Pemodelan Pemenang Penghargaan All-NBA Menggunakan *Machine Learning*

Dennis Jonathan[1]

[1]*STEM Universitas Prasetiya Mulya, Indonesia, 15339*
dennis.jonathan@student.pmsbe.ac.id

Abstrak. *Setiap tahunnya, liga bola basket di Amerika Serikat yang dikenal dengan NBA memberikan penghargaan All-NBA bagi pemain-pemain terbaik di musim tersebut. Setiap tim All-NBA terdiri atas 2 guards, 2 forwards, dan 1 centres dan NBA membentuk 3 tim All-NBA yang dipilih oleh media dan jurnalis. Oleh sebab itu, peneliti mencoba untuk membuktikan bahwa ada pola tertentu yang membuat pemain tertentu dipilih oleh jurnalis dan media sehingga pemilihan tidak dilakukan secara acak. Peneliti berhasil membuat model Neural Network yang dapat digunakan untuk memprediksi pemain-pemain tersebut menggunakan metrik Advanced Statistics yang tersedia dari Basketball-Reference. Model memiliki performa metrik F1 sebesar 82.5% dengan akurasi 98.8%. Uji statistik juga membuktikan bahwa pemilihan tersebut tidaklah acak. Peneliti juga mencoba untuk memprediksi pemain yang akan mendapatkan penghargaan untuk musim 2021-2022 dan model berhasil memprediksi 12 dari 15 pemain yang terpilih*

Keywords. *NBA dan All-NBA, Machine Learning, Neural Network, Prediksi*

1. Pendahuluan

1.1. Latar Belakang

Dengan adanya peningkatan teknik-teknik pengumpulan, pengolahan, dan penggunaan data di zaman sekarang, pandangan orang-orang terhadap bidang-bidang tertentu juga berubah. Salah satu bidang yang terdampak dari peningkatan data adalah dunia olahraga. Dunia olahraga yang dahulu hanya berupa manusia mendorong batas performa fisik dan mental untuk berkompetisi dengan satu sama lain, sekarang memiliki makna-makna baru di dunia olahraga. Dari segi positif, data mendorong manusia untuk melakukan perubahan dari cara kita mengeksekusikan sebuah rencana dalam suatu pertandingan olahraga. Misalkan, permainan pion yang agresif dengan gerakan h2-h4 di catur yang dibuat populer dari AlphaZero [1]. Selain itu, data juga membantu untuk melakukan simulasi strategi pitstop untuk menentukan kapan waktu optimal untuk mengganti ban atau menambah bahan bakar mobil dalam dunia balap [2]. Di dunia bola basket, terjadi peningkatan jumlah tembakan *three-point* dari tahun ke tahunnya yang disebabkan karena peningkatan penggunaan *data analytics* di bola basket [3].

Dari segi negatif, banyak orang berkata bahwa penggunaan data yang berlebihan dapat merusak esensi dan keindahan dari berkompetisi. Permainan yang sederhana dibuat menjadi terlalu kompleks dengan banyaknya data yang ada sekarang ini, baik ada nutrisi, data pergerakan pemain, dan masih banyak lagi. Chandrasekhar mengatakan bahwa bintang dari pertandingan olahraga fisik adalah atlet-atlet yang bercucur keringat bermain bukan komputer atau statistikawan yang menganalisa data tersebut [4].

Di sisi manapun anda berpihak, tentu kita tidak bisa menutupi fakta bahwa terjadi peningkatan penggunaan data di dunia olahraga sehingga peneliti merasa bahwa peningkatan pengumpulan, pengolahan, serta penggunaan data ini merupakan suatu sarana untuk memberikan arti dari keputusan-keputusan yang dibuat di dunia olahraga, salah satunya adalah di dunia bola basket. Sekarang, banyak sekali liga-liga bola basket yang berlangsung di mancanegara, salah satunya adalah topik dari penelitian ini. *National Basketball Association*, atau lebih sering disapa dengan

sebutan NBA, merupakan sebuah liga bola basket yang berasal dari Amerika Serikat. Liga tersebut seringkali disebut sebagai liga bola basket terbaik di dunia karena kualitas dari kurang lebih 450 atlet di 30 tim yang bermain disana [5].

Untuk bisa mengklasifikasikan pemain-pemain terbaik di liganya, setiap tahunnya NBA mengadakan sebuah penghargaan yang disebut sebagai All-NBA, dimana 15 pemain terbaik di NBA dipilih oleh media dan jurnalis melalui proses pemungutan suara menjadi tiga tim (All-NBA *first team*, *second team*, dan *third team*) [6]. Karena tim tersebut dipilih oleh media dan jurnalis , tentu ada faktor bias atau subjektivitas dari pilihan-pilihan tersebut [7]. Namun, seperti yang telah tertera di paragraf pertama, NBA tidaklah asing dengan pertumbuhan *data analytics*. Penggemar-penggemar dapat mencari data dari tim ataupun pemain favoritnya melalui berbagai situs di internet, layaknya situs resmi NBA, situs seperti FiveThirtyEight, dan juga Basketball Reference. Karena itu, penggemar-penggemar NBA dapat secara langsung bergabung dalam melihat siapa pemain-pemain yang layak untuk bisa mendapatkan penghargaan-penghargaan yang ada, seperti seleksi tim All-NBA, melalui performa pemain.

Dengan alasan tersebut, peneliti ingin mencoba untuk mencari pola-pola baik implisit maupun eksplisit yang ada dari data-data yang mudah diakses oleh penggemar bola basket yang dapat membuat seorang pemain layak menjadi bagian dari tim All-NBA di mata seorang pemilih (jurnalis dan media). Tujuan utama penelitian yang dilakukan peneliti adalah mencoba untuk memprediksi pemenang All-NBA menggunakan teknik-teknik *machine learning* dan mencari model yang memiliki performa yang baik untuk memprediksi pemenang-pemenang di periode kedepannya. Data yang akan digunakan berasal dari situs basketball-reference.com dan dibataskan untuk data tahun 1988 hingga tahun 2021 sebagai data untuk melatih dan mengembangkan model dan peneliti juga akan menggunakan data musim 2021 - 2022 sebagai data untuk menguji model terhadap data yang belum diketahui hasilnya.

1.2. Rumusan Masalah

Masalah yang diangkat dari penelitian ini adalah mencari pola-pola implisit maupun eksplisit yang dapat membuat seorang pemain layak menjadi bagian dari tim All-NBA di mata seorang pemilih dan memprediksi pemain-pemain yang terpilih menjadi tim All-NBA menggunakan teknik-teknik machine learning berdasarkan data-data historis yang dapat diakses dari basketball-reference.com.

Adapun pertanyaan yang akan dipilih dan akan dijawab melalui penelitian ini adalah sebagai berikut:

1. Bagaimana hubungan dari statistik yang dimiliki oleh seorang pemain dalam satu musim terhadap terpilihnya sebagai anggota dari tim All-NBA?
2. Bagaimana performa model *machine learning* yang dapat melakukan pemilihan anggota-anggota dari tim All-NBA dapat diprediksi? Model apakah yang sesuai untuk melakukannya?

1.3. Tujuan Penelitian

Tujuan dari penelitian yang dilakukan adalah sebagai berikut:

1. Mencari hubungan dari statistik pemain terhadap terpilihnya pemain tersebut sebagai anggota tim All-NBA
2. Mencari model yang sesuai untuk memilih pemain All-NBA dengan *machine learning*
3. Memprediksi pemain-pemain yang akan mendapatkan penghargaan All-NBA untuk musim-musim kedepannya

1.4. Manfaat Penelitian

Manfaat dari penelitian ini adalah sebagai berikut memberikan sebuah penilaian dalam bentuk probabilitas yang objektif dari kelayakan suatu pemain untuk mendapatkan nominasi All-NBA dengan menggunakan analisa dari machine learning. Penilaian ini tidak bergantung dengan subjektivitas ataupun agenda-agenda yang dimiliki oleh pemilih dan masih banyak lagi. Selain itu, penilaian objektif ini dapat dijadikan acuan performa dari suatu pemain dalam suatu musim relatif terhadap pemenang-pemenang penghargaan All-NBA sebelumnya dan dapat diaplikasikan ke dalam penyusunan strategi dan operasional dari suatu tim.

1.5. Batasan Masalah

Data yang akan digunakan untuk melatih dan membangun model-model machine learning dari penelitian ini dibatasi di periode tahun 1989 hingga 2021. Hal ini dilakukan karena pembagian tim All-NBA dalam bentuk tiga tim baru dilaksanakan pada tahun 1989. Untuk data yang digunakan untuk menguji adalah data musim terbaru yaitu musim 2022. Variabel-variabel yang digunakan juga dibatasi hanya untuk statistika regular season dan tidak memasukan statistika playoffs karena tim All-NBA disusun berdasarkan performa pemain selama regular season. Variabel yang akan diambil adalah variabel advanced statistics dari setiap pemain.

Model yang akan digunakan selama penelitian ini adalah *Random Forest*, Regresi Logistik, dan *Neural Network*. Untuk menguji performa dari masing-masing model, peneliti juga akan menggunakan metrik skor F1 sebagai metrik penguji utama, *recall*, *sensitivity*, serta akurasi sebagai komplemen dari metrik utama.

1.6. Sistematika Penulisan

Penulisan laporan penelitian ini terdiri dari lima bagian yang akan dibagi menjadi bab. Pembahasan dan sistematika penulisan dari masing-masing bab dan subbab yang akan dibahas adalah sebagai berikut:

BAB 1 PENDAHULUAN

Bab pertama pendahuluan akan menjelaskan mengenai latar belakang masalah, rumusan masalah, stujuan, manfaat, sistematika, dan juga batasan masalah dalam penulisan laporan penelitian ini.

BAB 2 TINJAUAN PUSTAKA

Bab kedua akan menjabarkan tentang teori dan konsep yang akan digunakan untuk melakukan penelitian ini. Disana juga ada beberapa penelitian-penelitian dengan masalah yang serupa yang telah dilakukan sebelumnya serta hipotesis dari penelitian ini.

BAB 3 METODOLOGI PENELITIAN

Bab ketiga akan memberikan penjelasan lebih dalam tentang metodologi yang akan digunakan dalam penelitian ini. Penjabaran yang dimaksud adalah penjabaran untuk jenis dari penelitian, subjek dari penelitian, teknik dari pengumpulan dan periode pengumpulan data, serta teknik analisis data yang akan dilakukan di penelitian.

BAB 4 HASIL DAN PEMBAHASAN

Bab keempat akan menguraikan hasil-hasil yang didapat dari analisis data sesuai dengan rumusan masalah yang sudah ditulis di Bab 1. Bagian ini juga akan menguraikan perbandingan dari model-model serta metrik pengujian yang sudah dipilih, serta gagasan-gagasan berdasarkan teori-teori yang ada.

BAB 5 PENUTUP

Bab kelima merupakan bagian akhir dari laporan penelitian ini. Bagian ini akan berisi kesimpulan serta saran dan rekomendasi yang dapat dilakukan oleh orang-orang yang melakukan penelitian dengan topik serupa kedepannya.

2. Tinjauan Pustaka

2.1. Landasan Teori

2.1.1. National Basketball Association

Menurut Britannica.com, *National Basketball Association* atau biasa juga disapa dengan NBA merupakan sebuah liga bola basket profesional yang berasal dari Amerika Serikat yang terbentuk di tahun 1949 [5]. Pemain-pemain basket terpopuler di dunia kemungkinan besar pernah bermain atau sedang bermain di NBA, layaknya Michael Jordan, LeBron James, Kobe Bryant, Stephen Curry, dan masih banyak lagi. Saat ini, ada 30 tim yang bermain di NBA dimana 29 tim tersebut berasal dari Amerika Serikat dan 1 tim berasal dari Kanada [5]. Selain itu, 30 tim tersebut juga dibagi menjadi dua divisi, yaitu divisi barat *(western conference*) dan divisi timur (*eastern conference*) dan subdivisi, yaitu *Southwest*, *Pacific*, dan *North West* untuk divisi barat, dan *Atlantic*, *Central*, dan *Southeast* untuk divisi timur. Setiap tahunnya, 8 tim dengan rekor terbaik dari masing-masing divisi akan memasuki ajang turnamen playoff untuk menentukan juara dari musim tersebut.

2.1.2. Penghargaan All-NBA

Setiap tahunnya, NBA akan memberikan penghargaan bagi pemain-pemain yang berperforma tinggi di musim tersebut, salah satunya adalah penghargaan All-NBA. Tim All-NBA adalah sebuah penghargaan yang diberikan kepada pemain-pemain terbaik sesuai dengan posisi-posisi yang ia mainkan [6]. Sebuah tim All-NBA akan terdiri atas 2 *guards*, 2 *forwards* dan 1 *centre*. Untuk pemain-pemain yang memainkan beberapa posisi, ia akan dipilih sesuai dengan posisi yang sering ia mainkan . Sejak musim 1988-1989, jurnalis akan memilih 3 tim (All-NBA *first team*, *second team*, dan *third team*) sehingga akan ada 15 pemain yang terpilih ke tim-tim tersebut. Pemain-pemain tersebut dipilih oleh media dan jurnalis dengan sistem pemungutan suara untuk masing-masing tim (jurnalis akan memilih 5 pemain untuk All-NBA *first team* dan seterusnya). Mekanisme pemungutan suara adalah sebagai berikut:

- 5 poin untuk pemain yang mendapatkan suara untuk *first team*
- 3 poin untuk pemain yang mendapatkan suara untuk *second team*
- 1 poin untuk pemain yang mendapatkan suara untuk third team

Pemain-pemain akan diurutkan per posisi berdasarkan jumlah suara mereka, sehingga untuk ketiga tim tersebut, akan ada 6 *guards*, 6 *forwards*, dan 3 *centres* [7].

2.1.3. *Machine Learning*

Machine learning merupakan suatu hal yang sebenarnya tidak asing untuk ditemukan di keseharian manusia, misalkan di aplikasi-aplikasi telepon genggam, mobil *self-driving*, hingga rekomendasi-rekomendasi yang dapat ditemukan di situs *e-commerce*. IBM mendefinisikan *machine learning* sebagai sebuah bagian dari *artificial intelligence* dan juga *computer science* yang berfokus untuk menggunakan data dan algoritma-algoritma untuk mengemulasi cara seorang manusia belajar dan perlahan-lahan meningkatkan tingkat akurasinya [8]. Brown mendefinisikannya sebagai proses yang dimulai dari data dan dari sana, model komputer akan melatih dirinya dan mencari pola-pola yang ada di data tersebut [9]. Dasar dari algoritma-algoritma yang digunakan dalam proses *machine learning* adalah persamaan-persamaan matematika seperti layaknya Regresi Linear dan Regresi Logistik.

Salah satu pengembangan dari *machine learning* adalah teknik-teknik yang disebut sebagai *deep learning*. Andrew Ng mengatakan bahwa perbedaan utama dari teknik *machine learning* konvensional seperti regresi dan *Support Vector Machine* (SVM) dan teknik *deep learning* adalah kemampuan pembelajaran yang bisa dilakukan dengan peningkatan data [10]. Seiring berjalannya waktu, jumlah data yang dapat diakses juga meningkat pesat. Apabila menggunakan algoritma-algoritma konvensional, tingkat pembelajaran akan melandai dengan peningkatan jumlah data, sehingga performa model tidak akan meningkat secara signifikan. Penurunan tingkat pembelajaran algoritma *deep learning* akan menurun di jumlah data yang sangat besar dan model akan memiliki performa yang tinggi lebih dengan jumlah data yang relatif sama dengan algoritma konvensional.

2.2. Model Penelitian dan Metode Analisis

2.2.1. Regresi Logistik

Subasi mendefinisikan Regresi Logistik sebagai sebuah metode yang mudah dan sederhana untuk melakukan klasifikasi biner yang sering digunakan di beberapa industri [11]. Hasil dari Regresi Logistik tersebut adalah sebuah probabilitas yang berada di antara nilai $[0, 1]$. Hosmer, et.al mendefinisikan formula dari Regresi Logistik sebagai [12]:

$$\hat{y} = P(Y = 1|\mathbf{x}) = \frac{e^{g(\mathbf{x})}}{1+e^{g(\mathbf{x})}} \quad \textbf{(2.1)}$$

$$g(\mathbf{x}) = \beta_0 + \beta_1 x_1 + ... + \beta_n x_n \quad \textbf{(2.2)}$$

dimana:

- $\hat{y}$ adalah nilai prediksi
- x_n adalah nilai input x ke n
- β_n adalah nilai dari koefisien dari x_n

Nilai dari koefisien-koefisien dapat dicari menggunakan meminimalisir fungsi *cost* yang diadaptasikan dari fungsi *log-likelihood* dengan persamaan [13]:

$$J(\beta_0, \beta_1, ..., \beta_n) = -\frac{1}{m} \sum_{i=1}^{m} y_i \, ln\,(\hat{y_i}) + (1 - y_i) \, ln\,(1 - \hat{y_i}) \quad \textbf{(2.3)}$$

dimana:

- m adalah jumlah data
- y_i dan $\hat{y_i}$ adalah nilai asli dan prediksi untuk data i

Menurut Stoltzfus ada beberapa asumsi yang harus dipenuhi oleh regresi logistik yaitu *error* yang independen, linearitas dari fungsi logit untuk variabel kontinu, tidak ada multikolinearitas, dan tidak ada *outlier* [14].

2.2.2. *Random Forest*

Decision Tree adalah metode klasifikasi dengan struktur hirarki dimana interpretasi sederhananya adalah algoritma memecah data-data secara rekursif dengan aturan-aturan tertentu sehingga memudahkan klasifikasi dari input yang diberikan [15]. Grömping mendefinisikan sebuah model *Random Forest* merupakan kumpulan dari beberapa *Decision Tree* [16].

Grömping juga menambahkan bahwa salah satu algoritma yang sering digunakan di baik untuk *Random Forest* maupun *Decision Tree* adalah algoritma CART (*Classification and Regression Tree*) yang dibuat oleh Breiman di tahun 1984. CART memilih pemisahan untuk setiap titik sehingga pengurangan *impurity* dilakukan secara maksimum, dimana *impurity* adalah selisih kuadrat dari titik-titik tengah. Secara singkat, CART bekerja dengan cara membangun pohon yang besar dan perlahan-lahan memangkas ranting-ranting menggunakan *cross-validation* (validasi dengan data lain) [17]. Hasil akhir dari CART adalah fitur dan batasan yang menghasilkan pembelajaran terbesar (*information gain*) di titik tersebut.

Salah satu konsep yang perlu diperhatikan dalam *Random Forest* adalah entropi. Entropi didefinisikan sebagai besaran *impurity* dalam suatu data [18]. Entropi dapat dihitung dengan cara:

$$E = -\sum_{i=1}^{N} p_i \, log_2 \, p_i \quad \textbf{(2.4)}$$

dimana:

- N adalah jumlah kelas atau sampel yang ada
- p_i adalah probabilitas mengambil sebuah sampel i di titik tersebut.

Dengan entropi, *information gain* (besaran entropi yang hilang karena terpisahnya cabang) juga dapat diukur dengan persamaan:

$$\text{Gain} = E_{\text{Parent}} - E_{\text{Children}} \quad \textbf{(2.5)}$$

dimana:

- E_{Parent} adalah titik induk di *Decision Tree*
- E_{Children} adalah titik anak atau cabang di D*ecision Tree*

2.2.3. *Neural Network*

Mueller mendefinisikan *Neural Network* adalah sebuah algoritma yang bisa bekerja dengan data yang kompleks karena model tersebut bisa mengolah beberapa input dan melalui proses dari beberapa lapisan bisa menghasilkan suatu hasil [19]. Dalam satu lapisan, akan ada unit-unit kecil yang disebut neuron. Untuk setiap lapisan ada sebuah proses yang dinamakan *forward propagation*.

Proses yang dilakukan dalam *forward propagation* adalah sebagai berikut:

$$Z^{[l]} = W^{[l]} A^{[l-1]} + b^{[l]} \quad \textbf{(2.6)}$$

$$A^{[l]} = g(Z^{[l]}) \quad \textbf{(2.7)}$$

dimana:

- $W^{[l]}$ adalah beban di lapisan l yang di transpose
- $b^{[l]}$ adalah konstanta di lapisan l
- $A^{[l]}$ adalah nilai dari fungsi aktivasi di lapisan tersebut
- $g(Z^{[l]})$ adalah fungsi aktivasi yang digunakan di lapisan tersebut

Perlu dicatat bahwa notasi di atas berlaku untuk seluruh data yang digunakan untuk melatih model. X atau input dari model juga dapat ditulis $A^{[0]}$. Untuk mempercepat pelatihan, mengurangi resiko dari *covariate shift*, dan mempermudah untuk melatih algoritma dengan banyak lapisan, penggunaan *batch normalization* juga dapat dilakukan. Persamaan dari *batch normalization* adalah sebagai berikut [20]:

$$z^{(i)}_{\text{norm}} = \frac{z^{(i)} - \mu}{\sqrt{\sigma^2 + \epsilon}} \quad \textbf{(2.8)}$$

$$\tilde{z}^{(i)} = \gamma z^{(i)}_{\text{norm}} + \beta \quad \textbf{(2.9)}$$

dimana:

- $z^{(i)}_{\text{norm}}$ adalah nilai neuron i pre-aktivasi yang sudah di normalisasi$z^{(i)}_{\text{norm}}$ adalah nilai neuron i pre-aktivasi yang sudah di normalisasi
- $z^{(i)}$ adalah nilai neuron i pre-aktivasi$z^{(i)}$ adalah nilai neuron i pre-aktivasi
- μ adalah rata-rata nilai neuron i pre-aktivasiμ adalah rata-rata nilai neuron i pre-aktivasi
- σ^2 adalah variansi nilai neuron i pre-aktivasiσ^2 adalah variansi nilai neuron i pre-aktivasi
- ϵ adalah angka yang sangat kecil agar pembagi di pecahan tidak 0ϵadalah angka yang sangat kecil agar pembagi di pecahan tidak 0
- $\tilde{z}^{(i)}$ adalah nilai *batch normalization* untuk neuron i
- β *dan* γ *adalah parameter persamaan batch normalization*

Ada beberapa fungsi aktivasi yang digunakan yaitu fungsi Sigmoid dan fungsi ReLU (*Rectified Linear Unit*). Persamaan dari kedua fungsi tersebut adalah sebagai berikut:

$$\text{Sigmoid}(z) = \frac{1}{1+e^{-z}} \tag{2.10}$$

$$\text{ReLU}(z) = max(0, z) \tag{2.11}$$

dimana:

- z adalah input dari fungsi, baik ReLU ataupun Sigmoid

Untuk melakukan pembaruan dari variabel $W^{[l]}$ dan $b^{[l]}$, diperlukan sebuah fungsi *cost*, dan untuk klasifikasi biner, fungsi *cost* yang digunakan sama dengan Persamaan 2.3 dan perlu dicari turunan dari kedua variabel tersebut terhadap fungsi *cost* .

Algoritma optimasi yang akan digunakan adalah algoritma Adam. Tilawah mengatakan bahwa algoritma Adam merupakan campuran dari algoritma optimasi RMSprop dan juga *Momentum Gradient Descent* [21].

Persamaan dari algoritma Adam adalah [22]:

$$W^{[l]} = W^{[l]} - \alpha \frac{V^{[l]}_{dW\,\text{Adjusted}}}{\sqrt{S^{[l]}_{dW\,\text{Adjusted}}} + \epsilon} \tag{2.12}$$

$$b^{[l]} = b^{[l]} - \alpha \frac{V^{[l]}_{db\,\text{Adjusted}}}{\sqrt{S^{[l]}_{db\,\text{Adjusted}}} + \epsilon} \tag{2.13}$$

dimana:

- α adalah *learning rate*
- ϵ adalah angka yang sangat kecil agar hasil pembagi tidak akan nol.

$$V^{[l]}_{dW\,\text{Adjusted}} = \frac{\beta_1 V^{[l]}_{dW} + (1-\beta_1) dW}{1-\beta_1^t} \tag{2.14}$$

$$V^{[l]}_{db\,\text{Adjusted}} = \frac{\beta_1 V^{[l]}_{db} + (1-\beta_1) db}{1-\beta_1^t} \tag{2.15}$$

$$S^{[l]}_{dW\,\text{Adjusted}} = \frac{\beta_2 S^{[l]}_{dW} + (1-\beta_2) dW^2}{1-\beta_2^t} \tag{2.16}$$

$$S^{[l]}_{db\,\text{Adjusted}} = \frac{\beta_2 S^{[l]}_{db} + (1-\beta_2) db^2}{1-\beta_2^t} \tag{2.17}$$

$S^{[l]}_{dW\,\text{Adjusted}}$ dan $S^{[l]}_{db\,\text{Adjusted}}$ merupakan komponen yang berasal dari RMSprop dengan parameter β_2 sebagai *exponentially weighted average* dan t adalah iterasi sekarang untuk RMSprop. dW dan db adalah turunan dari fungsi *cost* yang digunakan terhadap W dan b. Proses pembaruan dari kedua variabel tersebut disebut juga sebagai *backward propagation.*

Apabila terjadi *overfitting*, suatu solusi yang dapat digunakan *regularization*. *Regularization* bekerja dengan cara memberikan penalti yang lebih besar ke beban dari komponen linear di setiap neuron dalam *Neural Network*, sehingga saat melakukan optimasi dari *cost* function, beban yang akan didapatkan adalah beban-beban yang relatif kecil.

Dalam penelitian ini, jenis *regularization* yang akan digunakan adalah L2 *regularization* menggunakan *Frobenius Norm* dengan persamaan [23]:

$$||A||_F^2 = \sum_{i=1}^{m} \sum_{j=1}^{n} |a_{i,j}|^2 \quad \textbf{(2.18)}$$

dimana:

- $a_{i,j}$ adalah nilai dari matriks di posisi i dan j

Selain menggunakan L2 *regularization*, cara lain untuk mengurangi *overfitting* untuk model dengan banyak lapisan adalah dengan menggunakan suatu metode *Dropout*. Wan, et.al. menggambarkan proses yang dinamakan *Dropout* atau *DropConnect* sebagai sebuah algoritma *regularization* dimana beberapa aktivasi neuron dalam suatu lapisan dimatikan atau dijadikan menjadi 0 secara acak [24]. Efek dari metode ini adalah berkurangnya ketergantungan neuron-neuron lain dalam lapisan selanjutnya terhadap nilai dari neuron yang dimatikan, sehingga terjadi efek *regularization* dan memungkinkan untuk melatih sebuah model dengan lapisan yang lebih banyak lagi.

Mueller juga menggambarkan model *Neural Network* layaknya air mengalir di sungai, dimana data input mengalir di *network* dari satu lapisan ke lapisan lainnya [19]. Proses untuk mencari empiris yang memerlukan implementasi solusi-solusi berbeda dan melakukan *testing* terhadap target yang kita inginkan.

2.2.4. *Synthetic Minority Oversampling Technique*

Brownlee menyebutkan bahwa *Synthetic Minority Oversampling Technique* atau disingkat SMOTE merupakan sebuah teknik untuk menangani masalah klasifikasi *imbalanced* yang disebabkan karena kurangnya jumlah contoh dari kelas minoritas [25]. Algoritma SMOTE pertama kali dicetuskan oleh Nitesh Chawla dan rekan-rekannya di tahun 2002 [26]. Cara algoritma SMOTE melakukan *oversampling* dengan menggunakan contoh-contoh acak dari kelompok minoritas yang dekat dengan *K-Nearest Neighbor* dan membentuk data sintetik melalui tetangga-tetangga tersebut dengan acak juga. Brownlee menambahkan bahwa proses pembentukan sintetik baru yang memungkinkan dengan karakteristik yang mirip dengan kelas minoritas.

2.3. Metrik Pengujian

Untuk menguji performa dari model yang telah dibuat, ada beberapa metrik untuk model *classifier* yang bisa digunakan. Metrik-metrik tersebut adalah sebagai berikut:

2.3.1. Akurasi

Lee mendefinisikan metrik akurasi sebagai jumlah dari seluruh prediksi yang benar dibagi dengan total dari jumlah prediksi [27]. Secara matematis, persamaan tersebut dapat ditulis sebagai

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}} \quad \textbf{(2.19)}$$

dimana:

- TP menandakan *true positive*
- FP menandakan *false positive*
- TN menandakan *true negative*
- FN menandakan *false negative*

2.3.2. *Sensitivity*

Lee mendefinisikan *sensitivity* atau *precision* sebagai metrik yang menghitung proporsi dari prediksi positif yang terhitung sebenarnya merupakan nilai positif [27]. Secara matematis, *sensitivity* dapat dituliskan sebagai:

$$\text{Sensitivity} = \frac{\text{TP}}{\text{TP}+\text{FP}} \tag{2.20}$$

dimana:

- TP menandakan *true positive*
- FP menandakan *false positive*

2.3.3. *Recall*

Recall atau juga sering disapa dengan *true positive rate* atau TPR merupakan metrik yang mendeskripsikan proporsi dari nilai yang sebenarnya positif yang terprediksi positif [27].

Definisi matematis dari *recall* adalah sebagai berikut:

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}} \tag{2.21}$$

dimana:

- TP menandakan *true positive*
- FN menandakan *false negative*

2.3.4. F1 *Score*

F1 *Score* merupakan rata-rata harmonik dari metrik *sensitivity* dan *recall* [27]. F1 *Score* adalah metrik yang cocok untuk digunakan untuk model dengan kategori yang tidak seimbang. Definisi matematis dari metrik ini adalah sebagai berikut

$$\text{F1 Score} = 2 \cdot \frac{\text{S}\cdot\text{R}}{\text{S}+\text{R}} \tag{2.22}$$

dimana:

- S menandakan nilai dari metrik *sensitivity*
- R menandakan nilai dari metrik *recall*

2.4. Penelitian Serupa

Sebagai referensi dari penelitian ini, peneliti juga telah meninjau penelitian-penelitian serupa yang telah dilakukan oleh peneliti-peneliti sebelumnya. Ringkasan dari penelitian serupa sebelumnya dapat dibaca di Lampiran 1.

Uudmae mencoba menggunakan beberapa Teknik *machine learning* untuk memprediksi hasil dari pertandingan-pertandingan NBA periode tahun 2013-2016 [28]. Format data yang beliau gunakan hanyalah data tim yang bermain saat ini dalam satu pertandingan (tim rumah dan pendatang), jumlah pertandingan yang sudah dimenangkan oleh kedua belah pihak, dan pemenang dari pertandingan tersebut. Teknik *machine learning* yang digunakan oleh Uudmae adalah *Support Vector Machine* (SVM), Regresi Linear, dan juga *Neural Network regression* (NNR). Hasil yang beliau capai menggunakan SVM adalah akurasi 62,07%, 63,75% untuk Regresi Linear, dan 64,95% untuk NNR. Beliau juga hanya menggunakan data dari tahun 2013 hingga 2016, sehingga jumlah data yang digunakan tidak terlalu besar yaitu 4.500 data.

Penelitian yang dilakukan Albert, Lopez, Allbright, dan Blas menggunakan kombinasi dari *machine learning* untuk memprediksi pemain yang akan masuk ke dalam tim NBA *All-Star* [29]. Untuk melakukan pengujian model, mereka menggunakan metrik "*sensitivity*" dengan pengujian *K-Fold-Cross-Validation*. penelitian ini juga mengambil data dari situs basketball-reference.com dengan periode 1980 hingga 2021. Untuk memangkas pemain *outlier*, mereka menggunakan syarat bahwa seorang pemain haruslah bermain setidaknya 50 pertandingan dan menjadi *starter* di 25 pertandingan Model yang digunakan adalah *Random Forest classifier* (pengembangan dari model *Decision Tree*), *Adaboost classifier*, dan *Multi-Layer Perceptrons classifier*. Hasil *sensitivity* yang mereka dapatkan adalah 0,524, 0,619, dan 0,619 untuk ketiga model tersebut. Mereka juga

membuat sebuah kombinasi dari ketiga model menggunakan *Artificial Neural Network* (ANN) dan algoritma *gradient descent* untuk mencari beban dari nilai ketiga model tersebut. Alhasil ANN yang dibuat adalah peningkatan *sensitivity* dari model menjadi 0,81. Tim penelitian tersebut menyarankan penggunaan kombinasi dari ketiga model awal dan menggabungkannya menjadi sebuah model ANN. Penggunaan ANN dari multimodel ini membuat kompleksitas dari model menjadi tinggi namun terbayar dengan hasil *sensitivity* yang lebih rendah. Menggunakan *gradient descent* untuk mencari beban dari masing-masing variabel juga merupakan sebuah ide yang menarik.

Wang dan Fan menggunakan data yang tersedia di situs basketball-reference.com untuk mencari tiga hal, yaitu memprediksi pemain All-Star, hasil dari *playoff*, membuktikan kebenaran atas *hot streak*, dan terakhir melihat tren di NBA [30]. Model-model yang digunakan oleh mereka berupa *Random Forest Classifier, Decision Tree*, *K-Nearest Neighbor classifier, Gradient Boosting*, Regresi Linear, dan Regresi Logistik. Mereka juga melakukan *Principal Component Analysis* (PCA), namun hasil yang ada lebih terlihat sebagai fase analisis awal data (*exploratory data analysis*). Untuk permasalahan *classifier*, metrik yang digunakan sebagai acuan adalah akurasi. Model terbaik untuk memprediksi All-Star adalah Regresi Logistik dengan akurasi 97,9%. Mereka juga menemukan bahwa *hotstreak* itu bukanlah hal yang nyata. Selain itu, data *advanced statistics* cenderung menghasilkan model yang lebih baik.

Dos Santos, Wang, Carlsson, dan Lambrix melakukan penelitian mengenai hasil dari musim di NBA dengan data tahun 2008-2009 hingga 2017-2018 [31]. Hasil yang mereka cari lebih tepatnya adalah bagaimana performa tim di akhir tahun menggunakan data individu, boxscore, dan masih banyak lagi. Data yang mereka gunakan juga berasal dari basketball-reference.com. Sama seperti penelitian sebelumnya, metrik pengukuran model yang digunakan adalah akurasi dengan model seperti Regresi Logistik, *linear Support Vector Machine*, *Random Forest* dan *Multi-Layer Perceptron*. Model yang terbaik yaitu *Random Forest* memiliki akurasi 69,88%. Model tersebut mereka gunakan untuk memprediksi hasil dari setiap pertandingan dalam satu musim, dan untuk menambahkan elemen *randomness,* mereka menggunakan sebuah sistem *coin toss*. Mereka menyimulasikan kalender tahun 2017-2018 sebanyak 10.000 kali untuk mendapatkan frekuensi dari masing-masing tim. Mereka juga menggabungkan data dari fivethirtyeight.com untuk mencari nilai ELO dari setiap tim dan membandingkan hasil yang mereka dapatkan dibandingkan dengan nilai ELO (apakah model yang mereka miliki akan *outperform* kinerja model ELO dari situs tersebut atau tidak).

Cheng, Zhang, Kyebambe, dan Kimbugwe mencoba untuk memprediksi hasil dari NBA *playoffs* menggunakan model yang dibuat dari dasar menggunakan prinsip *maximum entropy* [32]. Mereka menggunakan 15 fitur untuk masing-masing tim dan diambil dari situs stats-nba.com dengan data tahun 2008 - 2018.. Hasil dari model mereka adalah sebuah probabilitas yang pada akhirnya akan diuji menggunakan metrik akurasi. Mereka juga melakukan pengetesan *threshold* probabilitas yang terbaik untuk pembulatan dari probabilitas tersebut. Selain akurasi, mereka juga menggunakan *Receiver Operating Characteristics* (ROC) dan *Area Under Curve* (AUC) untuk mengevaluasi model yang mereka buat. Model yang dibuat juga dibandingkan dengan model-model *classifier* lainnya seperti *Naïve Bayes*, Regresi Logistik*, Neural Network*, dan *Random Forest* dan model yang mereka secara umum lebih unggul dibandingkan model-model *machine learning* lainnya. Hasil dari model tersebut adalah akurasi 74,4%. Tujuan utama dari penelitian yang dilakukan oleh tim ini adalah pembuatan model sendiri dan hasilnya pun terlihat saat akurasi dari model tersebut lebih tinggi dari hasil algoritma *machine learning*. Namun seperti yang dinyatakan oleh tim, memprediksi NBA *playoffs* sangatlah sulit karena banyak faktor yang tidak bisa ditangkap oleh model tersebut.

Penelitian yang dilakukan oleh Jones bertujuan untuk memprediksi statistik yang muncul di sebuah pertandingan NBA yaitu *points spread differential* (PSD) atau selisih dari poin antara kedua tim. [33] Data yang beliau ambil adalah 144 data hasil dari *stratified random sampling* dari pertandingan antara tahun 2008 dan 2011 dari website USAtoday.com. Beliau menggunakan tingkat signifikansi untuk menentukan variabel yang signifikan terhadap model dengan . Untuk menentukan PSD, Regresi Linear digunakan oleh beliau dengan mengabaikan parameter (*intercept*). Untuk menentukan pemenang dari pertandingan, beliau menggunakan Regresi logistik. Setelah melakukan kedua regresi, beliau melakukan validasi dengan 50 pertandingan yang tidak

digunakan untuk melatih model. Hasil akhir dari penelitian adalah sebuah model PSD dengan akurasi 94% dan model kemenangan dengan akurasi 88%.

Wilkens juga melakukan pemodelan untuk cabang olahraga tenis dengan tujuan untuk memprediksi hasil dari pertandingan tenis dan mencoba untuk menggambarkan sedikit aplikasi dari model-model tersebut terhadap dunia perjudian [34]. Data yang digunakan merupakan menggunakan data ATP, WTA, ITF, dan *grand slams* (*Australian Open*, *Roland Garros*, *Wimbledon*, dan *US Open*). Wilkens melakukan penelitian ini dengan beberapa model yaitu Regresi Logistik, *Neural Network* (NN), *Random Forest, Gradient Boosting Machine* (GBM) dan juga *Support Vector Machine* (SVM). Dari penelitian yang ia lakukan, akurasi tertinggi yang didapatkan adalah 69,1% saat melakukan prediksi menggunakan GBM. Beliau juga menambahkan bahwa tidak ada model yang mampu menembus akurasi lebih dari 70% saat melakukan prediksi hasil pertandingan tersebut.

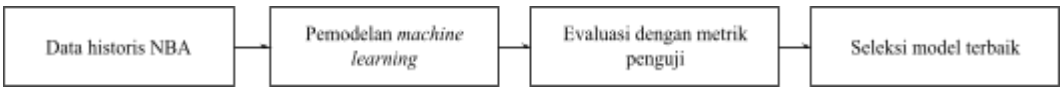
Migliorati menemukan bahwa untuk memprediksi hasil dari pertandingan basket di NBA, penggunaan fitur dengan sistem skor ELO menghasilkan model yang lebih baik dari pada model dengan fitur *box-score* (dalam penelitian ini, model fitur yang digunakan untuk *box-score* adalah fitur yang dinamakan Oliver's *four factors* yaitu *effective field goal%*, *turnover ratio*, *offensive rebound%*, dan *free throw rate*) [35]. penelitian yang dilakukan oleh beliau menggunakan model *Neural Network* dengan bantuan *Keras*. Beliau juga menggunakan algoritma optimasi Adam, fungsi loss *binary cross entropy* dan metrik akurasi untuk melakukan "*sanity check"*. Hasil dari penelitian beliau adalah model ELO secara konsisten memiliki akurasi yang lebih tinggi selama periode yang beliau observasi (2005-2020) dengan akurasi musiman terbaik 70,53%.

2.5. Kerangka Pikir Teoritis

Dengan bertambahnya jumlah data yang dapat diakses oleh publik, penggunaan algoritma-algoritma *machine learning* semakin meningkat. Baik dari teori, definisi, dan riset-riset serupa, terlihat jelas bahwa ada cara-cara untuk melakukan prediksi terutama berupa klasifikasi dari dunia olahraga terutama di objek dari penelitian ini yaitu di NBA. Metode-metode yang dilakukan menggunakan proses *machine learning* cukup menunjukan bahwa ada pola-pola tersurat dan juga tersirat dari data-data yang ada di dalam data-data untuk NBA sehingga bisa membuat sebuah prediksi yang relatif akurat. Meskipun kebanyakan dari penelitian yang serupa memprediksi hasil-hasil dari pertandingan, beberapa teknik dan metode dari penelitian sebelumnya dapat dijadikan contoh untuk diimplementasikan di penelitian ini.

Dalam penelitian ini, peneliti ingin menggali lebih dalam mengenai cara-cara untuk bisa melakukan prediksi dari target yaitu pemenang penghargaan All-NBA setiap tahunnya dengan menggunakan beberapa teknik *machine learning*. Seperti yang sudah dibahas sebelumnya, pemilihan pemenang penghargaan All-NBA dapat dinilai subjektif karena dipilih berdasarkan media dan jurnalis, oleh sebab itu, ada ruang untuk mencari pola-pola yang ada dalam kasus ini serta memberikan evaluasi yang dapat dinilai lebih objektif berdasarkan pemenang-pemenang penghargaan tersebut sebelumnya.

Pemaparan dari teori-teori beberapa model merupakan gambaran singkat mengenai cara dan alat-alat analisis yang akan digunakan selama jalan penelitian ini. Ketiga model yang sudah tertera di Bab 2.2 akan digunakan untuk melakukan pemodelan tersebut. Performa dari ketiga model akan dibandingkan dengan metrik pembanding utama akurasi, namun juga akan ditambahkan metrik-metrik lain untuk melihat performa tersebut. Adapun diagram sederhana dari kerangka pikir teoritis adalah sebagai berikut:



Gambar 2.1. Skema Kerangka Pikir Teoritis

2.6. Hipotesis penelitian

Dugaan awal atau hipotesis dari sebuah penelitian adalah sebuah tebakan dari hasil penelitian yang akan dilakukan. Objektif dari penelitian ini adalah menemukan hubungan antara statistika yang diperoleh oleh seorang pemain basket di liga NBA dalam suatu musim terhadap probabilitas pemain tersebut memperoleh penghargaan All-NBA. Apabila hubungan tersebut tidaklah acak, maka model jelas ada suatu hubungan antara variabel independen terhadap dependen. Adapun perumusan hipotesis dari penelitian ini adalah sebagai berikut:

$$H_0 : \pi = 0,5$$

$$H_1 : \pi \neq 0,5$$

Pengujian hipotesis akan dilakukan menggunakan uji binomial. Hasil dari probabilitas tersebut merupakan *p-value* sehingga dapat dibandingkan dengan tingkat signifikansi yang digunakan di penelitian ini yaitu . Jika *p-value* kurang dari $\alpha = 0,05$, maka H_0 akan ditolak sehingga dapat dilihat bahwa pemilihan pemain All-NBA tidak dilakukan secara acak, dan sebaliknya juga berlaku.

3. Metodologi Penelitian

3.1. Metode Penelitian

Berdasarkan masalah yang diangkat untuk penelitian ini, jenis dari penelitian yang akan dilakukan adalah penelitian kuantitatif dengan teknik statistika untuk menganalisa data yang telah diperoleh. Arikunto mendefinisikan pendekatan kuantitatif sebagai pendekatan penelitian yang berfokus ke penggunaan angka-angka, dimulai dari pengumpulan, penafsiran, dan pemaparan dari hasil penelitian [36].

Secara spesifik, jenis penelitian kuantitatif yang dilakukan adalah penelitian korelasional dimana peneliti mencari hubungan antara variabel-variabel independen terhadap variabel dependen. Bhandari mengatakan bahwa pendekatan penelitian korelasional merupakan penelitian yang mencari hubungan-hubungan variabel tanpa mengontrol atau memanipulasi nilai dari variabel tersebut [37].

3.2. Objek, Tempat dan Waktu Penelitian

3.2.1. Objek Penelitian

Hal yang menjadi prioritas dari sebuah penelitian adalah apa yang akan diteliti atau juga disebut objek penelitian. Hal tersebut disebabkan karena masalah dari penelitian terkandung dalam objek yang diteliti serta solusi-solusi pemecahannya. Objek dari penelitian yang dilakukan peneliti adalah NBA dan pemain-pemain yang bermain di liga tersebut. Alasan pemilihan objek penelitian adalah objek tersebut akan dimodelkan menggunakan *machine learning* untuk melihat apakah ada cara untuk memprediksi pemenang All-NBA dengan statistika yang ada.

3.2.2. Tempat dan Waktu Penelitian

Penelitian ini dilakukan secara daring atau *remote* karena data yang digunakan oleh peneliti merupakan data resmi sekunder yang dapat diperoleh melalui situs basketball-reference.com [38]. Waktu penelitian yang dilakukan dimulai di bulan April 2022 hingga bulan Juni 2022.

3.3. Data Penelitian

3.3.1. Populasi

Menurut Glen, populasi adalah keseluruhan atau seluruh anggota dari sebuah segmentasi atau grup [39]. Populasi yang diangkat dalam penelitian ini adalah statistika dari pemain-pemain NBA. Hingga saat ini, sudah ada 75 musim yang dimainkan di NBA. Peneliti juga akan menggabungkan pemenang-pemenang dari penghargaan All-NBA untuk setiap musimnya.

3.3.2 Sampel

Britanica.com menggambarkan sampel sebagai proses atau metode untuk menarik grup yang dapat merepresentasikan sebuah populasi [40]. Teknik yang digunakan peneliti untuk mengambil sampel yang digunakan selama penelitian ini adalah *purposive sampling*. Salmaa menyebutkan bahwa *purposive sampling* adalah sebuah metode *sampling* yang *non-random* dimana peneliti memilih sampel-sampel yang dapat menggambarkan atau representatif populasi yang diangkat di penelitian yang dilakukan [41].

Sampel yang akan diambil oleh peneliti adalah statistika *advanced* dari pemain-pemain musim 1988-1989 hingga 2021-2022. peneliti menggunakan metode penarikan sampel tersebut karena musim 1988-1989 adalah musim pertama dimana jurnalis dan media memilih tiga tim All-NBA. Peneliti juga tidak akan menggunakan statistika dari pemain yang kurang dari 10 pertandingan di setiap musimnya. Pemotongan tersebut dilakukan karena kontrak terpendek yang dapat diberikan kepada seorang pemain adalah kontrak 10 hari (dengan asumsi pemain akan bertanding setiap harinya) [42] dan pemain-pemain dengan kontrak tersebut sangat tidak memungkinkan untuk mendapatkan penghargaan All-NBA. Pemotongan dengan aturan ini juga akan menghilangkan pemain-pemain yang dapat disebut *outlier*.

3.3.3. Sumber Data dan Metode Pengumpulan Data

Data yang digunakan oleh peneliti merupakan data sekunder berupa data yang publik yang dapat diakses melalui situs basketball-reference.com. Data tersebut ditarik dari situs basketball-reference.com di tanggal 11 April 2022. Seperti yang ditulis sebelumnya, data yang diambil adalah data *advanced* dari setiap pemain dengan aturan yang sudah disebut di Bab 3.2.2. Penarikan data tersebut dilakukan menggunakan algoritma bahasa pemrograman Python yang terinspirasi dari algoritma yang dibuat oleh Dietzel di tahun 2020 [43].

Data tersebut akan dibagikan menjadi 3 kelompok, yaitu data untuk melatih model atau *training set*, data untuk melakukan validasi atau *development set*, dan data untuk melakukan test atau *test set*. Data untuk *training set* dan *development set* akan berasal dari data *advanced statistics* untuk pemain-pemain musim 1988-1989 hingga 2020-2021 dan akan dipisahkan secara acak menggunakan rasio 75% *training set* dan 25% *development set*. Data yang digunakan untuk *test set* merupakan data dari musim 2021-2022.

3.4. Variabel Penelitian

3.4.1. Variabel Dependen

Variabel dependen adalah variabel yang menjadi tujuan, diukur, dan dilakukan pengujian dari sebuah eksperimen [44]. Dalam penelitian ini, variabel dependen dari penelitian ini adalah apakah pemain tersebut memenangkan penghargaan All-NBA di musim tersebut. Tipe dari variabel dependen yang dicari merupakan sebuah data *boolean* atau biner sehingga untuk setiap pemain, hanya ada dua kemungkinan nilai variabel dependen, yaitu 0 atau 1.

3.4.2. Variabel Independen

Menurut Helmenstine, variabel independen adalah variabel yang dapat diubah atau dikontrol dalam sebuah eksperimen yang dapat secara langsung menyebabkan perubahan pada variabel dependen [45]. Dalam riset ini, akan ada 27 variabel independen yang digunakan untuk mencari variabel dependen yang disebutkan di bagian sebelumnya. Variabel-variabel yang digunakan sebagai variabel independen adalah Player, Pos, Age, Tm, G, TMP, PER, TS%, 3PAR, FTR, ORB%, DRB%, TRB%, AST%, STL%, BLK%, TOV%, USG%, OWS, DWS, WS, WS/48, OBPM, DBPM, BPM, VORP, dan Year. Variabel-variabel ini merupakan statistika yang dapat digolongkan sebagai *advanced statistics* yang menggambarkan performa pemain selama satu musim. Variabel-variabel tersebut juga tertera melalui data yang sudah diambil dari situs basketball-reference.com.

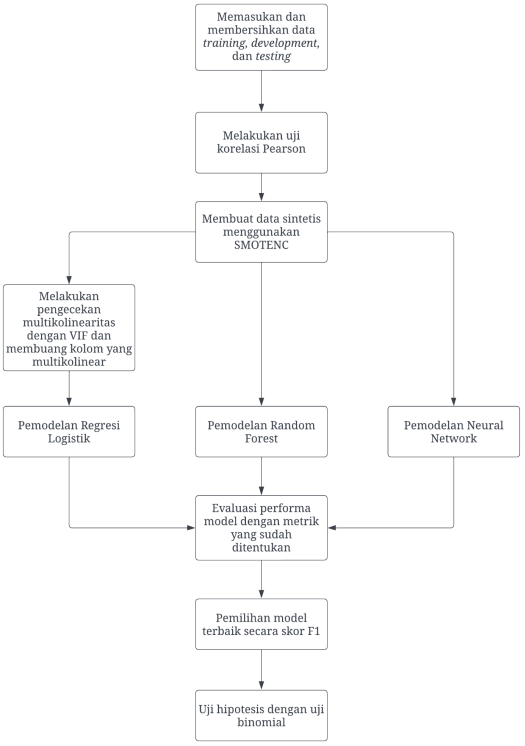
Penggunaan dari variabel independen tidak bersifat permanen, artinya untuk mendapatkan model yang terbaik, akan ada beberapa variabel independen yang tidak signifikan ataupun tidak bisa digunakan (layaknya nama dan tim pemain) terhadap model yang dicari. Oleh sebab itu, variabel-variabel tersebut tidak akan dimasukkan ke dalam model yang digunakan. Semua variabel selain Player, Pos, dan Tm merupakan data numerik. Data non-numerik digunakan hanya untuk melakukan *sanity check*. Data untuk Pos akan dikonversi menjadi variabel

dummy. Variabel Year tidak akan digunakan dalam analisis, namun hanya digunakan saat proses pembersihan data. Penjelasan lengkap dari variabel independen yang digunakan dapat dilihat di Lampiran 1 [46 - 48].

3.5. Teknik dan Alur Analisis Data

Analisa dari data yang sudah diperoleh akan digunakan menggunakan bahasa pemrograman Python 3.9 menggunakan Jupyter Notebook. Analisa data juga akan dibantu menggunakan beberapa *libraries* Python yang akan dijelaskan secara spesifik di sub bab berikutnya, namun secara umum ada 4 *library* yang akan digunakan yaitu NumPy dan Pandas untuk memproses data serta Matplotlib dan Seaborn untuk melakukan pembentukan grafik.

Berikut adalah skema dari teknik dan alur analisis data:



Gambar 3.1. Teknik dan Alur Analisis Data Penelitian

3.5.1. Uji Korelasi Pearson

Korelasi Pearson atau sering ditulis sebagai variabel r adalah sebuah metrik yang menggambarkan sebuah variabel dengan variabel lainnya, baik variabel dependen maupun independen. Korelasi Pearson memiliki rentang nilai $[-1, 1]$ dimana korelasi linear negatif menandakan hubungan berbanding balik dan korelasi linear positif menandakan hubungan berbanding lurus. Nilai $r = 0$ menandakan ketiadaan hubungan. Rumsey mendefinisikan korelasi Pearson dengan persamaan [49]:

$$r = \frac{1}{n-1} \sum \frac{(x-\bar{x})(y-\bar{y})}{S_x \, S_y} \tag{3.1}$$

dimana:

- $\bar{x}$ dan $\bar{y}$ adalah mean dari sampel
- S_x dan S_y adalah standar deviasi dari sampel.

Untuk mencari korelasi Pearson dari variabel yang dimiliki, *library* Pandas memiliki metode ".corr()".

3.5.2. Pemodelan Regresi Logistik

Pemodelan Regresi Logistik akan dilakukan menggunakan *library* Statsmodels dengan fungsi "statsmodels.logit()". Seperti yang tertera di Bab 2.1.2, salah satu syarat dari Regresi Logistik adalah tidak adanya multikolinearitas antar variabel independen. Untuk bisa menguji multikolinearitas, penggunaan uji *Variance Inflation Factor* atau VIF dapat digunakan.

Faraway mendefinisikan formula dari VIF dengan persamaan [50]:

$$\text{VIF} = \frac{1}{1-R_j^2} \tag{3.2}$$

dimana:

- R_j^2 adalah koefisien determinasi untuk variabel j

Pengecekan VIF di Python dapat dilakukan menggunakan *library* Statsmodels menggunakan fungsi yang dinamakan "variance_inflation_factor".

Untuk menggunakan VIF, kita bisa menggunakan skala yang dijelaskan oleh Glen yaitu [51]:

Tabel 3.1. Penjelasan hasil VIF

No.	Interval VIF	Deskripsi
1	$\text{VIF} \leq 1$	Tidak ada multikolinearitas
2	$1 < \text{VIF} < 5$	Multikolinearitas sedang
3	$\text{VIF} \geq 5$	Multikolinearitas tinggi

Selain mengeliminasi multikolinearitas, pengecekan linearitas di fungsi logit untuk variabel kontinu juga perlu diperhatikan. Artinya, fitur atau variabel independen yang tidak signifikan terhadap model akan dihilangkan dari model yang dibuat.

Untuk melakukan itu, kita bisa menggunakan "model.summary()" dari Statsmodels untuk mengecek *p-value* atau nilai probabilitas dari variabel independen relatif terhadap distribusi t tersebut dengan hipotesis:

$$H_0 : \beta_i = 0$$

$$H_1 : \beta_i \neq 0$$

dimana:

- β_i adalah parameter untuk variabel independen i

Apabila *p-value* lebih kecil atau sama dengan tingkat signifikan (α), H_0 akan ditolak sehingga parameter untuk variabel tersebut tidaklah nol atau artinya signifikan. Penelitian ini akan menggunakan $\alpha = 0,05$.

3.5.4. Pemodelan Random Forest

Berbeda dengan Regresi Logistik, algoritma *Random Forest* tidak memiliki prasyarat spesifik untuk bisa digunakan. Penjelasan dari *Random Forest* dapat dilihat di Bab 2.1.3. Pemodelan *Random Forest* akan dilakukan menggunakan *library* Scikitlearn dengan fungsi "RandomForestClassifier()".

Dalam penelitian ini, fokus dari pemodelan *Random Forest* adalah melakukan pengaturan parameter dari fungsi tersebut. Parameter yang akan diatur adalah jumlah *estimator* dan kriteria dari pemisahan cabang. Ada dua kemungkinan yaitu entropi dapat dilihat di Persamaan 2.4 atau penggunaan kriteria impuritas Gini.

Impuritas Gini dapat dihitung dengan persamaan sebagai berikut [52]:

$$Gini(D) = 1 - \sum_{i=1}^{k} p_i^2 \quad \textbf{(3.3)}$$

dimana:

- p_i adalah probabilitas untuk mendapatkan sampel kelas i di suatu titik tertentu

3.5.5. Pemodelan Neural Network

Sama layaknya dengan *Random Forest*, tidak ada prasyarat untuk bisa mengaplikasikan model *Neural Network*. Untuk mempermudah pembentukan *Neural Network*, *library* Tensorflow akan digunakan selama penelitian ini. Dalam pembentukan model *Neural Network*, pengaturan yang dilakukan untuk membuat model yang lebih baik adalah jumlah dan jenis lapisan, jumlah neuron dalam satu lapisan, durasi pelatihan, fungsi *cost* dan algoritma optimasi.

Untuk penelitian ini, fungsi *cost* yang akan digunakan untuk mengoptimasi model dinamakan "binary_cross_entropy" atau fungsi *cost* yang sama seperti Regresi Logistik yang dapat dilihat di Persamaan 2.3. Algoritma untuk melakukan optimasi fungsi *cost* tersebut adalah algoritma Adam.

3.5.6. Perbandingan Model

Data akan dibagi menjadi dua kelompok yaitu data musim 1988-1989 hingga 2020-2021 yang akan disebut data *traindev* dan data musim 2021-2022 yang disebut data *test*. Ketiga model yang dibuat di penelitian ini akan dilatih di *training set* dengan proporsi 75% dari jumlah data *traindev* yang diambil secara acak.

Untuk menguji performa selama pengaturan model, data yang akan digunakan adalah data *development set* dengan proporsi 25% dari data *traindev* yang juga dipilih secara acak. Karena kurangnya jumlah pemain All-NBA relatif dengan pemain yang tidak memenangkan penghargaan tersebut, algoritma SMOTE akan digunakan untuk melakukan *oversampling* dari kelas minoritas hanya untuk data *traindev*.

Metrik yang digunakan untuk melakukan pengujian dari model dapat dilihat di Bab 2.1.6. Metrik-metrik tersebut akan digunakan untuk menguji performa model terhadap *training set* dan juga *dev set*. Model yang memiliki performa skor F1 terbaik akan digunakan untuk memprediksi data musim 2021-2022 dan probabilitas tertinggi dari 6 *guards*, 6 *forwards*, dan 3 *centre*s akan ditampilkan. Perlu diketahui bahwa pemenang dari penghargaan All-NBA untuk musim 2021-2022 belum diumumkan saat penulisan dan penarikan data yang digunakan untuk penelitian.

3.5.7. Uji Hipotesis

Hipotesis yang dinyatakan di Bab 2.4 akan diuji menggunakan uji binomial. Uji binomial merupakan uji statistika yang dibuat berdasarkan distribusi diskrit binomial. Akurasi dari model yang terbaik akan dikalikan dengan jumlah *dev set* dan dibulatkan ke bawah sehingga mendapatkan jumlah data yang benar diprediksi oleh model. Setelah itu, perhitungan probabilitas dapat dilakukan dengan persamaan berikut:

$$\Pr(X = x) = \binom{n}{x} p^x (1-p)^{n-x} \quad \textbf{(3.4)}$$

Untuk uji binomial dalam penelitian, variabel-variabel tersebut didefinisikan sebagai:

- n adalah jumlah data
- x adalah hasil kali dari akurasi dan jumlah data.
- p adalah probabilitas untuk mendapatkan jawaban benar, atau sama dengan π.

Melakukan uji binomial di Python dapat dilakukan dengan menggunakan bantuan *library* SciPy dengan fungsi "binom_test()".

4. Hasil dan Pembahasan

4.1. Pembersihan Data

Setelah data diperoleh dan dilakukan pengecekan dimensi, dapat terlihat jelas bahwa dimensi data mentah adalah 14.838 baris dengan 27 kolom. Peneliti melakukan pengecekan data yang kosong dan menghitung persentase dari data yang kosong relatif terhadap jumlah data. Hasil dari perhitungan tersebut dapat dilihat di Tabel 4.1 adalah hasil yang diperoleh. Terdapat 12 kolom yang memiliki data yang kosong, namun karena data yang kosong kurang dari satu persen jumlah data total serta imputasi akan bisa mengubah isi dari baris tersebut, peneliti memilih untuk membuang data-data tersebut.

Tabel 4.1. Persentase Kolom yang Memiliki Data Kosong

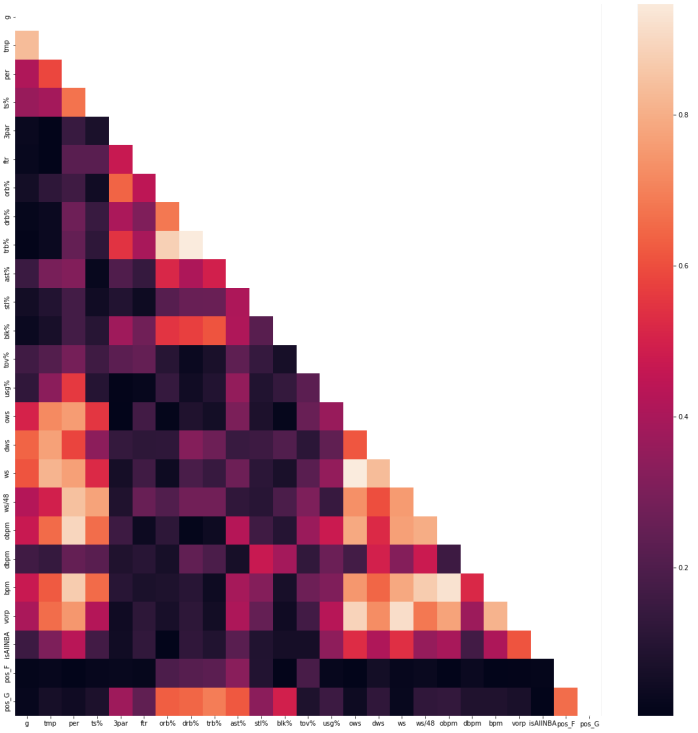
No.	Nama Fitur	Data Kosong (%)
1	3PAR	0,35
2	FTR	0,35
3	TS%	0,32
4	TOV%	0,26
5	ORB%	0,02
6	DRB%	0,02
7	TRB%	0,02
8	AST%	0,02
9	STL%	0,02
10	BLK%	0,02
11	USG%	0,02
12	WS/48	0,02

Seperti yang telah tertera di Bab 3.3, peneliti juga melakukan pemotongan data berdasarkan jumlah permainan yang sudah dimainkan pemain dalam satu musim dengan minimal permainan 10 pertandingan. Data-data yang tidak memenuhi syarat akan dibuang dari data yang digunakan untuk pemodelan. Analisis dari data yang dimiliki menemukan 1.245 pemain yang tidak memenuhi syarat pemotongan tersebut.

Peneliti juga membuang kolom ‘Age’ dan ‘Tm’ karena kedua kolom tersebut tidak akan digunakan untuk pemodelan, namun nama dari pemain akan disimpan untuk melakukan *sanity check*. Bab 3.4 juga menjelaskan konversi kolom ‘Pos’ menjadi *dummy variable* yang juga dilakukan peneliti. Sebelum dikonversi menjadi *dummy variable*, kolom ‘Pos’ juga dikonversi nilainya untuk melambangkan tiga posisi dalam tim All-NBA yaitu *guard* (G), *forward* (F), dan *centre* (C). Hasil dari semua operasi yang tertera di atas adalah data yang memiliki dimensi 13.593 baris dan 26 kolom atau pengurangan sebesar 8,4 persen baris dari data.

4.2. Eksplorasi Data

4.2.1. Korelasi Pearson



Gambar 4.1. *Heatmap* dari Korelasi Pearson antar Kolom

Korelasi Pearson bisa digambarkan secara intuitif melalui *heatmap*. Dapat dilihat dari Gambar 4.1, semakin terang warna dari *heatmap*, semakin tinggi korelasi Pearson mutlak yang dimiliki antara dua kolom. Untuk kolom target yang dicari, yaitu 'isAllNBA', ada beberapa kolom yang tampaknya memiliki korelasi yang lebih besar dari 35 persen, yaitu 'PER', 'OWS', 'DWS', 'VORP', 'BPM', 'OBPM', 'WS' dan 'WS/48'. Nilai dari korelasi Pearson mutlak kolom 'isAllNBA' dengan kolom-kolom tersebut adalah sebagai berikut:

Tabel 4.2. Korelasi Pearson Mutlak

No.	Nama Fitur	Korelasi Mutlak (%)
1	VORP	61
2	WS	53,7
3	OWS	53
4	PER	43,4
5	DWS	41,1
6	BPM	40,8
7	OBPM	39,6
8	WS/48	35,5

Tabel 4.2 juga menggambarkan seberapa pentingnya metrik-metrik komposit terhadap pemenang penghargaan tersebut. Dapat terlihat juga bahwa WS atau *Win Share* dan variasinya memiliki peranan yang penting saat menentukan pemain yang layak untuk mendapatkan penghargaan All-NBA. Meskipun korelasi-korelasi tersebut tidak bisa dinilai sangat tinggi terhadap variabel dependen, kombinasi dari kolom tersebut dengan kolom-kolom yang lain tentu dapat menciptakan model yang cukup baik untuk memprediksi variabel dependen yang ingin dimodelkan.

4.2.2. Statistika Deskriptif

Dari tahun 1988-1989 hingga 2020-2021, ada 495 pemain yang mendapatkan penghargaan tersebut dan ada 13.098 pemain yang tidak mendapatkan penghargaan All-NBA. Untuk bisa menggambarkan pemain yang mendapatkan penghargaan All-NBA dan tidak mendapatkan penghargaan tersebut, melakukan analisis statistika deskriptif seperti rata-rata, simpangan baku, dan masih banyak lagi dapat dijadikan acuan. Peneliti mengagregasikan data berdasarkan pemain yang memenangkan All-NBA dan tidak memenangkan All-NBA dan melihat delapan kolom yang memiliki korelasi tertinggi terhadap variabel dependen tersebut.

Value Over Replacement Player (VORP) menggambarkan seberapa penting seorang pemain kepada timnya dengan cara membandingkan seberapa mudah pemain ini untuk digantikan kontribusinya dengan pemain lain. Pemain yang memiliki VORP yang tinggi sangat sulit untuk digantikan di timnya dan sebaliknya. Pemain yang mendapatkan penghargaan All-NBA rata-rata VORP sebesar 5,159, nilai yang hampir 10 kali lebih tinggi dibandingkan pemain yang tidak mendapatkan penghargaan tersebut (0,523).

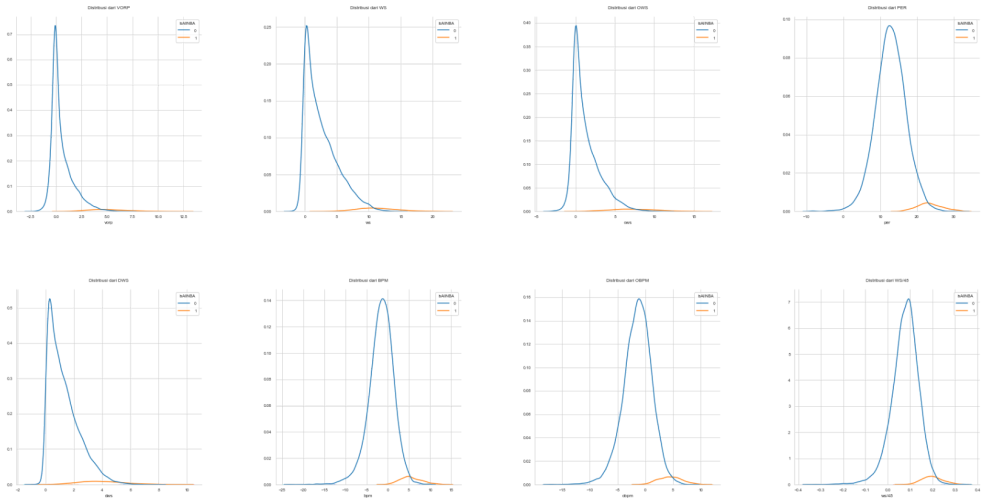
Player Efficiency Rating (PER) merupakan statistika yang melihat seberapa efisien seorang pemain saat bertanding. Rata-rata PER di NBA setiap tahunnya adalah 15, sehingga pemain yang berada di atas 15 merupakan pemain yang efisien, dan pemain yang berada dibawah 15 merupakan pemain yang kurang efisien. Rata-rata PER pemain All-NBA juga lebih tinggi dibanding pemain yang tidak mendapatkannya dengan nilai 23,562 untuk pemain All-NBA dan 12,887 untuk pemain yang tidak mendapatkannya.

Box Plus Minus (BPM) dan *Offensive Box Plus Minus* (OBPM) menggambarkan kontribusi pemain di sebuah pertandingan basket. OBPM lebih berfokus kepada aspek menyerang sementara BPM lebih bersifat umum. Semakin tinggi nilai tersebut, semakin baik kontribusi pemain. Nilai ini juga bisa berupa nilai negatif yang berarti pemain bukannya membantu timnya, tetapi membantu tim musuh. Rata-rata BPM dan OBPM juga lebih tinggi untuk pemain All-NBA (5,497 dan 4,519) dibandingkan pemain yang tidak mendapatkannya (-1,524 dan -1,377).

Terakhir adalah *Win Share* atau WS dan variasinya. WS menggambarkan seberapa banyak kemenangan yang secara teori dapat diperoleh jika pemain ini bermain di sebuah tim dalam suatu musim. Pemain All-NBA memiliki rata-rata WS sebesar 11,286 dan pemain non-All-NBA memiliki rata-rata sebesar 2,565. Jika WS diskalakan agar pemain misalnya bermain satu pertandingan NBA penuh dari awal hingga akhir, metrik yang didapatkan adalah *Win Share per 48 Minutes* (WS/48). WS/48 pemain All-NBA (0,199) juga lebih tinggi dibandingkan pemain yang tidak mendapatkannya (0,077). WS juga bisa dibagi menjadi dua kategori, yaitu *Offensive Win Share* (OWS) dan *Defensive Win Share* (DWS) yang menggambarkan WS untuk aspek menyerang dan bertahan. Untuk OWS, pemain All-NBA memiliki rata-rata 7,327, nilai yang jauh lebih tinggi dibandingkan pemain non-All-NBA (1,276). Hal tersebut juga sama untuk DWS dimana rata-rata pemain All-NBA berada di nilai 3,961 dibandingkan 1,288 untuk pemain yang bukan All-NBA.

Dapat disimpulkan melalui statistika deskriptif bahwa pemain All-NBA memiliki nilai yang lebih tinggi untuk statistika *advanced* yang memiliki korelasi tinggi terhadap variabel dependen. Metrik-metrik tersebut menandakan besarnya kontribusi pemain terhadap performa timnya baik dalam sebuah pertandingan ataupun dalam suatu musim. Dapat terlihat dari OWS dan OBPM, pemain-pemain All-NBA memiliki performa yang sangat baik saat menyerang dibandingkan dengan rata-rata pemain yang bukan All-NBA. Namun tentu observasi ini merupakan generalisasi dari kolom-kolom yang memiliki korelasi terbesar terhadap variabel dependen. Tentu akan ada metrik-metrik lain yang bisa memberikan gambaran yang lebih lengkap lagi terhadap peluang pemain mendapatkan penghargaan All-NBA.

4.2.3. Distribusi Kolom dengan Korelasi Tertinggi



Gambar 4.2. Distribusi Fitur Delapan Korelasi Pearson Tertinggi

Gambar 4.2 memberikan sedikit persepsi mengenai seberapa langkanya performa pemain yang mendapatkan penghargaan All-NBA. Garis biru menggambarkan pemain yang tidak mendapatkan penghargaan tersebut dan garis oranye menggambarkan pemain All-NBA. Dapat dilihat bahwa sesuai dengan Bab 4.2.2, pemain All-NBA merupakan pemain yang secara umum memiliki nilai statistika *advanced* yang lebih tinggi dari pemain-pemain lainnya. Jumlah pemain tersebut juga sangat langka dibandingkan dengan mayoritas pemain di NBA.

4.3. Pemodelan

4.3.1. Preparasi Data untuk Dimodelkan

Setelah data yang digunakan dibersihkan dan diseleksi di Bab 4.1, peneliti perlu menyiapkan data yang akan dimodelkan. Pertama adalah pembagian data untuk *training* dan *development* menggunakan data musim 1989-1990 hingga 2020-2021 sesuai dengan yang tertera di Bab 3.3.3. Pembagian dilakukan menggunakan *library* Scikit-Learn. Besar dari data *development* adalah seperempat dari jumlah data total, sehingga data *training* berjumlah 10.194 baris dan data *development* 3,399 baris.

Setelah melakukan pembagian data, peneliti melakukan penskalaan dari fitur-fitur (kecuali *dummy variable*) menggunakan metode *standard scaling* menggunakan mean dan simpangan baku dari data *training* untuk kedua data (*training* dan *development*). Penskalaan merupakan suatu hal yang sering dilakukan dalam *machine learning* agar model tidak sensitif terhadap perubahan *input* yang drastis dan perbedaan skala *input*. Penskalaan dilakukan menggunakan *library* Scikit-Learn.

Karena dataset yang dimiliki memiliki masalah *imbalance class* (di data keseluruhan, jumlah kelas positif yaitu pemain All-NBA berjumlah 495 sementara pemain yang bukan All-NBA berjumlah 13.098), penulis menggunakan algoritma SMOTE untuk menciptakan data sintetis dari kelas minoritas hanya untuk data *training*. Implementasi SMOTE yang digunakan adalah algoritma *Synthetic Minority Over-sampling Technique for Nominal and Continuous* (SMOTENC) dari *library* Imbalanced-Learn. Penggunaan SMOTENC dilakukan karena variabel independen memiliki data dengan tipe kategorik dan numerik. Setelah menciptakan data sintetis menggunakan SMOTENC, dimensi data *training* meningkat jadi 19,646.

4.3.2. Regresi Logistik

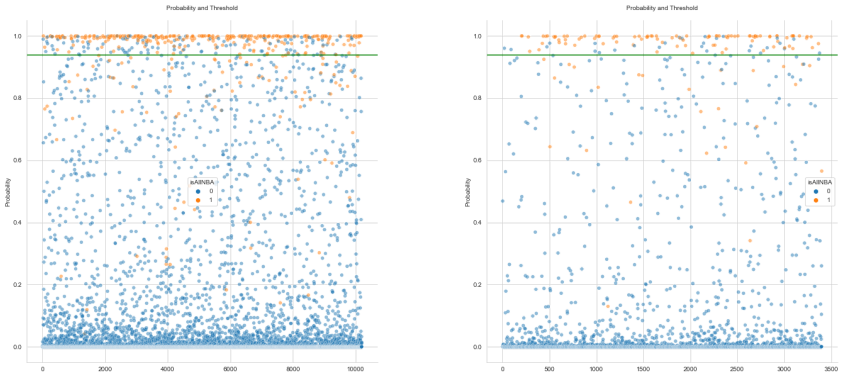
Karena Regresi Logistik memiliki prasyarat bahwa variabel independen tidak boleh memiliki multikolinearitas, peneliti menggunakan *Variance Inflation Factor* (VIF) untuk mencari nilai multikolinearitas setiap variabel independen dan membuang kolom-kolom yang memiliki VIF lebih besar dari lima. Untuk mencari VIF, peneliti menggunakan *library* Statsmodels. Variabel-variabel independen yang tidak terbuang adalah '3PAR', 'FTR', 'AST%', 'BLK%', 'OWS', 'DWS', ' OBPM', 'DBPM', dan 'Pos_F'. 'Pos_F' merupakan *dummy variable* untuk pemain yang memiliki posisi *forward*.

Peneliti juga membuang variabel independen yang tidak memiliki pengaruh signifikan secara statistik terhadap model Regresi Logistik. Variabel independen yang akan disimpan merupakan kolom yang memiliki *p-value* dari nilai z untuk setiap fitur yang berada dibawah tingkat signifikansi (α) dibawah 0,05 sehingga menghasilkan model dengan fitur '3PAR', 'FTR', 'AST%', 'BLK%', 'OWS', 'DWS', ' OBPM' dan 'DBPM'.

$$\hat{y} = P(Y = 1|\mathbf{x}) = \frac{e^{g(\mathbf{x})}}{1+e^{g(\mathbf{x})}} \tag{4.1}$$

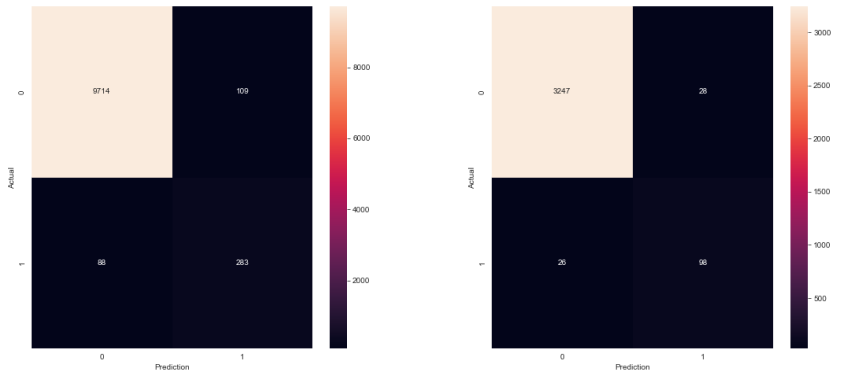
$$\begin{aligned} g(\mathbf{x}) = &-7.6295 - 0.9289\ 3\text{PAR} + 0.3525\ \text{FTR} \\ &+0.1648\ \text{AST\%} + 0.2121\ \text{BLK\%} + 0.1899\ \text{OWS} \\ &+2.014\ \text{DWS} + 4.1413\ \text{OBPM} - 0.3089\ \text{DBPM} \end{aligned} \tag{4.2}$$

Persamaan 4.2 adalah persamaan yang didapatkan setelah melakukan pemodelan Regresi Logistik menggunakan *library* Statsmodels. Semua fitur yang digunakan di dalam model sudah tidak memiliki masalah multikolinearitas dan juga sudah signifikan secara statistik. Setelah mencari *threshold* probabilitas yang memaksimalkan nilai F1, *threshold* yang sesuai untuk model tersebut adalah 0,94. Berikut adalah hasil prediksi dari data *training* dan *development* menggunakan *threshold* yang sudah ditentukan:



Gambar 4.3. Prediksi Regresi Logistik Data *Training* (kiri) dan Data *Development* (kanan)

Berdasarkan Gambar 4.3, algoritma Regresi Logistik terlihat bisa memprediksi baik data *training* dan data *development*. *Threshold* sebesar 0,94 dapat memisahkan prediksi kelas positif (titik berwarna oranye) dan kelas negatif (titik berwarna biru) yang dihasilkan model dengan baik, terlihat di garis berwarna hijau pada Gambar 4.3. Terlihat bahwa sedikit titik yang salah diprediksi relatif terhadap *threshold* yang ditentukan. Evaluasi lebih lanjut untuk model *classifier* bisa dilakukan menggunakan *confusion matrix*. Berikut adalah *confusion matrix* untuk data *training* dan *development*.



Gambar 4.4. *Confusion Matrix* Regresi Logistik Data *Training* (kiri) dan Data *Development* (kanan)

Hasil dari *confusion matrix* di Gambar 4.4 dapat digunakan untuk mengevaluasi model menggunakan metrik-metrik performa model. Seperti yang sudah ditentukan, metrik yang ditentukan sebagai prioritas untuk pembanding adalah nilai F1 dan metrik lain, seperti akurasi, *recall*, dan *precision*, digunakan untuk memberikan gambaran tambahan tentang performa model. Berikut adalah metrik yang diperoleh untuk model Regresi Logistik:

Tabel 4.3. Performa Regresi Logistik

Metrik	*Training*	*Development*
Akurasi	0,981	0,984
Recall	0,763	0,79
Precision	0,722	0,778
F1	0,742	0,784

Data *training* memiliki performa yang lebih buruk dari data *development*. Hal tersebut dapat terjadi apabila data *development* lebih mudah untuk digeneralisasikan dibandingkan dengan data *training*. Dapat dilihat bahwa model ini memiliki akurasi yang sangat tinggi (98,1% untuk data *training* dan 98,4% untuk data *development*), namun akurasi tersebut memberikan sebuah konklusi yang menyesatkan karena masalah *imbalanced data* yang telah disebutkan sebelumnya. Model Regresi Logistik dapat memprediksi 76,3% pemain yang sesungguhnya merupakan All-NBA di data *training* dan 79% di data *development*. Dari seluruh prediksi positif yang diperoleh model, 72,2% prediksi di data *training* dan 77,8% di data *development* merupakan pemain yang sebenarnya mendapatkan All-NBA. Karena *recall* dan *precision* dari data *development* lebih tinggi dibandingkan data *training*, nilai dari skor F1 untuk data *development* lebih tinggi dari data *training*. Namun secara umum, model berperforma cukup baik menurut nilai F1, dan memiliki performa pada metrik untuk data *development* yang mendekati data *training* sehingga generalisasi model baik.

4.3.3. *Random Forest*

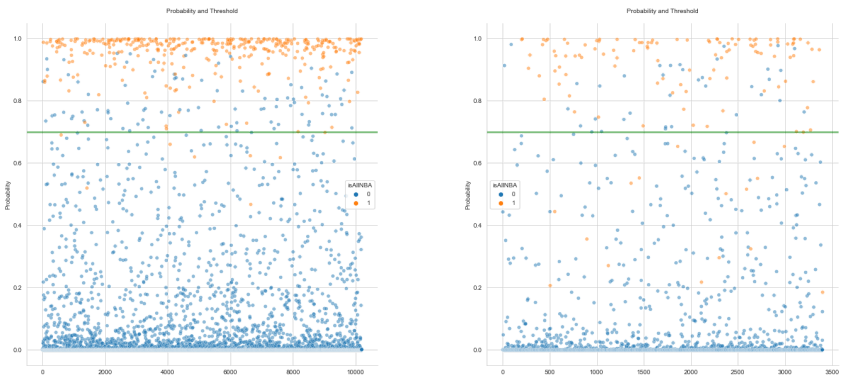
Seperti yang tertera di Bab 3.5.4, tidak ada prasyarat khusus yang perlu dipenuhi dalam melakukan pemodelan menggunakan algoritma *Random Forest*. Seluruh pemodelan dilakukan menggunakan Scikit-Learn. Setelah melakukan *random searching* terhadap 100 calon model dengan *cross-validation* tiga kali untuk mencari empat *hyperparameter*, yaitu jumlah estimator, kriteria pemisahan, jumlah sampel minimum untuk pemisahan, dan jumlah maksimum daun di titik, model terbaik didapatkan dengan *hyperparameter* jumlah estimator 96, kriteria pemisahan entropi, tiga sampel minimum untuk pemisahan, dan 95 daun maksimum di sebuah titik.

Salah satu fitur dari pemodelan *Random Forest* menggunakan Scikit-Learn adalah metode yang bisa digunakan untuk mencari seberapa besar pengaruh fitur-fitur atau variabel independen di pohon-pohon yang dimodelkan. Metode tersebut akan memberikan persentase dari kontribusi masing-masing variabel independen terhadap variabel dependen. Berikut adalah tabel yang menggambarkan lima pengaruh tertinggi dari fitur yang dipilih terhadap model *Random Forest* yang dilatih peneliti:

Tabel 4.4. Peranan Fitur dalam *Random Forest*

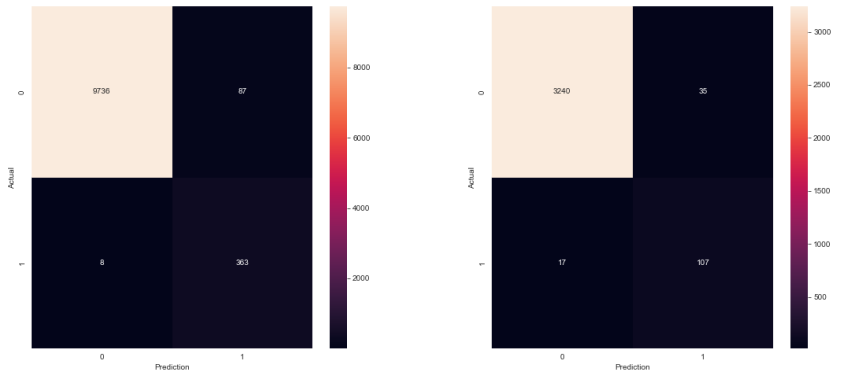
No.	Nama Fitur	Pengaruh (%)
1	VORP	19,1
2	PER	13,6
3	WS	13,1
4	OBPM	9,98
5	BPM	9,46

Tabel 4.4 menggambarkan lima fitur dengan pengaruh tertinggi, namun fitur-fitur lainnya juga masih memiliki peranan terhadap model dengan nilai pengaruh yang lebih kecil. Terlihat dari tabel tersebut, fitur-fitur yang memiliki pengaruh-pengaruh tertinggi adalah fitur yang memiliki korelasi Pearson mutlak yang cukup tinggi terhadap variabel dependen yang dapat dilihat di Tabel 4.2. Sama layaknya dengan model Regresi Logistik, peneliti mencari *threshold* terbaik untuk memaksimalkan nilai F1 untuk model *Random Forests*. *Threshold* terbaik untuk model ini adalah 0,7. Berikut adalah hasil prediksi dari data *training* dan *development* menggunakan *threshold* yang sudah ditentukan:



Gambar 4.5. Prediksi *Random Forests* Data *Training* (kiri) dan Data *Development* (kanan)

Jika dilihat dari Gambar 4.5, performa model berdasarkan *threshold* yang ditentukan dapat dinilai cukup bagus. Pemisahan dari probabilitas yang dihasilkan model terlihat sangat jelas di gambar tersebut dengan melihat garis yang berwarna hijau. Konklusi tersebut juga didukung dengan sedikitnya kelas yang tidak diklasifikasi model *Random Forest* dengan benar (dapat dilihat dari jumlah titik berwarna biru atau kelas bukan All-NBA dan titik berwarna oranye atau kelas All-NBA relatif dengan garis hijau). Selanjutnya peneliti melakukan analisis menggunakan *confusion matrix* untuk data *training* dan *development*, berikut adalah grafik dari kedua *confusion matrix* tersebut:



Gambar 4.6. *Confusion Matrix Random Forest* Data *Training* (kiri) dan Data *Development* (kanan)

Threshold yang sudah ditentukan untuk *Random Forest* menghasilkan hasil prediksi yang digambarkan melalui *confusion matrix* yang dapat dilihat di Gambar 4.6. Sama halnya dengan Regresi Logistik, algoritma *Random Forest* juga dievaluasikan menggunakan metrik-metrik yang sama. Berikut adalah metrik yang diperoleh untuk model *Random Forest*:

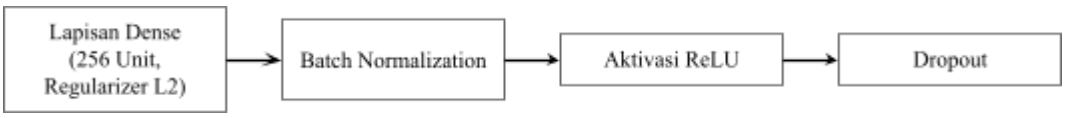
Tabel 4.5. Performa *Random Forest*

Metrik	*Training*	*Development*
Akurasi	0,991	0,985
Recall	0,978	0,863
Precision	0,807	0,753
F1	0,884	0,805

Tabel 4.5 menunjukkan bahwa model *Random Forest* mengalami sedikit *overfitting* terhadap data *training*. Hal tersebut terlihat dari perbedaan nilai-nilai metrik jika membandingkan data *training* dan data *development*. Namun sedikit *overfitting* untuk algoritma *Random Forest* merupakan suatu hal yang dapat dinilai wajar, sehingga tidak diperlukan perhatian khusus untuk mengubah *hyperparameter* model. Nilai akurasi dari algoritma ini sangat tinggi, bahkan hanya melakukan misklasifikasi terhadap 1% data *training* dan 1,5% data *development*. *Recall* dari model tersebut berada di nilai 97,8% dan 86,3%, terdapat perbedaan kurang lebih 11,5% untuk data *training* dan data *development*. Sebanyak 80,7% prediksi positif model merupakan pemain yang sebenarnya merupakan All-NBA di data *training*. Selisih dari nilai tersebut (*precision*) untuk data *testing* adalah 5,4%. Karena *recall* dan *precision* data *training* lebih tinggi, skor F1 dari dataset tersebut lebih tinggi dibandingkan data *development*.

4.3.4. *Neural Network*

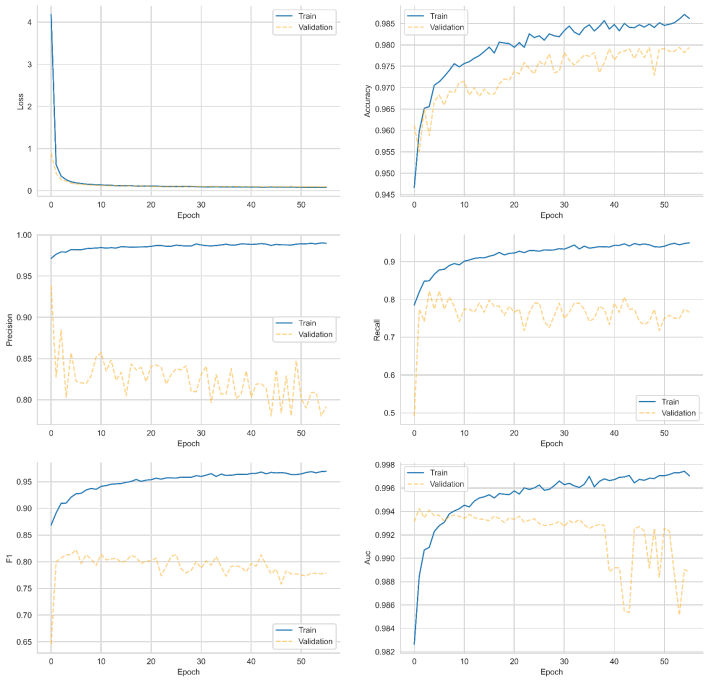
Sama halnya algoritma *Random Forest*, algoritma *Neural Network* tidak memiliki prasyarat yang perlu dilakukan. Pemodelan *Neural Network* dilakukan menggunakan Tensorflow 2. Peneliti menemukan bahwa model *Neural Network* terbaik dapat diperoleh dengan membagi model tersebut menjadi beberapa blok kecil. Skema dari sebuah blok tersebut dapat dilihat di Gambar 4.7 di bawah ini.



Gambar 4.7. Skema Blok *Neural Network*

Peneliti menggunakan teknik *regularization* untuk bisa melatih model yang lebih dalam dan mengurangi kemungkinan *overfitting* yang sudah dibahas di Bab 2.2.3. Peneliti menemukan bahwa *hyperparameter* λ untuk *regularization* L2 terbaik adalah 0,001. Untuk *dropout*, probabilitas terbaik adalah 0,6125. Setiap blok tersebut akan disusun menjadi sebuah *Neural Network* yang memiliki delapan lapisan yang terdiri atas beberapa blok-blok kecil seperti yang dapat dilihat di Gambar 4.7. Sebanyak tujuh blok tersebut akan digunakan dengan *hyperparameter* λ dan probabilitas yang sama. Lapisan kedelapan merupakan lapisan Dense dengan besar satu unit yang juga diberikan *regularization* L2 dengan nilai *hyperparameter* yang sama dengan tujuh blok lainnya. Lapisan tersebut juga memiliki aktivasi dengan fungsi Sigmoid untuk menghasilkan *output* probabilitas yang diinginkan. Peneliti juga menemukan penggunaan ide *residual connection* dapat meningkatkan performa dari model. Penggunaan *residual connection* digunakan di lapisan Dense keempat, keenam, dan kedelapan dengan menambahkan input dari lapisan pertama untuk lapisan keempat, ketiga untuk lapisan keenam, dan kelima untuk lapisan kedelapan. Hasil dari rancangan di atas adalah sebuah model dengan 408.577 parameter.

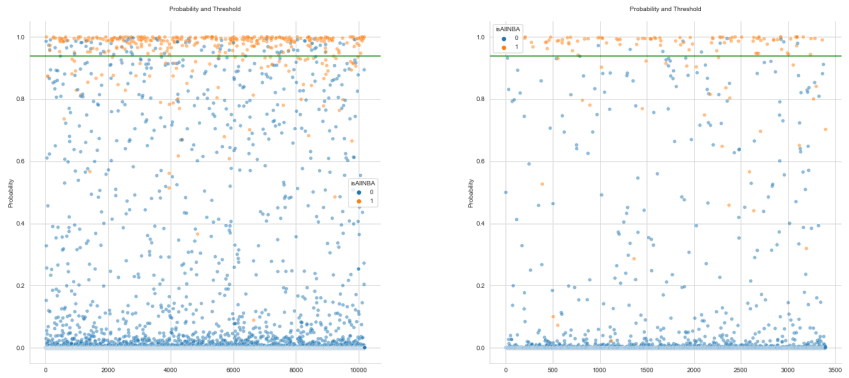
Fungsi *cost* yang digunakan adalah *binary cross entropy* dengan algoritma optimasi Adam. *Learning rate* yang digunakan akan mengecil seiring meningkatnya iterasi untuk memastikan kekonvergenan ke nilai yang dekat dengan minimum fungsi *cost*, atau disebut juga teknik *learning schedule* dengan tipe *inverse time decay*. *Hyperparameter* yang digunakan untuk teknik tersebut adalah *learning rate* awal bernilai 0,1, *decay rate* bernilai 0,35 dan *decay step* bernilai setiap 2 epoch. Model tersebut akan dilatih selama 200 epoch dengan *early stopping* untuk mengambil model yang memiliki nilai F1 yang terbaik.



Gambar 4.8. Hasil Pelatihan *Neural Network*

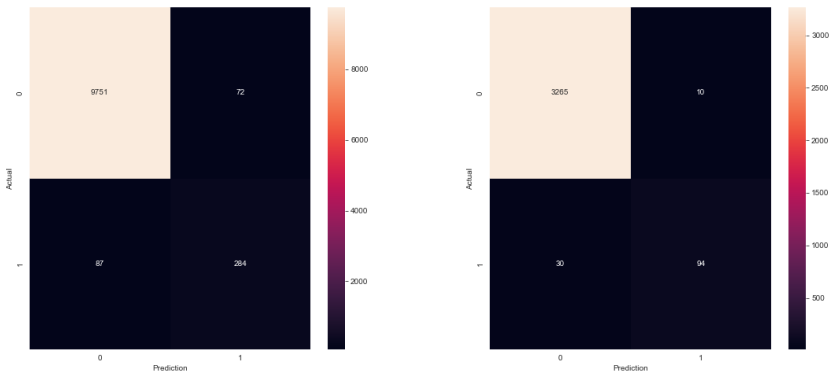
Gambar 4.8 merangkum hasil dari pelatihan model terbaik yang diperoleh oleh peneliti. Dapat dilihat bahwa selisih dari fungsi *cost* untuk data *training* dan hasil dari validasi terhadap data *development* sangatlah kecil dan nilai dari fungsi *cost* juga cukup kecil, sehingga model tidak memiliki isu *overfitting* yang besar. Akurasi dari model juga memiliki selisih yang kecil, sehingga memperkuat klaim sebelumnya. Menurut AUC, model yang digunakan dapat merepresentasikan model dengan baik. Untuk metrik seperti *recall*, *precision*, dan F1, metrik-metrik tersebut sangat dependen terhadap *threshold* yang digunakan.

Peneliti juga melakukan pencarian *threshold* terbaik dan menemukan bahwa *threshold* yang memaksimalkan nilai F1. Untuk model *Neural Network* adalah 0,94. Sama halnya dengan model-model sebelumnya, visualisasi dari *threshold* dapat dilakukan untuk memberikan gambaran kecil mengenai performa model. Berikut adalah hasil prediksi dari data *training* dan *development* menggunakan *threshold* yang sudah ditentukan:



Gambar 4.9. Prediksi *Neural Network* Data *Training* (kiri) dan Data *Development* (kanan)

Threshold dengan nilai 0,94 memiliki kemampuan yang baik untuk memisahkan kelas positif atau pemain All-NBA (titik berwarna oranye) dan kelas negatif atau pemain bukan All-NBA (titik berwarna biru). *Threshold* ini lebih tinggi dibandingkan dengan dua model sebelumnya namun data nilai tersebut dapat menghasilkan performa yang cukup baik. Selanjutnya peneliti melakukan analisis menggunakan *confusion matrix* untuk data *training* dan *development*, berikut adalah grafik dari kedua *confusion matrix* tersebut:



Gambar 4.10. *Confusion Matrix Neural Network* Data *Training* (kiri) dan Data *Development* (kanan)

Gambar 4.10 memperlihatkan hasil dari *confusion matrix* yang dihasilkan oleh model *Neural Network* saat menggunakan *threshold* yang sudah ditentukan, yaitu 0,94. Kedua *confusion matrix* di atas sudah menggambarkan *Confusion matrix* digunakan oleh peneliti untuk mengkalkulasikan metrik-metrik evaluasi model seperti layaknya model-model sebelumnya. Hasil dari kalkulasi tersebut dapat dilihat di Tabel 4.6 berikut ini.

Tabel 4.6. Performa *Neural Network*

Metrik	*Training*	*Development*
Akurasi	0,984	0,988
Recall	0,765	0,758
Precision	0,798	0,903
F1	0,781	0,825

Tabel 4.6 memberikan gambaran bahwa performa dari *Neural Network* lebih baik di data *development* dibandingkan data *training*. Dapat dilihat juga di tabel bahwa tidak terjadi *overfitting* di model ini karena performa yang mendekati untuk metrik-metrik yang diobservasi baik untuk data *training* maupun data *development* dan juga hasil dari pelatihan model yang dapat dilihat di Gambar 4.8. Akurasi dari model ini dapat dinilai sangat baik dengan nilai 98,4% untuk data *training* dan 98,8% untuk data *development*. Untuk nilai *recall*, performa dari model untuk memprediksi data yang sebenarnya adalah pemain All-NBA cukup baik, dengan nilai 76,5% untuk data *training* dan 75,8% untuk data *testing*. Metrik *precision* menunjukkan bahwa ada perbedaan performa yang drastis untuk model *Neural Network*. *Precision* untuk data *training* bernial 79,8% sementara data *development* 90,3%, sehingga terlihat bahwa mungkin data *development* lebih mudah untuk diprediksi model dibandingkan data *training*. Perbedaan yang signifikan di data *development* menyebabkan nilai dari skor F1 lebih tinggi untuk data tersebut (82,5%) dibandingkan data *training* (78,1%).

4.4. Perbandingan Model

Tabel 4.7. Performa Model di Data *Development*

Metrik	Regresi Logistik	*Random Forest*	*Neural Network*
Akurasi	0,984	0,985	0,988
Recall	0,79	0,863	0,758
Precision	0,778	0,753	0,903
F1	0,784	0,805	0,825

Untuk melakukan perbandingan dari performa model, peneliti membandingkan performa model terhadap data yang belum dilihat dari model, yaitu data *development*. Dapat dilihat melalui Tabel 4.7, metrik yang telah diseleksi peneliti untuk menguji model, yaitu skor F1, memiliki nilai tertinggi saat menggunakan model *Neural Network* jika dibandingkan dengan model lainnya. Model tersebut juga memiliki skor akurasi dan *precision* tertinggi di antara ketiga model tersebut. Akan tetapi, kelemahan dari model *Neural Network* adalah kemampuannya untuk memprediksi pemain yang sebenarnya positif di dalam dataset *development*. Karena model tersebut memiliki skor F1 terbesar, sesuai dengan objektif yang telah ditetapkan peneliti, model *Neural Network* adalah model yang terbaik untuk dataset yang digunakan.

Seperti yang tertera di Bab 3.5.4, peneliti menggunakan model terbaik untuk memprediksi pemain yang akan mendapatkan penghargaan All-NBA menurut model di musim yang telah berlalu. Tabel 4.8 adalah hasil prediksi dengan menggunakan data musim 2021-2022 yang diurutkan dari probabilitas terbesar ke terkecil.

Tabel 4.8. Hasil Prediksi *Neural Network* untuk Musim 2021-2022

Nama Pemain	Posisi	Probabilitas	Seleksi
Luka Dončić	G	0.991841	*First*
Trae Young	G	0.985421	*First*
Giannis Antetokounmpo	F	0.995235	*First*
Jayson Tatum	F	0.990297	*First*
Nikola Jokić	C	0.997815	*First*
Stephen Curry	G	0.96798	*Second*
Devin Booker	G	0.965536	*Second*
Kevin Durant	F	0.969836	*Second*
LeBron James	F	0.959276	*Second*
Joel Embiid	C	0.996623	*Second*
Ja Morant	G	0.961248	*Third*
Donovan Mitchell	G	0.950246	*Third*
DeMar DeRozan	F	0.942273	*Third*
Jimmy Butler *	F	0.932874	*Third*
Rudy Gobert	C	0.990094	*Third*

Seperti yang dapat dilihat di Tabel 4.8 di atas, ada 14 pemain yang memiliki probabilitas yang lebih tinggi dari *threshold* untuk model *Neural Network* yaitu 0,94. Menurut segi posisi yang dipilih, model hampir berhasil untuk memprediksi semua tim All-NBA namun hanya kehilangan satu pemain lagi di posisi *forward* untuk kategori All-NBA *Third Team*. Jika peneliti mematahkan *threshold* yang sudah ditentukan dan menyeleksi pemain berposisi *forward* berikutnya, pemain yang akan mendapatkan seleksi untuk masuk ke All-NBA *Third Team* adalah Jimmy Butler.

Per tanggal 24 Mei 2022, NBA sudah mengumumkan pemenang tim All-NBA dan beberapa penghargaan lainnya. Peneliti juga memiliki keyakinan yang besar bahwa model dapat digunakan untuk secara tidak langsung memprediksi *Most Valuable Player* atau MVP di NBA dengan cara memilih pemain dengan probabilitas tertinggi untuk diseleksi ke tim All-NBA. Untuk musim ini, pemain tersebut adalah Nikola Jokić, pemenang penghargaan tersebut di dunia nyata. [53]. Model bahkan berhasil untuk memprediksi pemain yang mendapatkan peringkat kedua dan ketiga untuk penghargaan MVP, yaitu Joel Embiid dan Giannis Antetokounmpo.

Untuk tim All-NBA, model yang digunakan peneliti berhasil memprediksi 12 dari 15 pemain yang mendapatkan penghargaan tersebut [54]. Pemain yang ada di hasil prediksi model yang tidak mendapatkan penghargaan tersebut adalah Donovan Mitchell, Jimmy Butler, dan Rudy Gobert. Ketiga pemain seharusnya mendapatkan penghargaan All-NBA adalah Chris Paul di posisi *guard* dengan probabilitas model 0.486975, Pascal Siakam di posisi *forward* dengan probabilitas model 0,843599, dan Karl-Anthony Towns di posisi *centre* dengan probabilitas model 0,932283.

4.5. Uji Hipotesis

Sebagai pengingat, peneliti membuktikan bahwa pemilihan pemenang All-NBA tidak dilakukan secara acak menggunakan uji binomial. Berikut adalah hipotesis yang digunakan oleh peneliti:

$$H_0 : \pi = 0,5$$

$$H_1 : \pi \neq 0,5$$

.Jika *p-value* kurang dari $\alpha = 0,05$, maka H_0 akan ditolak sehingga dapat dilihat bahwa pemilihan pemain All-NBA tidak dilakukan secara acak, dan sebaliknya juga berlaku. Seperti yang telah dibahas di Bab 3.5.7, peneliti akan menggunakan akurasi dari model terbaik yakni model *Neural Network* untuk melakukan uji binomial. $\Pr(X = 3359)$ dapat dicari dengan persamaan dibawah ini

$$\Pr(X = 3359) = \binom{3399}{3359} 0.5^{3359}(1 - 0.5)^{3399-3359} = 0 \quad \textbf{(4.3)}$$

Karena *p-value* dari uji binomial kurang dari α, maka H_0 akan ditolak, sehingga ada bukti statistik bahwa pemilihan pemain All-NBA tidak dilakukan secara acak, sehingga ada suatu pola tertentu yang digunakan oleh pemilih untuk mendapatkan pemain All-NBA. Pola dari pemilihan tersebut berhasil ditangkap dengan baik oleh model *Neural Network* yang dimiliki peneliti.

5. Penutup

5.1. Kesimpulan

Peneliti memulai penelitian ini dengan dua pertanyaan yang telah terjawab selama durasi penelitian ini. Ada beberapa metrik *advanced statistics* yang memiliki korelasi mutlak yang tinggi terhadap terpilihnya seorang pemain ke dalam tim All-NBA, seperti layaknya VORP, WS, OWS, PER, DWS, BPM, OBPM, dan WS/48. Karena adanya korelasi yang tinggi, tentu ada cara untuk menggabungkan metrik-metrik tersebut dalam sebuah model *machine learning* untuk bisa memprediksi peluang seorang pemain mendapatkan penghargaan tersebut kedepannya. Model yang peneliti temukan untuk bisa memprediksi dengan baik adalah *Neural Network* delapan lapisan yang pada akhirnya bisa memperoleh metrik performa model F1 sebesar 82.5%. Model tersebut berhasil memprediksi 12 dari 15 pemain yang mendapatkan penghargaan All-NBA di musim 2021-2022 dan juga berhasil secara tidak langsung memprediksi ketiga pemain yang mendapatkan nominasi dan pemenang dari *Most Valuable Player* di musim tersebut.

5.2. Limitasi dan Saran

Limitasi yang dihadapi oleh peneliti selama durasi penelitian adalah kurangnya data pemain yang mendapatkan penghargaan All-NBA, oleh sebab itu perlu adanya penggunaan algoritma *oversampling* untuk membuat contoh-contoh sintetis. Peneliti juga terbatas di segi komputasi.

Peneliti sangat menyarankan untuk memperkaya fitur yang digunakan dalam pembentukan model dan menggunakan metrik-metrik layaknya RAPTOR dari FiveThirtyEight dan masih banyak lagi. Penambahan jumlah data juga dapat membantu meningkatkan performa model karena menambahkan contoh-contoh baru yang dapat digunakan untuk melatih model, misalnya menggunakan musim sebelum 1988-1989. Peneliti tidak menggunakan data sebelum tahun 1988-1989 karena pembagian penghargaan sebelum musim tersebut hanya diberikan kepada 10 pemain untuk dijadikan dua tim.

Ucapan Terimakasih

Peneliti ingin mengucapkan terima kasih sebesar-besarnya kepada Kelvin Andersen dari Institut Teknologi Sepuluh Nopember atas inspirasi dan ide-ide yang diberikan selama jalannya penelitian ini. Seluruh hasil analisis penelitian ini dapat diakses di *repository* ini.

Daftar Pustaka

[1] Sadler, M., Regan, N., & Kasparov, G. (2019). *Game Changer: AlphaZero's Groundbreaking Chess Strategies and the Promise of AI*. New In Chess.

[2] Stanchev, N. (2021). *Formula 1 Race Strategy Simulator GeekF1 - Nikola Stanchev*. Medium. https://medium.com/@nikolastanchev777/formula-1-race-strategy-simulator-geekf1-ac704ff62e1a

[3] Schuhmann, J. (2021). *NBA's 3-point revolution: How 1 shot is changing the game*. NBA.Com. https://www.nba.com/news/3-point-era-nba-75

[4] Chandrasekhar, S. (2022). *Analytics Have Transcended Sports But Is It Ruining The Art and Purpose of Sports?* Medium. https://sandeep-chandrasekhar.medium.com/analytics-have-transcended-sports-but-is-it-ruining-the-art-and-purpose-of-sports-3f5c5059ae87

[5] Britannica. (n.d.-a). *National Basketball Association | History & Facts*. Encyclopedia Britannica. https://www.britannica.com/topic/National-Basketball-Association

[6] Weinstein, B. (2016). *Warriors' Stephen Curry and Cavaliers' LeBron James headline 2015–16 All-NBA First Team*. NBA.Com: NBA Communications. https://pr.nba.com/2015-16-all-nba-teams/

[7] Larsen, A. (2020). *My NBA Awards ballot, Part 2: All-NBA, All-Defense, All-Rookie*. The Salt Lake Tribune. https://www.sltrib.com/sports/jazz/2020/07/27/andy-larsen-my-nba-awards/

[8] IBM Cloud Education. (2021). *Machine Learning*. IBM. https://www.ibm.com/cloud/learn/machine-learning

[9] Brown, S. (2021). *Machine learning, explained*. MIT Sloan. https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained

[10] DeepLearningAI. (2017). *Why is deep learning taking off?* YouTube. https://www.youtube.com/watch?v=xflCLdJh0n0&feature=youtu.be

[11] Subasi, A. (2020). *Practical Machine Learning for Data Analysis Using Python* (1st ed.). Academic Press.

[12] Hosmer, D. W., Jovanovic, B., & Lemeshow, S. (1989). Best Subsets Logistic Regression. *Biometrics*, *45*(4), 1265. https://doi.org/10.2307/2531779

[13] Pant, A. (2021). *Introduction to Logistic Regression - Towards Data Science*. Medium. https://towardsdatascience.com/introduction-to-logistic-regression-66248243c148

[14] Stoltzfus, J. C. (2011). Logistic Regression: A Brief Primer. *Academic Emergency Medicine*, *18*(10), 1099–1104. https://doi.org/10.1111/j.1553-2712.2011.01185.x

[15] IYKRA. (2018). *Mengenal Decision Tree dan Manfaatnya - Iykra*. Medium. https://medium.com/iykra/mengenal-decision-tree-dan-manfaatnya-b98cf3cf6a8d

[16] Grömping, U. (2009). Variable Importance Assessment in Regression: Linear Regression versus Random Forest. *The American Statistician*, *63*(4), 308–319. https://doi.org/10.1198/tast.2009.08199

[17] Scikit-Learn. (n.d.). *1.10. Decision Trees*. https://scikit-learn.org/stable/modules/tree.html#tree-algorithms-id3-c4-5-c5-0-and-cart

[18] Ayuya, C. (2021). *Entropy and Information Gain to Build Decision Trees in Machine Learning*. Engineering Education (EngEd) Program | Section. https://www.section.io/engineering-education/entropy-information-gain-machine-learning/

[19] Mueller, J. P., & Massaron, L. (2019). *Deep Learning For Dummies* (1st ed.). For Dummies.

[20] DeepLearningAI. (2017). Normalizing Activations in a Network (C2W3L04). YouTube. https://www.youtube.com/watch?v=tNIpEZLv_eg

[21] Tilawah, S. (2021). *Adam Optimizer - Sari Tilawah*. Medium. https://medium.com/@saritilawah9/adam-optimizer-80cc267522af

[22] DeepLearningAI. (2017). *Adam Optimization Algorithm*. YouTube. https://www.youtube.com/watch?v=JXQT_vxqwIs

[23] Golub, G. H., & Loan, C. V. F. (1996). *Matrix Computations (Johns Hopkins Studies in Mathematical Sciences)(3rd Edition)* (3rd ed.). Johns Hopkins University Press.

[24] Wan, L., Zeiler, M., Zhang, S., & Lecun, Y. (2013). Regularization of Neural Networks using DropConnect. *Conference: International Conference on Machine Learning*, *1*. http://proceedings.mlr.press/v28/wan13.pdf

[25] Brownlee, J. (2021). *SMOTE for Imbalanced Classification with Python*. Machine Learning Mastery. https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

[26] Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic Minority Over-sampling Technique. *Journal of Artificial Intelligence Research*, *16*, 321–357. https://doi.org/10.1613/jair.953

[27] Lee, W. M. (2019). *Python Machine Learning*. Wiley.

[28] Uudmae, J. (2016). Predicting NBA Game Outcomes. *CS229 Final Project*. http://cs229.stanford.edu/proj2017/final-reports/5231214.pdf

[29] Albert, A. A., de Mingo López, L. F., Allbright, K., & Gómez Blas, N. (2021). A Hybrid Machine Learning Model for Predicting USA NBA All-Stars. *Electronics*, *11*(1), 97. https://doi.org/10.3390/electronics11010097
[30] Wang, J., & Fan, Q. (2021). Application of Machine Learning on NBA Data Sets. *Journal of Physics: Conference Series*, *1802*(3), 032036. https://doi.org/10.1088/1742-6596/1802/3/032036
[31] Dos Santos, T., Wang, C., Carlsson, N., & Lambrix, P. (2021). Predicting Season Outcomes for the NBA. https://www.ida.liu.se/research/sportsanalytics/projects/conferences/MLSA21-basketball/MLSA21-paper.pdf
[32] Cheng, G., Zhang, Z., Kyebambe, M., & Kimbugwe, N. (2016). Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle. *Entropy*, *18*(12), 450. https://doi.org/10.3390/e18120450
[33] Eric Scot Jones, & Rhonda C. Magel. (2016). Predicting Outcomes of NBA Basketball Games. *Journal of Advance Research in Business Management and Accounting (ISSN: 2456–3544)*, *2*(5), 01–13. https://doi.org/10.53555/nnbma.v2i5.99
[34] Wilkens, S. (2021). Sports prediction and betting models in the machine learning age: The case of tennis. *Journal of Sports Analytics*, *7*(2), 99–117. https://doi.org/10.3233/jsa-200463
[35] Migliorati, M. (2021). Features selection in NBA outcome prediction through Deep Learning. https://arxiv.org/abs/2111.09695
[36] Arikunto, S. (2006). *Metode Penelitian Kualitatif*. Bumi Aksara.
[37] Bhandari, P. (2022). *An introduction to correlational research*. Scribbr. https://www.scribbr.com/methodology/correlational-research/
[38] Basketball Reference. (n.d.). *Basketball Statistics and History*. Basketball-Reference.Com. https://www.basketball-reference.com/
[39] Glen, S. (2013). *What is a Population in Statistics?* Statistics How To. https://www.statisticshowto.com/what-is-a-population/
[40] Britannica. (n.d.). *Sampling | statistics*. Encyclopedia Britannica. https://www.britannica.com/science/sampling-statistics
[41] Salmaa, S. (2021). *Teknik Pengambilan Sampel: Pengertian, Jenis-Jenis, dan Contohnya*. Penerbit Deepublish. https://penerbitdeepublish.com/teknik-pengambilan-sampel
[42] Heffernan, B. (2022). *How Much Is A 10-Day Contract In The NBA? (Salary Data Here!)*. Dunk or Three. https://dunkorthree.com/nba-10-day-contract/
[43] Dietzel, A. (2021). *A Step-by-Step Guide to Web Scraping NBA Data With Python, Jupyter, BeautifulSoup and Pandas*. Medium. https://betterprogramming.pub/a-step-by-step-guide-to-web-scraping-nba-data-with-python-jupyter-beautifulsoup-and-pandas-7e2d334d4195
[44] Cherry, K. (2022). *Why the Dependent Variable Is So Important to Valid Experiments*. Verywell Mind. https://www.verywellmind.com/what-is-a-dependent-variable-2795099
[45] Helmenstine, A. (2021). *Independent Variable Definition and Examples*. ThoughtCo. https://www.thoughtco.com/definition-of-independent-variable-605238
[46] Basketball Reference. (n.d.-b). *Glossary*. Basketball-Reference.Com. https://www.basketball-reference.com/about/glossary.html
[47] Almazov, A. (2021). *FTR (Free Throw Rate)*. Alvin Almazov: free sport tips. https://alvin-almazov.com/basketball-eng/ftr-free-throw-rate/
[48] Captain Calculator. (n.d.). *Basketball Calculators*. https://captaincalculator.com/sports/basketball/
[49] Rumsey, D. J. (2021). *Statistics II For Dummies* (2nd ed.). For Dummies.
[50] Faraway, J. J. (2002). *Practical Regression and Anova Using R.* Amsterdam University Press.
[51] Glen, S. (2015). Variance Inflation Factor. Statistics How To. https://www.statisticshowto.com/variance-inflation-factor/
[52] Ambielli, B. (2017). *Gini Impurity (With Examples)*. Bambielli's Blog. https://bambielli.com/til/2017-10-29-gini-impurity/
[53] Weinstein, B. (2022). Nuggets' Nikola Jokić wins 2021–22 Kia NBA Most Valuable Player Award. NBA.Com: NBA Communications. https://pr.nba.com/nikola-jokic-wins-2021-22-kia-nba-mvp-award/
[54] Weinstein, B. (2022). Giannis Antetokounmpo, Luka Dončić, Nikola Jokić, Devin Booker and Jayson Tatum selected to 2021–22 Kia All-NBA First Team. NBA.Com: NBA Communications. https://pr.nba.com/2021-22-kia-all-nba-team/

Lampiran

Lampiran 1. Ringkasan Penelitian Serupa

Nama Artikel	Peneliti	Objek Penelitian	Metode	Hasil Penelitian
CS229 Final Project: Predicting NBA Game Outcomes	Uudmae (2016)	Statistik *per game* tim untuk memprediksi hasil pertandingan dari tahun 2013 - 2016	Regresi Linear, SVM, dan NNR	● SVM dengan akurasi 62,07% ● Regresi Linear dengan akurasi 63,75% ● NNR dengan akurasi 64,95%
A Hybrid Machine Learning Model for Predicting USA NBA All-Stars	Albert, Lopez, Allbright, Blas (2022)	Memprediksi pemenang penghargaan All-Star menggunakan statistik *per game* individu dari tahun 1980 - 2021	*Random Forest classifier*, *Adaboost classifier*, *MLP classifier*, hybrid dari ketiga model	● *Random Forest* dengan *sensitivity* 52,4% ● Adaboost dengan *sensitivity* 61,9% ● MLP dengan *sensitivity* 61,9% ● Hybrid dengan *sensitivity* 81%
Application of Machine Learning on NBA Data Sets	Wang, Fan (2020)	Memprediksi pemain All-Star, hasil dari *playoff*, membuktikan kebenaran atas *hot streak*, dan melihat tren di NBA di tahun 2019	Random forest classifier, Decision Tree, KNN classifier, Gradient Boosting, Regresi Linear, Regresi Logistik, dan PCA	● Model terbaik untuk memprediksi All-Star adalah Regresi Logistik dengan akurasi 97,9% ● Data terbaik untuk mencari hasil *playoff* adalah data statistika *advanced* dengan model *gradient boosting* dengan akurasi 92% ● *Hot streak* itu tidak ada
Predicting Season Outcomes for the NBA	Dos Santos, Wang, Carlsson, Lambrix (2021)	Performa tim di akhir tahun dan prediksi hasil musim menggunakan data individu, boxscore, dan masih banyak lagi dari tahun 2008 - 2018.	LSVM, *Random forest*, dan MLP	Model yang terbaik yaitu Random Forest memiliki akurasi 69.88%.
Predicting the Outcome of NBA Playoffs Based on the Maximum Entropy Principle	Cheng, Zhang, Kyebambe, Kimbugwe (2016)	Pembentukan model *maximum entropy* untuk memprediksi hasil *playoff* tahun 2007 hingga 2015	Naive Bayes, BP NN, Regresi Logistik, model buatan sendiri	Akurasi dari model yang dibuat melampaui akurasi algoritma konvensional dengan akurasi 74,4%.

Predicting Outcomes of NBA Basketball Games	Jones (2016)	Memprediksi hasil pertandingan dan *points spread differential* dengan statistika tim tahun 2008 hingga 2011	Regresi Linear untuk PSD dan logistik untuk hasil pertandingan	Model PSD dengan akurasi 94% dan model kemenangan dengan akurasi 88%
Sports prediction and betting models in the machine learning age: The case of tennis	Wilkens (2021)	Memprediksi hasil dari pertandingan tenis menggunakan data ATP, WTA, ITF, dan *grandslam*	Regresi Logistik, NN, Random Forest, GBM, dan SVM	Model terbaik yang ditemukan beliau adalah GBM dengan akurasi 69,1%.
Features selection in NBA outcome prediction through Deep Learning	Migliorati (2021)	Membandingkan hasil prediksi pertandingan menggunakan model NN dengan fitur ELO dan *boxscore* di tahun 2005-2020	*Neural Network*	Model terbaik diperoleh menggunakan fitur ELO dengan akurasi 70,53%. Model ELO lebih konsisten dibanding *boxscore*

Lampiran 2. Daftar Variabel Independen yang Penelitian

No.	Variabel Independen	Penjelasan Singkat
1	Player	Nama dari pemain yang bersangkutan
2	Pos	Posisi yang dimainkan oleh pemain, ada 5 posisi yaitu PG, SG, SF, PF, dan C
3	Age	Umur dari pemain di musim tersebut
4	Tm	Nama tim dimana pemain tersebut bermain
5	G	Jumlah pertandingan yang dimainkan oleh pemain di musim tersebut
6	TMP	Total dari menit yang dimainkan oleh pemain dalam satu musim
7	PER	Singkatan dari *Player Efficiency Rating* yang dibuat oleh John Hollinger. Perhitungan PER dapat dilihat melalui situs ini.
8	TS%	Persentase *true shooting*. Dihitung dengan persamaan $\mathrm{TS\%} = \frac{\mathrm{PTS}}{2(\mathrm{FGA}+0.44\ \mathrm{FTA})}$
9	3PAR	Singkatan dari *three point attempt ratio.* Hasil pembagian dari 3PA dan FGA

10	FTR	Singkatan dari *free throw rate*. Merupakan rasio dari FTA dan FGA
11	ORB%	Singkatan dari *offensive rebound percentage* yang digunakan untuk mengestimasi rata-rata *offensive rebound* yang diperoleh pemain selama ia bermain per pertandingan. $$\text{ORB\%} = \frac{\text{ORB} \cdot \frac{\text{TeamMP}}{5}}{\text{MP} \cdot (\text{TeamORB} + \text{OpponentDRB})} \cdot 100$$
12	DRB%	Singkatan dari *defensive rebound percentage* yang digunakan untuk mengestimasi rata-rata *defensive rebound* yang diperoleh pemain selama ia bermain per pertandingan. $$\text{DRB\%} = \frac{\text{DRB} \cdot \frac{\text{TeamMP}}{5}}{\text{MP} \cdot (\text{TeamDRB} + \text{OpponentORB})} \cdot 100$$
13	TRB%	Singkatan dari *total rebound percentage* yang digunakan untuk mengestimasi rata-rata *total rebound* yang diperoleh pemain selama ia bermain per pertandingan. $$\text{TRB\%} = \frac{\text{TRB} \cdot \frac{\text{TeamMP}}{5}}{\text{MP} \cdot (\text{TeamTRB} + \text{OpponentTRB})} \cdot 100$$
14	AST%	Singkatan dari *assist percentage*. Metrik ini mengukur seberapa banyak tembakan tim yang merupakan *assist* dari pemain tersebut. $$\text{AST\%} = 100 \cdot \frac{\text{AST}}{\frac{5 \text{ MP}}{\text{TeamMP}} \cdot \text{TeamFG} - \text{FG}}$$
15	STL%	Singkatan dari *steal percentage*. Metrik ini mengukur seberapa banyak *possession* musuh yang berakhir dengan *steal* dari pemain ini $$\text{STL\%} = 100 \cdot \frac{\text{STL} \cdot \frac{\text{TeamMP}}{5}}{\text{MP} \cdot \text{OpponentPossession}}$$
16	BLK%	Singkatan dari *block percentage*. Metrik ini mengukur persentase dari tembakan 2 poin lawan yang diblokir oleh seorang pemain. $$\text{BLK\%} = 100 \cdot \frac{\text{BLK} \cdot \frac{\text{TeamMP}}{5}}{\text{MP} \cdot (\text{OpponentFGA} - \text{Opponent3PA})}$$
17	TOV%	Singkatan dari *turnover percentage*. Metrik ini mengukur seberapa banyak *possession* yang berakhir dengan *turnover* per 100 *plays* (*plays* artinya adalah skema). $$\text{TOV\%} = 100 \cdot \frac{\text{TOV}}{\text{FGA} + 0.44\ \text{FTA} + \text{TOV}}$$

18	USG%	Singkatan dari *usage percentage* atau sering disebut *usage rate*. Metrik ini menghitung seberapa banyak *plays* dari sebuah tim yang digunakan pemain saat ia bermain. $$\text{USG\%} = 100 \cdot \frac{(\text{FGA} + 0.44\,\text{FTA} + \text{TOV})(\frac{\text{Team MP}}{5})}{\text{MP} \cdot (\text{Team FGA} + 0.44\,\text{Team FTA} + \text{Team TOV})}$$
19	OWS	Singkatan dari *offensive win share*. Perhitungan variabel ini dapat dilihat melalui situs ini.
20	DWS	Singkatan dari *defensive win share*. Perhitungan variabel ini dapat dilihat melalui situs ini.
21	WS	Singkatan dari *win share*. Metrik ini mengkalkulasikan seberapa banyak kemenangan pertandingan yang merupakan kontribusi dari pemain ini. Perhitungan variabel ini dapat dilihat melalui situs ini.
22	WS/48	Singkatan dari *win share* per 48. Metrik ini mengkalkulasikan seberapa banyak kemenangan pertandingan yang merupakan kontribusi dari pemain ini per 48 menit (satu pertandingan NBA tanpa jeda). Perhitungan variabel ini dapat dilihat melalui situs ini.
23	OBPM	Singkatan dari *Offensive Box-Plus-Minus*. Merupakan metrik yang mengestimasi kontribusi pemain saat pemain ini menyerang. Penjelasan lebih detail dapat dilihat melalui situs ini.
24	DBPM	Singkatan dari *Defensive Box-Plus-Minus*. Merupakan metrik yang mengestimasi kontribusi pemain saat pemain ini bertahan. Penjelasan lebih detail dapat dilihat melalui situs ini.
25	BPM	Singkatan dari *Box-Plus-Minus*. Merupakan metrik yang mengestimasi kontribusi pemain saat pemain ini bermain. Penjelasan lebih detail dapat dilihat melalui situs ini.
26	VORP	Singkatan dari *Value over Replacement Player*. Metrik ini mengukur seberapa besar kontribusi seorang pemain jika dibandingkan pemain pengganti. Pemain pengganti sendiri didefinisikan sebagai pemain dengan kontrak minimum dan tidak merupakan pemain yang berada di dalam rotasi tim. Pemain pengganti juga dapat didefinisikan sebagai pemain dengan $\text{BPM} \leq -2$. Penjelasan lebih detail dapat dilihat melalui situs ini.
27	Year	Musim statistika ini dicapai oleh pemain. Apabila Year menunjukan suatu tahun, musim yang dimaksud adalah musim Year-1 hingga Year.