwa pada Tabel 4 bahwa

pada pertandingan round of 16 akurasinya adalah 83%, untuk pertandingan quarter finals akurasinya adalah 50%, untuk pertandingan semi finals akurasinya adalah 100%, dan final akurasinya adalah 0%. Dengan adanya akurasi pada hasil prediksi di eksperimen 2 lebih tinggi dibandingkan eksperimen 1, ini membuktikan bahwa hasil analisis yang dilakukan oleh Ulrich, kurang tepat untuk diterapkan pada penelitian ini. Faktor lain bisa saja disebabkan oleh lemahnya representasi data hasil analisis dari Ulrich menggunakan prediktor yang ada pada dataset di penelitian ini.

Tabel 3  
Tabel hasil prediksi pada pada eksperimen 1.

Babak	Tim	Output dari Model			Prediksi	Hasil Nyata	Benar?	Akurasi
		Kelas 0	Kelas 1	Kelas 2				
round_of_16	switzerland	0.063792	0.6328	0.3033	Lose	d	-	3/6 (50%)
	poland	0.038636	0.4823	0.4790	Win	d	-	
	wales	0.038636	0.4823	0.4790	Win	w	Benar	
	north ireland	0.324366	0.6733	0.0022	Lose	l	Benar	
	croatia	0.038636	0.4823	0.4790	Win	w	Benar	
	portugal	0.324366	0.6733	0.0022	Lose	l	Benar	
	france	0.038636	0.4823	0.4790	Win	l	Salah	
	republic of ireland	0.239698	0.7509	0.0093	Lose	w	Salah	
	germany	0.038636	0.4823	0.4790	Win	l	Salah	
	slovakia	0.13411	0.7924	0.0734	Lose	w	Salah	
	hungary	0.157277	0.7972	0.0454	Lose	w	Salah	
	belgium	0.056951	0.5985	0.3444	Win	l	Benar	
	italy	0.06049	0.6168	0.3226	Lose	l	Benar	
	spain	0.038636	0.4823	0.4790	Win	w	Salah	
england	0.221265	0.7657	0.0130	Lose	w	Salah		
iceland	0.038636	0.4823	0.4790	Win	l	Salah		
quarter_final	poland	0.001255	0.1189	0.8797	Win	d	-	1/2 (50%)
	portugal	0.010178	0.3506	0.6391	Lose	d	-	
	wales	0.001511	0.1305	0.8679	Lose	w	Salah	
	belgium	0.000927	0.1022	0.8968	Win	l	Benar	
	germany	0.000141	0.0388	0.9609	Win	d	-	
	italy	0.001752	0.1404	0.8578	Lose	d	-	
	france	0.05475	0.7719	0.1733	Lose	l	Benar	
	iceland	0.002122	0.1542	0.8436	Win	w	Benar	
semi_final	portugal	0.006398	0.6544	0.3391	Lose	l	Benar	2/2 (100%)
	wales	0.000012	0.2429	0.7570	Win	w	Benar	
	germany	0.000022	0.2764	0.7235	Win	w	Benar	
	france	0.000447	0.4671	0.5323	Lose	l	Benar	
final	portugal	0.302628	0.6605	0.0368	Lose	w	Salah	0/1 (0%)
	france	0.216428	0.6101	0.1733	Win	l	Salah	

Tabel 4  
Tabel hasil prediksi pada pada eksperimen 2.

Babak	Tim	Output dari Model			Prediksi	Hasil Nyata	Benar?	Akurasi
		Kelas 0	Kelas 1	Kelas 2				
round_of_16	switzerland	2.00E-09	0.0485	0.9514	Lose	d	-	5/6 (85%)
	poland	3.42E-13	0.0050	0.9949	Win	d	-	
	wales	3.60E-08	0.0955	0.9044	Win	w	Benar	

	north ireland	0.649837	0.3498	0.0002	Lose	1		
	croatia	2.00E-08	0.0799	0.9200	Win	w	Benar	
	portugal	1.71E-03	0.9643	0.0339	Lose	l		
	france	9.27E-04	0.1022	0.8968	Lose	l	Benar	
	republic of ireland	1.03E-08	0.0728	0.9271	Win	w		
	germany	2.95E-12	0.0091	0.9908	Lose	l	Benar	
	slovakia	0.004992	0.9848	0.0101	Win	w		
	hungary	2.11E-07	0.2437	0.7562	Win	w	Benar	
	belgium	3.66E-07	0.3308	0.6691	Lose	l		
	italy	5.72E-07	0.1952	0.8047	Lose	l	Benar	
	spain	5.76E-12	0.0110	0.9889	Win	w		
	england	0.000175	0.7320	0.2678	Lose	w	Salah	
	iceland	0.000113	0.8145	0.1853	Win	l		
	poland	1.26E-06	0.1906	0.8093	Win	d		
	portugal	0.000172	0.3647	0.6350	Lose	d	-	
	wales	1.52E-06	0.1931	0.8068	Win	w	Benar	
quarter_	belgium	1.98E-06	0.1968	0.8031	Lose	l		1/2
final	germany	2.47E-07	0.1692	0.1692	Win	d	-	(50%)
	italy	1.47E-06	0.1927	0.8072	Lose	d		
	france	5.59E-06	0.2118	0.7881	Win	l	Salah	
	iceland	0.000951	0.8042	0.1948	Lose	w		
	portugal	1.20E-08	0.3105	0.6894	Lose	l	Benar	
semi_fi	wales	1.51E-14	0.0208	0.9791	Win	w		2/2
nal	germany	3.50E-16	0.0096	0.9903	Win	w	Benar	(100%)
	france	4.22E-14	0.0284	0.9715	Lose	l		
final	portugal	3.55E-11	0.1864	0.8135	Lose	w	Salah	0/1 (0%)
	france	8.50E-18	0.0092	0.9907	Win	l		

Pertandingan pada sepak bola tidak bisa ditebak dengan mudah, contohnya pada penelitian ini, banyak tim yang bermain agresif, penguasaan lapangan yang banyak, akurasi mengoper yang tinggi, dan aspek lain yang meningkatkan kemungkinan menang. Tetapi, walaupun ada tim yang memiliki data statistik seperti itu, tim tersebut tercatat tetap kalah, memang model tidak salah jika menebak bahwa tim tersebut dapat menang dengan kemungkinan yang besar, tetapi pada nyatanya tim tersebut kalah. Hal itulah yang merupakan salah satu faktor mengapa akurasi pada hasil prediksi di penelitian ini masih terdapat kesalahan dalam menebak kemenangan tim pada pertandingan.

### PEMBAHASAN

Proses eksplorasi dan pengolahan data tidak menjadi masalah, namun perlu ditelusuri lebih lanjut mengenai penggunaan fungsi aktivasi, kelas membuat bobot acak, kelas mencari loss, dan kelas optimasi yang digunakan model. Penulis merasakan bahwa hasil prediksi yang dihasilkan menggunakan argumen dan model yang dibuat masih dapat ditingkatkan, mungkin dengan beberapa eksperimen menggunakan argumen yang berbeda dalam Tensorflow.Keras. Penelitian serupa seperti pada Huang & Chen (2011) menghasilkan akurasi yang tidak tinggi dan juga tidak rendah, hal ini membuat pertanyaan lebih lanjut mengenai model yang digunakan dalam memprediksi kemenangan dalam pertandingan sepak bola. Jaringan saraf tiruan yang diketahui sejauh ini terdapat Feed Forward Neural Network, Multilayer Perceptron, Convolutional Neural Network, Radial Basis Functional Neural Network, Recurrent Neural Network, LSTM – Long Short-Term Memory, Sequence to Sequence Models, Modular Neural Network, apakah dengan penggunaan neural network yang berbeda dapat menghasilkan akurasi yang signifikan? Penulis berharap untuk kedepannya peneliti dapat menggunakan jenis neural network lain untuk melakukan tujuan yang sama, dalam memprediksi kemenangan di olah raga sepak bola.

Pengumpulan data statistik merupakan hal tersulit dari penelitian ini, dengan banyaknya batasan akses terhadap setiap hasil pertandingan membuat penulis mengambil data secara manual dari pertandingan yang sudah berjalan dalam situs resmi UEFANYA sendiri. Perlu ditekankan bahwa dataset yang digunakan tidak terlalu banyak dan perlu ditambah lagi dengan sejarah pertandingan sepak bola negara atau tim tersebut. Namun, terdapat faktor-faktor lain yang dapat mempengaruhi suatu pertandingan berjalan, seperti stadion yang digunakan, mental pemain, komposisi tim, formasi, cuaca atau bahkan faktor pendukung (penonton). Faktor-faktor ini selain data statistik

pertandingan tersebut dapat dijadikan pertimbangan dalam melakukan prediksi kedepannya, penulis yakin bahwa hal-hal ini dapat berpengaruh besar terhadap model yang dibuat.

### KESIMPULAN

Penelitian ini berhasil dibangun sebuah model regresi MLP yang mampu memprediksi kemenangan pertandingan sepak bola dengan akurasi tinggi, berkisar antara 90% hingga 94%. Model ini menggunakan delapan kolom pada dataset sebagai prediktornya dan memiliki arsitektur MLP 8-7-3. Selain itu, telah berhasil dikembangkan perangkat lunak yang dapat menerima dataset statistik turnamen sepak bola, mengolahnya, dan menghasilkan hasil prediksi menggunakan model MLP 8-7-3 dengan delapan prediktor. Untuk penelitian selanjutnya, ada beberapa saran yang dapat diimplementasikan, seperti meningkatkan perangkat lunak agar dapat mengatasi dataset dari tipe turnamen seperti double round-robin, memperluas pemilihan fitur prediksi dengan memasukkan data statistik pemain dan kondisi lapangan permainan, serta memperbaiki perangkat lunak agar dapat memprediksi pertandingan tim yang berhadapan dengan tim selain lawan utamanya dalam suatu turnamen. Penambahan dataset dan penggunaan faktor-faktor eksternal lain dapat dipertimbangkan untuk dijadikan sebagai data training dalam model yang dibuat.

### REFERENSI

- McCulloch, W.S. and Pitts, W. (1990). A logical calculus of the ideas immanent in nervous activity. *Bulletin of Mathematical Biology*, 52, 99-115.
- Zayegh, A. and Bassam, N. A. (2018). Neural network principles and applications. Part of Asadpour, V. (ed.), *Digital Systems*. Intech Open, London, UK.
- Walvin, J. (2000) *The People's Game: The History of Football Revisited*, 2nd edition, Britannica, Chicago, Illinois.
- Shiffman, D. (2012) *The Nature of Code*, 1st Edition. Free Software Foundation, California, USA.
- Han, J., Kamber, M., and Pei, J. (2011) Advanced pattern mining. Part of Asadpour, V. (ed.), *Data Mining Concepts and Techniques*. Morgan Kaufmann Publishers, USA.
- Brownlee, J. (2018). *Better Deep Learning: Train Faster, Reduce Overfitting, and Make Better Predictions*. Machine Learning Mastery, USA.
- Memmert, D. and Raabe, D. (2018) *Data Analytics in Football: Positional Data Collection, Modeling and Analysis*, 1st edition. Taylor & Francis Group, London and New York, USA.
- Huang, K.-Y. and Chen, K.-J. (2011) Multilayer perceptron for prediction of 2006 world cup football game. *Advances in Artificial Neural Systems*, 2011
- Mirchandani, G. and Cao, W. (1989) On hidden nodes for neural nets. *IEEE Transactions on Circuits and Systems*, 36, 661-664.
- Anderson, G. (2021). Neural Networks: The basics of neural networks, and the math behind how they learn. <https://www.3blue1brown.com/topics/neural-networks>
- Das, S. (2021). Top 10 Most Popular Sports In The World October 2023. *Sports Monkie*. <https://sportsmonkie.com/most-popular-sports/>
- Anfilets, Sergei, Sergei Bezobrazov, Vladimir Golovko, Anatoliy Sachenko, Myroslav Komar, Raman Dolny, Valery Kasyanik, Pavlo Bykovyy, Egor Mikhno, and Oleksandr Osolinskyi. (2020). "Deep Multilayer Neural Network For Predicting The Winner Of Football Matches."
- Fialho, Gabriel, Aline Manhães, and João Paulo Teixeira. (2019). "Predicting Sports Results with Artificial Intelligence – A Proposal Framework for Soccer Games." *Procedia Computer Science* 164:131–36. doi: [10.1016/j.procs.2019.12.164](https://doi.org/10.1016/j.procs.2019.12.164).
- Hassan, Amr, Abdel-Rahman Akl, Ibrahim Hassan, and Caroline Sunderland. (2020). "Predicting Wins, Losses and Attributes' Sensitivities in the Soccer World Cup 2018 Using Neural Network Analysis." *Sensors* 20(11):3213. doi: [10.3390/s20113213](https://doi.org/10.3390/s20113213).
- Malamatinos, Marios-Christos, Eleni Vrochidou, and George A. Papakostas. (2022). "On Predicting Soccer Outcomes in the Greek League Using Machine Learning." *Computers* 11(9):133. doi: [10.3390/computers11090133](https://doi.org/10.3390/computers11090133).