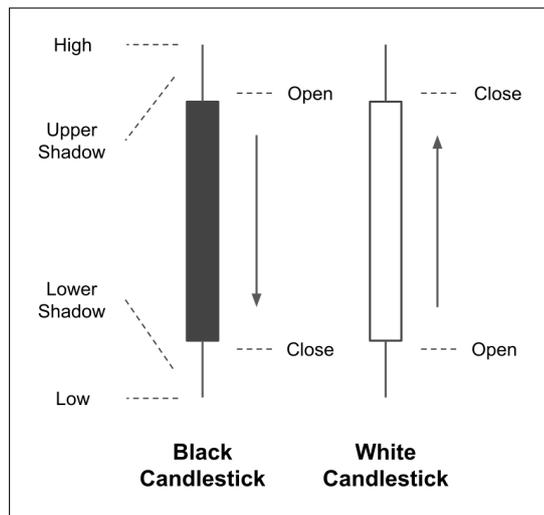


BAB 2

LANDASAN TEORI

2.1 Japanese Candlestick

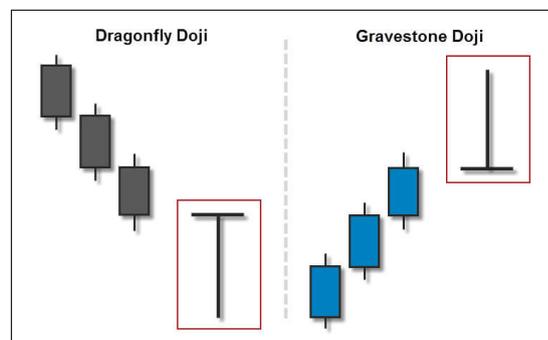
Japanese candlestick atau bisa disebut juga *candlestick chart*, adalah salah satu alat teknik analisis yang dapat merangkum harga *open*, *high*, *low*, *close* sekaligus ke dalam sebuah *bar chart* dimana *chart* per satu bar melambangkan kondisi saham selama 1 hari (Jearanaitanakij dan Passaya, 2019). Pola yang ada pada *candlestick chart* dapat membantu untuk memberikan petunjuk harga di masa depan menurut (Thammakesorn dan Sornil, 2019). Bentuk *candlestick* bar dapat dilihat pada Gambar 2.1. *Candlestick* hitam menggambarkan kondisi saham sedang *bearish* atau turun sedangkan *candlestick* putih menggambarkan kondisi saham sedang *bullish* atau naik.



Gambar 2.1. *Japanese candlestick*
(Chen et al., 2020)

Pola doji umumnya dapat dilihat berdasarkan keadaan satu *candle*. Pola ini terbentuk ketika *close price* bernilai sama atau hampir sama dengan *open price* (Noertjahyana et al., 2019). Pola *dragonfly doji* memiliki badan *candle* yang sangat

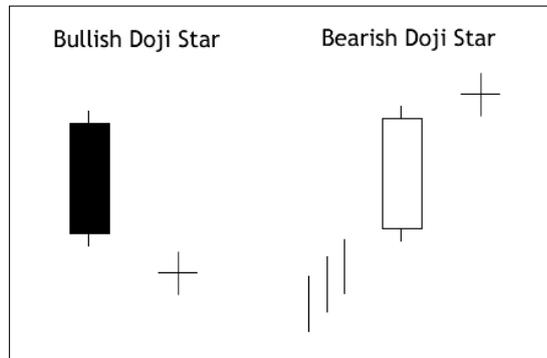
kecil atau tidak memiliki badan *candle* sama sekali pada bagian atas *candle* dan tidak memiliki *upper shadow* namun memiliki *lower shadow* yang panjang (Jain dan Jain, 2016). Sedangkan Pola *gravestone doji* memiliki badan *candle* yang sangat kecil atau tidak memiliki badan *candle* sama sekali pada bagian bawah *candle* dan tidak memiliki *lower shadow* namun memiliki *upper shadow* yang panjang (Jain dan Jain, 2016). Pola *dragonfly doji* akan mengubah pergerakan nilai saham yang sebelumnya turun menjadi naik, sementara *gravestone doji* akan mengubah pergerakan nilai saham yang sebelumnya naik menjadi turun (Jain dan Jain, 2016). Gambar 2.2 merupakan contoh pola *dragonfly* dan *gravestone doji*.



Gambar 2.2. Dragonfly dan gravestone doji (FXDayJob, 2016)

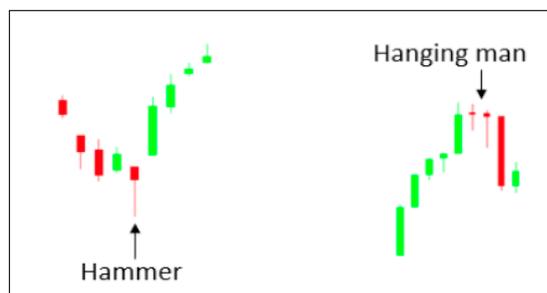
Selanjutnya, pola *bullish doji star* dan *bearish doji star* juga merupakan bagian dari *candle doji*. Namun, kedua jenis *candle doji* ini dilihat berdasarkan keadaan dari dua *candle* dan tren *candlestick* (Hu et al., 2019). *Bullish doji star* dimulai dengan adanya pergerakan tren yang turun serta *candle* hitam yang panjang dengan kondisi *closing price* di hari pertama lebih besar dibandingkan dengan *opening* dan *closing price* pada *candle doji* di hari kedua (Hu et al., 2019). Sementara itu, *bearish doji star* dimulai dengan adanya pergerakan tren yang naik serta *candle* putih yang panjang dan adanya *gap* antara badan *candle* di hari pertama dan kedua yang terbentuk ketika badan *candle* bagian bawah hari kedua lebih besar dibandingkan badan *candle* bagian atas hari pertama (Hu et al., 2019). *Doji candle* pada *bullish doji star* dan *bearish doji star* tidak boleh memiliki *shadows* yang

terlalu panjang (Hu et al., 2019). Contoh pola *bullish doji star* dan *bearish doji star* dapat dilihat pada Gambar 2.3.



Gambar 2.3. Bullish dan bearish doji star (Stock Market Watch, n.d.)

Kemudian, Pola *hammer* dan *hanging man* dapat dilihat dari kondisi satu *candle* dan bentuknya terlihat sama. Keduanya memiliki badan *candle* yang kecil dengan *upper shadows* yang pendek atau tidak ada sama sekali dan *lower shadows* yang panjang (Pongsena et al., 2018). Perbedaannya adalah pola *hammer* akan membalikkan keadaan nilai saham yang sebelumnya turun menjadi naik dan pola *hanging man* membalikkan keadaan nilai saham yang sebelumnya naik menjadi turun. Gambar 2.4 merupakan contoh pola *hammer* dan *hanging man* serta arah balik sinyal dari pola.



Gambar 2.4. Hammer dan hanging man (Jearanaitanakij dan Passaya, 2019)

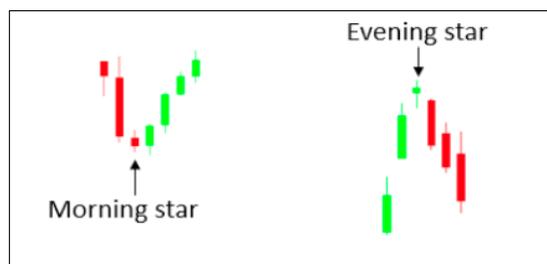
Pola *candlestick* berikutnya yaitu *bearish engulfing pattern* dan *bullish engulfing pattern* terbentuk dengan melihat keadaan *candle* selama dua hari. *bearish engulfing pattern* akan memberikan sinyal keadaan nilai yang sebelumnya

naik menjadi turun, sedangkan *bullish engulfing pattern* memberikan sinyal keadaan nilai yang sebelumnya turun menjadi naik (Heinz et al., 2021). Gambar 2.5 merupakan contoh pola *bearish engulfing pattern* dan *bullish engulfing pattern* serta arah balik sinyal dari pola.



Gambar 2.5. Bearish dan bullish engulfing (Jearanaitanakij dan Passaya, 2019)

Terakhir, pola *morning star* dan *evening star* terbentuk berdasarkan keadaan *candle* selama tiga hari. *Morning star* memberikan sinyal keadaan nilai yang sebelumnya turun menjadi naik sehingga memberikan sinyal keadaan market akan menjadi *bullish*, sedangkan *evening star* memberikan sinyal keadaan nilai yang sebelumnya naik menjadi turun dan memberikan sinyal keadaan market akan menjadi *bearish* (Horton, 2009). Gambar 2.6 merupakan contoh pola *morning star* dan *evening star* serta arah balik sinyal dari pola.



Gambar 2.6. Morning dan evening star (Jearanaitanakij dan Passaya, 2019)

2.2 TA-Lib

TA-Lib merupakan sebuah open source yang berisikan *library* untuk membantu melakukan *Technical Analysis*. TA-Lib memiliki banyak *function* yang

dapat digunakan dalam analisis data dan sudah banyak digunakan oleh *trading software developer*. Salah satu kemampuan dari TA-Lib adalah dapat melakukan *pattern recognition* terhadap pola candlestick. Proses *pattern recognition* pada TA-Lib dilakukan dengan mengidentifikasi *candlestick pattern* yang diinginkan. Apabila terdapat banyak *candlestick pattern* yang ingin ditemukan, maka function harus dipanggil sebanyak *candlestick pattern* yang diinginkan. Untuk setiap pola yang dipanggil dengan *function* yang ada akan mengembalikan nilai positif untuk pola yang teridentifikasi *bullish*, negatif pola yang teridentifikasi *bearish*, dan 0 apabila tidak terklasifikasi pada pola yang diinginkan (DUVINAGE et al., 2013).

2.3 Data Preprocessing

Berikut ini adalah hal-hal yang dilakukan dalam tahap data *preprocessing*.

1. Normalisasi Data

Dataset yang digunakan adalah data *time series* yang berbentuk numerik, dimana rentang nilai dari setiap atributnya berbeda-beda. Oleh karena itu, dibutuhkan normalisasi data terhadap *dataset* yang akan digunakan agar rentang nilai dari setiap atribut yang digunakan memiliki skala yang sama. Metode normalisasi data yang akan digunakan pada penelitian ini adalah metode *Min-Max Normalization*, untuk membantu melakukan transformasi linear terhadap *dataset* yang digunakan, dengan rentang skala 0 - 1 (Singh dan Singh, 2020).

$$x_{scaled} = \frac{x - min_x}{max_x - min_x} \quad (2.1)$$

2. Random Undersampling

Random Undersampling akan menghilangkan data terbanyak menjadi sesuai dengan banyak sampel yang dibutuhkan atau menjadi sama rata dengan data

sampel yang minoritas. *Random Undersampling* dilakukan agar model yang dibuat tidak berat sebelah hanya berpihak terhadap data yang mayoritas karena data mayoritas juga penting dalam melakukan klasifikasi (Gong dan Kim, 2017).

3. Synthetic Minority Over-sampling Technique (SMOTE)

Metode SMOTE digunakan untuk menghasilkan sampel baru pada kelas minoritas agar jumlah sampel data setiap kelas menjadi seimbang. Sampel baru yang dihasilkan ini berdasarkan nilai dari tetangga terdekatnya (Elreedy dan Atiya, 2019). Umumnya terdapat dua parameter yang dibutuhkan untuk metode ini, yaitu N (jumlah *oversampling* yang dibutuhkan) dan k (jumlah tetangga terdekat yang ditentukan untuk membuat sampel *synthetic* (Khadijah et al., 2018)).

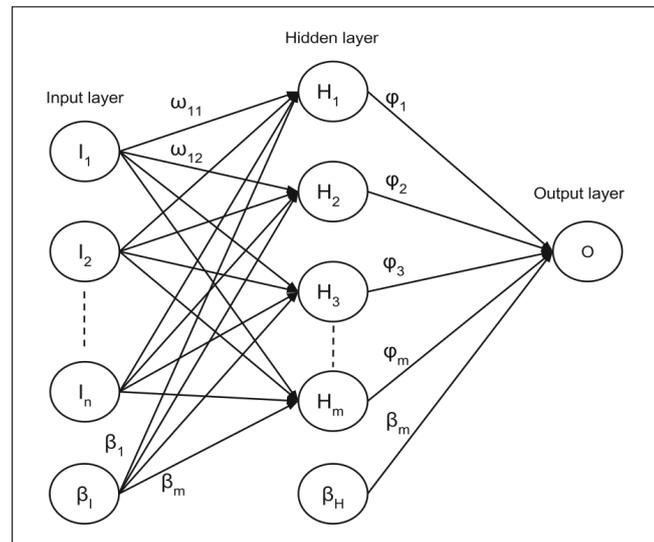
4. One-Hot Encoding

One-Hot Encoding digunakan untuk mengubah data yang masih berbentuk *categorical* menjadi bentuk numerik sehingga dapat diproses oleh komputer. Bentuk numerik yang dihasilkan biasanya dalam bentuk binary atau vektor sebanyak jumlah variabel yang ada.

2.4 Feedforward Neural Network

Feedforward Neural Network (FNN) merupakan salah satu arsitektur dari *Artificial Neural Network*. FNN atau bisa juga disebut dengan *perceptron network* adalah sebuah jaringan saraf dimana koneksi antar *neuron* tidak membentuk sebuah siklus terarah (Fuangkhn, 2014). Secara umum, sebuah *neural network* memiliki setidaknya tiga *layer* yaitu *input layer*, *hidden layer*, dan *output layer*. *Input layer* adalah sebuah *layer* pertama yang akan dilalui oleh parameter yang diinput untuk diproses. Selanjutnya pada *hidden layer* akan dilakukan komputasi yang berada diantara *input layer* dan *output layer*. Terakhir, *output layer* merupakan *layer* yang

akan menghasilkan *output final*. Gambar 2.7 merupakan arsitektur sederhana dari FNN dengan satu *hidden layer*.



Gambar 2.7. Arsitektur FNN dengan satu hidden layer (Faris et al., 2016)

Setiap *layer* pada FNN memiliki sebuah fungsi aktivasi, dimana setiap *layer* bisa memiliki fungsi aktivasi yang berbeda-beda ataupun sama. Fungsi aktivasi ini sangat penting pada sebuah *neural network* untuk membantu dalam pelatihan dan pembelajaran dalam melakukan pemetaan terhadap input dan output yang akan dihasilkan (Sharma dan Sharma, 2020). Fungsi aktivasi akan mengubah output pada setiap neuron, sesuai dengan jenis aktivasi yang digunakan, dan menghasilkan nilai baru untuk di transfer pada layer berikutnya. Pemilihan fungsi aktivasi disesuaikan dengan kebutuhan permasalahan yang ingin diselesaikan. Dalam permasalahan klasifikasi, biasanya fungsi aktivasi softmax digunakan pada *output layer*. Aktivasi softmax akan memperhitungkan probabilitas untuk masing-masing kelas dengan *range output* diantara 0 hingga 1. Berikut rumus fungsi aktivasi softmax.

$$softmax(x)_i = \frac{exp(x_i)}{\sum_j exp(x_j)} \quad (2.2)$$

2.5 Evaluasi Performa

Evaluasi performa dari algoritma Feedforward Neural Network pada penelitian ini ditampilkan dengan *confusion matriks*. *Confusion matriks* merupakan tabel yang berisikan informasi mengenai perbandingan model hasil uji coba klasifikasi yang telah dilakukan terhadap hasil klasifikasi sebenarnya. Nilai yang dihitung adalah *accuracy*, *precision*, *recall* atau *specificity*, *F1-score*, dan *macro average*. Berikut ini adalah rumus perhitungan untuk mendapatkan confusion matriks (Mohajon, 2020).

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (2.3)$$

$$Precision = \frac{TP}{TP + FP} \quad (2.4)$$

$$Recall = \frac{TP}{TP + FN} \quad (2.5)$$

$$Specificity = \frac{TN}{TN + FP} \quad (2.6)$$

$$F_1 - score = \frac{2 \times Precision \times Recall}{Precision + Recall} \quad (2.7)$$

TP adalah *True Positive*, TN adalah *True Negative*, FP adalah *False Positive*, dan FN adalah *False Negative*. Sebagai contoh, Tabel 2.1 merupakan contoh confusion matriks untuk klasifikasi *multi-class*. Baris A,B,C merupakan kelas prediksi, sedangkan Kolom A,B,C merupakan kelas *actual*. Nilai TPA, TPB, dan TPC adalah nilai *True Positive* pada kelas A, B, dan C. Sementara *True Negative* kelas A = TPB + EBC+ ECB + TPC. *False Positive* kelas A = EBA + ECA dan *False Negative* kelas A = EAB + EAC. Kemudian, dari nilai precision, recall, dan

F1-score yang di dapatkan, akan dihitung nilai rata-rata dari setiap nilai precision, recall, dan F1-score seluruh kelas sebagai nilai *macro average* nilai precision, recall, dan F1-score.

Tabel 2.1. Confusion matriks untuk klasifikasi multi-class (Alaa, 2020)

Predicted class / True Class	A	B	C
A	TPA	EBA	ECA
B	EAB	TPB	ECB
C	EAC	EBC	TPC