BAB II

TINJAUAN PUSTAKA DAN DASAR TEORI

2.1 Studi Literatur.

Dalam beberapa tahun terakhir, peramalan penjualan produk telah menjadi aspek kritis dalam manajemen persediaan bagi Usaha Kecil dan Menengah (UKM) (Wardah, dkk, 2023). Beberapa penelitian terdahulu telah menginvestigasi berbagai metode peramalan untuk mengantisipasi permintaan produk yang efektif dengan mempertimbangkan tantangan yang unik dihadapi oleh UKM di sektor-sektor industri yang berbeda (Chowdhury, dkk, 2021). Rincian perbandingan antara penelitian-penelitian terdahulu dapat dilihat pada Tabel 2.1.

Tabel 2.1 Penelitian Terdahulu Terkait Peramalan Penjualan Produk Pada UKM

No	Penulis	Metode	Tujuan	Hasil
1.	Isnaini dan Sudiarso (2018)	Kausal, runtun waktu, dan kombinasi kausal-runtun waktu (SARIMA)	Perbandingan metode pada data agregat bulanan	ARIMA lebih unggul dengan nilai rata-rata MAPE 18,2% pada data validasi.
2.	Chatzipanagiot (2018)	SBA *dan Croston*	Perbandingan metode pada data agregat bulanan	SBA mendapat nilai MAD dan MSE lebih rendah dari *Croston*.
3.	Kolade dkk (2019)	ES, MA, LCM	Perbandingan metode pada data agregat bulanan	Model ES lebih unggul dengan nilai MAD 4,84, MSE 4,84, dan MAPE 18,74.
4.	Nivasanon dkk (2019)	MA, SES, DES, RAM	Menemukan metode terbaik masing-masing produk pada data agregat bulanan	Model RAM unggul untuk produk A, C dan L, Model SES untuk produk B, Model MA untuk produk E
5.	Herrera-Granda dkk (2019)	ANN dan ETS *Forecast Pro* v5.0	Perbandingan metode peramalan pada data agregat harian	Model ANN unggul dengan nilai MSE 0,96 dan RMSE 0,98.

Tabel 2.1. Penelitian Terdahulu Terkait Peramalan Penjualan Produk Pada UKM (Lanjutan)

No	Penulis	Metode	Tujuan	Hasil
6.	Oey dkk (2020)	MA, WMA, ES, ETS, MD, AD	Perbandingan metode pada data agregat bulanan	Nilai rata-rata MAD, MSE, dan MAPE dari AD lebih unggul dari model lainnya. 3%.
7.	Javaregowda dkk (2020)	XGBoost	Meningkatkan akurasi peramalan pada data agregat mingguan.	Rata-rata RMSE sebesar 0,68.
8.	Angulo-Baca dkk (2020)	Kombinasi ARIMA dan CPFR	Mengurangi biaya *inventory* manufaktur sepatu	Kombinasi model memungkinkan penurunan biaya *infentory* sebesar 17%.
9	Fahrudin dkk (2022)	LLS dan SVR	Perbandingan metode dan implementasi aplikasi	Metode LLS lebih unggul dengan nilai MSE 1563323871 60,88, RMSE 395388,91 dan MAE 247888,69
10.	Kolková dan Ključnikov (2022)	Naive, SES, RWD, MF, Theta, ARIMA, ETS, NNAR, Prophet, dan Uber	Perbandingan model pada data agregat harian	Model Prophet memiliki akurasi yang lebih tinggi dari model-model lainnya.
11.	Pratama dkk (2022)	Kombinasi MA dan MPS	Peningkatan perencanaan produksi	Penelitian menghasilkan MPS berdasarkan peramalan MA
12.	Purnamasari, dkk (2023)	SARIMA, SARIMAX, LSTM, XGBoost dan GPR	Perbandingan metode peramalan pada data agregat harian	Model XGBoost unggul dari model lainnya dengan nilai RMSE 55,77 dan MAPE 41,18
13.	Wardah dkk (2023)	Kombinasi ARIMA dan EOQ	Mengintegrasikan model peramalan dengan model kontrol inventori	*Inventory control scheduling* berdasarkan peramalan permintaan.

Penelitian-penelitian terdahulu menggunakan agregasi temporal yang berbeda-beda. Agregasi temporal bulanan digunakan untuk melakukan peramalan pada level strategis guna memproyeksikan permintaan satu tahun ke depan. Implikasi dari penentuan agregasi temporal ini menjadikan data yang digunakan dalam peramalan menjadi lebih sedikit, oleh karena itu pendekatan statistika klasik lebih banyak digunakan (Isnaini dan Sudiarso, 2018; Chatzipanagiot, 2018; Oey, dkk, 2020; Kolade, dkk, 2019; Nivasanon, dkk, 2019; Angulo-Baca, dkk, 2020; Pratama, dkk, 2022). Pendekatan agregasi temporal yang lebih detail (agregasi mingguan atau harian), digunakan untuk meramalkan permintaan produk pada *operational level* untuk periode satu hingga tiga minggu ke depan. Pada tingkat agregasi yang lebih detail ini, metode *machine learning* lebih banyak digunakan (Herrera-Granda, dkk, 2019; Javaregowda, dkk, 2020; Purnamasari, dkk, 2023).

Dalam penelitian sebelumnya, terdapat dua jenis pendekatan peramalan yang diterapkan. Pendekatan pertama adalah peramalan untuk satu produk tunggal. Pendekatan ini umumnya digunakan oleh UKM yang memiliki fokus penjualan hanya pada satu produk, seperti contoh pada penelitian Herrera-Granda dkk (2019) yang meramalkan permintaan air minum, dan pada studi Kolková dan Ključnikov (2022) yang meramalkan permintaan sewa sepeda. Pendekatan yang kedua adalah peramalan *multiple* produk. Peramalan *multiple product* lebih sulit daripada peramalan *single product*. Hal ini disebabkan oleh kompleksnya pola permintaan setiap produk. Untuk mengatasi tantangan ini, beberapa pendekatan telah diusulkan oleh penelitian sebelumnya. Isnaini dan Sudiarso (2018) melakukan agregasi data dari 30 produk berdasarkan atribut bersama, yaitu kategori produk, menjadi lima runtun waktu, kemudian dilakukan peramalan berdasarkan kategori tersebut. Purnamasari dkk (2023) menggunakan teknik *clustering K-Means* untuk mengelompokkan ribuan produk menjadi tiga kelompok sebelum melakukan peramalan. Kolade dkk (2019), Fahrudin dkk (2022), Pratama dkk (2022), dan Angulo-Baca dkk (2020) melakukan agregasi total penjualan produk sebelum melakukan peramalan. Pendekatan agregasi ini membantu menangani pola permintaan yang intermittent pada produk, membuat data runtun waktu menjadi lebih halus dan mudah diramalkan. Kelemahan dari pendekatan ini adalah hasil

peramalan menjadi tidak mendetail per produk. Oleh karena itu informasi mengenai produk apa yang terjual pada masa depan, tidak dapat terjawab oleh pendekatan ini.

Chatzipanagiot (2018) mengadopsi pendekatan serupa untuk mengatasi pola permintaan yang *intermittent* pada produk. Pertama, data setiap produk digabungkan berdasarkan *purchase likelihood*, kemudian dilakukan analisis Pareto untuk memilih kelompok produk yang signifikan, dan dilakukan peramalan pada setiap kelompok yang dipilih. Hasil peramalan kemudian di disagregasi menggunakan metode *top-down*. Pendekatan ini lebih unggul dari pendekatan-pendekatan sebelumnya karena menghasilkan peramalan yang lebih detail. Adapun kelemahan dari pendekatan ini yaitu, karena peramalan dilakukan pada kelompok produk, model yang digunakan mungkin tidak mampu menangkap pola keseluruhan dari data runtun waktu karena informasi yang hilang akibat agregasi. Terdapat beberapa pendekatan lainnya yang mampu menghasilkan peramalan dengan detail yang lebih baik. Nivasanon dkk (2019) meramalkan lima produk menggunakan empat model peramalan yang berbeda, model-model tersebut digunakan untuk meramalkan setiap produk, hasil peramalan kemudian bandingkan untuk mengetahui model yang sesuai pada masing-masing produk. Kelemahan pendekatan ini adalah jika jumlah produk yang diramalkan sangat banyak, maka kombinasi model yang dibutuhkan sangat banyak, oleh karena itu pendekatan ini tidak *scalable*. Javaregowda dkk (2020) mengatasi masalah ini menggunakan model XGBoost. Model ini digunakan untuk meramalkan banyak produk sekaligus, dan menghasilkan peramalan yang akurat dengan nilai rata-rata RMSE sebesar 0,68. Kelemahan dari penelitian ini adalah karena proses validasi dan penentuan *hyperparameter* menggunakan *K-fold* validasi, maka ada kemungkinan *information leaks* dari masa depan selama proses pelatihan dan validasi model.

Perbedaan penelitian ini dengan penelitian-penelitian sebelumnya adalah penelitian ini menggunakan pendekatan *hierarchical time series* yang direkonsiliasi menggunakan teknik optimasi rekonsiliasi non-negative MinT dengan kombinasi model *deep learning* N-BEATS dalam tugas peramalan *multiple product*. Pendekatan hierarkial rekonsiliasi *non negative* dengan kombinasi model *deep learning* N-BEATS bertujuan agar model yang dikembangkan dapat mengenal pola

data secara holistik antar runtun waktu dan antar hierarki sehingga dapat menghasilkan peramalan yang akurat untuk semua tingkatan hierarki produk sambil menjaga hasil peramalan tetap positif. Keberhasilan N-BEATS dalam penelitian sebelumnya, sebagaimana diungkapkan oleh Oreshkin dkk (2018) serta penelitian lainnya, memberikan landasan teoritis yang kuat untuk menerapkannya dalam konteks UKM retail. Untuk mendemonstrasikan efektivitas model yang diusulkan, dilakukan perbandingan dengan *baseline model* yang biasa digunakan dalam tugas peramalan penjualan produk seperti *Exponential Smoothing* dan ARIMA menggunakan teknik validasi pemisahan runtun waktu (*time series split*) untuk mencegah *information leaks* dari masa depan selama proses pelatihan dan validasi model.

2.2 Dasar Teori.

2.2.1 Peramalan Runtun Waktu.

Runtun waktu merupakan serangkaian data yang diukur melalui periode waktu tertentu (Dama dan Sinoquet, 2021). Secara umum, analisis runtun waktu dapat dibagi menjadi dua kelompok. Kelompok pertama bertujuan untuk mengumpulkan dan memeriksa pengamatan masa lalu dari suatu runtun waktu untuk mengembangkan model yang tepat dalam menjelaskan struktur inheren runtun waktu, seperti tren dan musiman (Dama dan Sinoquet, 2021). Sedangkan kelompok kedua bertujuan untuk membuat model, yang dapat melakukan prediksi di masa depan (Deb, dkk, 2017).

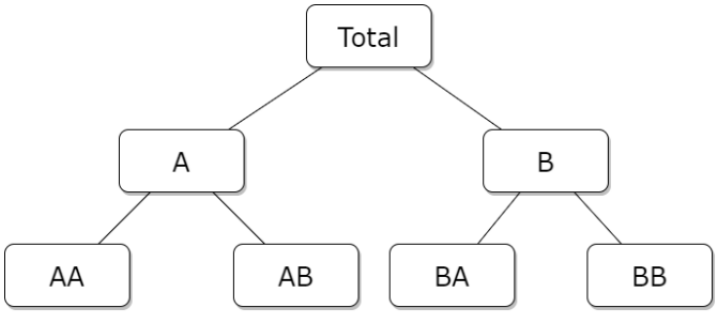
Runtun waktu ditandai oleh properti inheren yang berubah seiring waktu. Runtun waktu dapat menunjukkan adanya tren, variasi musiman atau siklik, dan dapat mengalami pergeseran level (Klein, dkk, 2021). Tren merujuk pada kecenderungan umum atau arah pergerakan runtun waktu yang diamati selama periode tertentu. Variasi musiman atau siklik adalah pola fluktuasi atau variasi periodik yang berulang secara teratur pada suatu runtun waktu selama periode waktu tertentu. Aspek khas lain dari runtun waktu adalah *autocorrelation*. *Autocorrelation* adalah hubungan korelasi antara dua pengamatan dari variabel yang sama pada waktu yang berbeda dalam rangkaian waktu yang sama (Wu dkk,

2021). *Autocorrelation* menunjukkan bahwa nilai masa lalu dapat mempengaruhi nilai saat ini (runtun waktu yang berkorelasi dengan dirinya sendiri). Properti penting lainnya dari runtun waktu adalah *stationarity.* Sebuah runtun waktu dianggap stasioner jika propertinya tidak dipengaruhi oleh waktu (memiliki rata-rata dan standar deviasi yang konstan sepanjang waktu) (Changqing Cheng dan Bukkapatnam, 2015).

Pendekatan dalam peramalan runtun waktu dapat bersifat *univariat* atau *multivariat*. Pendekatan *univariat* hanya mempertimbangkan satu rangkaian runtun waktu untuk meramalkan variabel target di masa depan, sementara pendekatan *multivariat* mempertimbangkan beberapa rangkaian waktu termasuk rangkaian waktu target untuk meramalkan variabel target di masa depan (Castán-Lascorz, dkk, 2022). Selain itu, peramalan runtun waktu juga dapat dibagi menjadi *single-step* atau *multi-step*. Pendekatan peramalan *single-step* hanya menghasilkan satu prediksi untuk selang waktu yang sama dengan tingkat agregasi runtun waktu tersebut, sedangkan pendekatan peramalan *multi-step* menghasilkan beberapa langkah prediksi ke depan (Lijuan dan Guohua, 2016). Pendekatan peramalan *multi-step* dapat bersifat rekursif atau langsung (Sahoo, dkk, 2020). Dalam pendekatan rekursif, prediksi $\hat{y}_{t+1}$ digunakan untuk menghitung $\hat{y}_{t+2}$ dan prediksi $\hat{y}_{t+1}$ dan $\hat{y}_{t+2}$ digunakan untuk menghitung $\hat{y}_{t+3}$, dan seterusnya. Kelemahan teknik ini adalah adanya potensi penyebaran kesalahan prediksi ke langkah-langkah berikutnya. Sebaliknya, pendekatan langsung melibatkan pengembangan model yang dapat menghasilkan nilai secara langsung untuk beberapa langkah waktu ke depan. Namun, pendekatan ini sering memerlukan lebih banyak sumber daya komputasi dibandingkan dengan teknik rekursif (Liu, dkk, 2017).

2.2.2 Hierarchical Time Series.

Hierarchical time series adalah tipe khusus runtun waktu *multivariat* yang masing-masing entitas runtun waktunya terikat satu sama lain melalui struktur hierarkis tertentu (Wickramasuriya, dkk, 2019). Ilustrasi dari struktur tersebut terdapat pada Gambar 2.1.



Gambar 2.1 Ilustrasi struktur runtun waktu hierarkis

Puncak hierarki dari runtun waktu di atas, merepresentasikan jumlah total dari seluruh runtun waktu yang terdapat dalam struktur hierarkis. Struktur di bawah puncak hierarki (satu tingkat di bawah puncak) merupakan total dari runtun waktu yang terbagi berdasarkan suatu parameter tertentu, seperti lokasi geografis, kategori produk, atau ID toko. Struktur ini kemudian dibagi lagi (menjadi tingkat yang lebih terperinci) dengan metode yang serupa, hingga mencapai runtun waktu tingkat terendah. Struktur hierarki pada Gambar 2.1 memiliki tiga tingkat; Akan tetapi pada umumnya rangkaian waktu hierarkis dapat memiliki tingkat yang bervariasi. Sebuah runtun waktu hierarkis harus mematuhi aturan bahwa jumlah simpul waktu pada setiap tingkat hierarkis harus sebanding dengan simpul pada tingkat tertinggi dalam struktur hierarkis.

Rangkaian waktu hierarkis dapat ditemukan pada berbagai domain data, seperti data pengunjung pariwisata pada sebuah negara (Hyndman, dkk, 2011); data konsumsi daya listrik (Taieb, dkk, 2021); dan data penjualan pada industri retail (Wickramasuriya, dkk, 2019). Sebagai contoh, pada Gambar 2.1, runtun waktu tingkat tertinggi dapat merepresentasikan total penjualan suatu organisasi retail, runtun waktu tingkat menengah A dan B dapat mewakili total penjualan pada setiap kategori SKU yang dijual, dan runtun waktu tingkat terendah AA, AB, BA, dan BB pada Gambar 2.1 dapat mencerminkan total penjualan dari masing-masing SKU pada setiap kategori yang dijual.

Suatu himpunan runtun waktu N dengan panjang T memiliki struktur hierarkis, dimana $y_t \in R^N$ adalah vektor yang berisi keseluruhan observasi dari setiap runtun waktu individu pada waktu t, dan $b_t \in R^M$ adalah vektor yang berisi

pengamatan dari M runtun waktu tingkat bawah (level paling bawah struktur hierarkis) pada waktu t (Wickramasuriya, dkk, 2019). Maka struktur hierarkis dari runtun waktu tersebut dapat direpresentasikan dalam bentuk matriks

$$y_t = Sb_t \,, \tag{2.1}$$

Keterangan:

y_t = Vektor yang berisi keseluruhan observasi dari setiap runtun waktu individu dalam hierarki pada waktu t

S = merupakan matriks konstanta $S \in R^{NxM}$ yang menggambarkan struktur runtun waktu dari tingkat paling bawah ke tingkat paling atas hierarki.

$\hat{y}_{t+h}$ = vektor yang berisi pengamatan dari M runtun waktu tingkat bawah (level paling bawah struktur hierarkis) pada waktu t

2.2.3 Metode Rekonsiliasi Dalam Runtun Waktu.

Sebuah rangkaian runtun waktu yang tersusun dalam struktur hierarki tertentu harus memenuhi kondisi bahwa jumlah simpul runtun waktu pada setiap tingkat hierarki harus sebanding dengan simpul runtun waktu pada tingkat tertinggi dalam struktur hierarkis. Hal yang sama harus berlaku untuk proyeksi atau peramalan dari runtun waktu tersebut. Jika proyeksi atau peramalan yang dibuat mematuhi aturan atau batasan ini, maka proyeksi atau ramalan tersebut disebut koheren (Wickramasuriya, dkk, 2019). Pada umumnya proyeksi atau ramalan yang dihasilkan oleh model dasar (*base model*) dari runtun waktu cenderung tidak koheren. Oleh karena itu diperlukan metode tambahan untuk memastikan koherensi dari hasil peramalan runtun waktu ini. Metode-metode tersebut umumnya dikenal sebagai metode rekonsiliasi.

Metode rekonsiliasi yang paling umum digunakan adalah linear rekonsiliasi (Wickramasuriya, dkk, 2019). Jika $\hat{y}_{t+h}$ didefinisikan sebagai peramalan dasar *multi-horizon* selama h-langkah dari seluruh runtun waktu dalam hierarki, dan S sebagai matriks penjumlahan yang menggambarkan struktur hierarki, maka setiap rekonsiliasi linear dapat dirumuskan sebagai

$$\tilde{y}_{t+h} = SP\hat{y}_{t+h}, \tag{2.2}$$

Keterangan:

$\tilde{y}_{t+h}$ = Hasil rekonsiliasi peramalan *multi-horizon* pada h-langkah ke depan dari seluruh runtun waktu dalam hierarki.

S = Matriks kovarian yang menggambarkan struktur runtun waktu.

$\hat{y}_{t+h}$ = Hasil peramalan dasar *multi-horizon* selama h-langkah ke depan dari seluruh runtun waktu dalam hierarki.

P = Matriks pemetaan optimal (*mapping matrix*) yang menghubungkan ramalan dasar ($\hat{y}_{t+h}$) dengan ramalan yang direkonsiliasi ($\tilde{y}_{t+h}$).

2.2.3.1 *Top-Down* dan *Bottom-Up* Rekonsiliasi.

Metode rekonsiliasi yang paling sederhana adalah metode *Top-Down* dan *Bottom-Up* (Wickramasuriya, dkk, 2019). Pendekatan *Bottom-Up* dilakukan dengan membuat peramalan dasar untuk semua runtun waktu di tingkat terendah, kemudian mengagregasikan ramalan dasar tersebut ke tingkat yang lebih tinggi. Keuntungan utama dari pendekatan ini adalah, karena peramalan dilakukan pada tingkat terendah, tidak ada informasi yang hilang akibat agregasi. Kelemahan dari metode ini adalah, bahwa runtun waktu pada tingkat terendah seringkali fluktuatif dan menunjukkan pola observasi yang tidak teratur (*intermittent*), sehingga lebih sulit untuk diramalkan dan biasanya kurang baik ketika data di agregasi pada level yang lebih tinggi. Pendekatan *Top-Down* dilakukan dengan peramalan pada runtun waktu tingkat hierarki paling atas dan membaginya pada level hierarki terbawah. Adapun kendala dari pendekatan *top-down* yang menggunakan proporsi historis cenderung menghasilkan peramalan yang kurang akurat pada tingkat hierarki yang lebih rendah.

2.2.3.2 Minimum Trace (MinT) Reconciliation.

Kedua metode rekonsiliasi *Top-Down* dan *Bottom-Up* mengalami kehilangan informasi karena peramalan untuk seluruh hierarki hanya didasarkan pada peramalan pada tingkat teratas atau tingkat terendah. Untuk memanfaatkan informasi dari semua tingkat dalam hierarki, Wickramasuriya dkk (2019)

mengusulkan pendekatan dengan meminimalkan jejak matriks kovarian P dari kesalahan ramalan rekonsiliasi.

Kesalahan ramalan dasar pada h langkah ke depan didefinisikan sebagai

$$\hat{e}(h) = y_{t+h} - \hat{y}_{t+h}, \tag{2.3}$$

Keterangan:

$\hat{e}(h)$ = Kesalahan ramalan dasar pada h langkah ke depan.

y_{t+h} = Nilai aktual pada waktu $t + h$.

$\hat{y}_{t+h}$ = Ramalan dasar untuk waktu $t + h$.

Jika peramalan dasar tidak bias maka ekspektasi dari kesalahan peramalan adalah nol, yaitu $E[\hat{e}(h)] = 0$, hal ini berarti ekspektasi nilai aktual harus sama dengan ekspektasi peramalan dasar yaitu $E[y_{t+h}] = E[\hat{y}_{t+h}]$. Selanjutnya jika $\hat{b}_{t+h}$ merupakan hasil peramalan dasar pada tingkat hierarki paling bawah maka nilai ekspektasi $\hat{b}_{t+h}$ adalah $E[\hat{b}_{t+h}] = \beta_{t+h}$, sehingga ekspektasi dari peramalan dasar dapat dinyatakan sebagai $E[\hat{y}_{t+h}] = S\beta_{t+h}$. Melalui asumsi tidak bias ini, maka ekspektasi rekonsiliasi harus sama dengan ekspektasi nilai aktual yaitu

$$\begin{aligned} E[\tilde{y}_{t+h}] &= E[y_{t+h}] \\ E[SP\hat{y}_{t+h}] &= S\beta_{t+h} \\ SPS\beta_{t+h} &= S\beta_{t+h} \\ SPS &= S \\ S'SPS &= S'S \\ PS &= I_N, \end{aligned} \tag{2.4}$$

dimana $I_N \in R^{N \times N}$ adalah matriks identitas. Selanjutnya kesalahan rekonsiliasi peramalan pada h langkah ke depan didefinisikan sebagai

$$\check{e}_t(h) = y_{t+h} - \tilde{y}_{t+h}, \tag{2.5}$$

Keterangan:

$\check{e}_t(h)$ = Kesalahan rekonsiliasi peramalan pada h langkah ke depan.

y_{t+h} = Nilai aktual pada waktu $t + h$.

$\tilde{y}_{t+h}$ = Ramalan yang telah direkonsiliasi untuk periode waktu $t + h$.

Wickramasuriya dkk (2019) membuktikan bahwa untuk setiap matriks P yang memenuhi persamaan $P\,S = I_N$, maka variasi matrik kovarian dari setiap kesalahan rekonsiliasi pada h langkah ke depan dapat didefinisikan sebagai

$$Var[\check{e}_t(h)] = SPW_hP'S' , \tag{2.6}$$

Keterangan:

$Var[\check{e}_t(h)]$ = Matriks kovarian dari kesalahan rekonsiliasi peramalan pada h langkah ke depan

S = Matriks kovarian yang menggambarkan struktur runtun waktu dari tingkat paling bawah ke tingkat paling atas hierarki.

W_h = Matriks kovarian dari kesalahan peramalan dasar pada h langkah ke depan.

P = Matriks pemetaan optimal (*mapping matrix*) yang menghubungkan ramalan dasar ($\hat{y}_{t+h}$) dengan ramalan yang direkonsiliasi ($\tilde{y}_{t+h}$).

Tujuan utama dari pendekatan rekonsiliasi *Minimum Trace* adalah untuk menemukan matriks kovarian P sedemikian rupa sehingga memenuhi persamaan (2.7) yaitu $P\,S = I_N$, karena meminimalkan jejak matriks kovarian setara dengan meminimalkan jumlah varians dari kesalahan ramalan yang direkonsiliasi. Solusi dari masalah ini dijabarkan dalam persamaan

$$P = (S'W_h{}^{-1}S)^{-1}S'W_h{}^{-1} \tag{2.7}$$

Keterangan:

P = Matriks pemetaan optimal (*mapping matrix*) yang menghubungkan ramalan dasar ($\hat{y}_{t+h}$) dengan ramalan yang direkonsiliasi ($\tilde{y}_{t+h}$).

S = Matriks kovarian yang menggambarkan struktur runtun waktu dari tingkat paling bawah ke tingkat paling atas hierarki.

W_h = Matriks kovarian dari kesalahan peramalan dasar pada h langkah ke depan.

Karena eror peramalan dasar tidak dapat dihitung sebelumnya, maka nilai W_h harus diestimasi. Berikut beberapa metode untuk mengestimasi matrix W_h.

OLS: Pada metode ini, nilai $W_h = I_N$. Metode ini sama dengan model *Ordinary Least Square estimator* pada tulisan Hyndman dkk (2011), sehingga nilai dari matrix

$$P = (S' I_N^{-1} S)^{-1} S' I_N^{-1} = S' S^{-1} S' \quad (2.8)$$

WLS$_s$: Pada metode ini, nilai dari $W_h = \Lambda$, dimana Λ merupakan matrix diagonal dengan nilai $S1$, dimana S merupakan matriks penjumlahan dan $1 \in R^{M \times 1}$ merupakan unit vektor. Metode ini membuat bobot untuk deret waktu tingkat atas akan menjadi *M* (jumlah total deret waktu tingkat bawah ke level atas) dan 1 untuk bobot deret waktu tingkat bawah. Oleh karena itu, metode ini disebut WLS$_s$ (*weighted least squares applying structural scaling*).

Salah satu masalah utama dari pendekatan rekonsiliasi MinT adalah bahwa metode ini tidak menjamin ramalan yang disesuaikan bersifat *non-negatif*. Kelemahan ini menjadi masalah serius dalam konteks peramalan penjualan yang secara inheren variabel targetnya bersifat *non-negatif*. Untuk mengatasi masalah ini rekonsiliasi dirumuskan melalui pemrograman kuadrat, dengan memberlakukan batasan non-negatif sehingga hasil akhir peramalan yang direkonsiliasi memiliki nilai *non-negatif*.

2.2.4 Model Peramalan.

2.2.4.1 *Benchmarks* Model.

Model *Benchmarks* dalam penelitian ini digunakan sebagai perbandingan untuk menilai efektivitas dari model *deep learning* yang diusulkan (N-BEATS). Model *benchmarks* yang digunakan dalam penelitian ini adalah *Exponential smoothing* (ES), dan *Autoregressive Integrated Moving Average* (ARIMA).

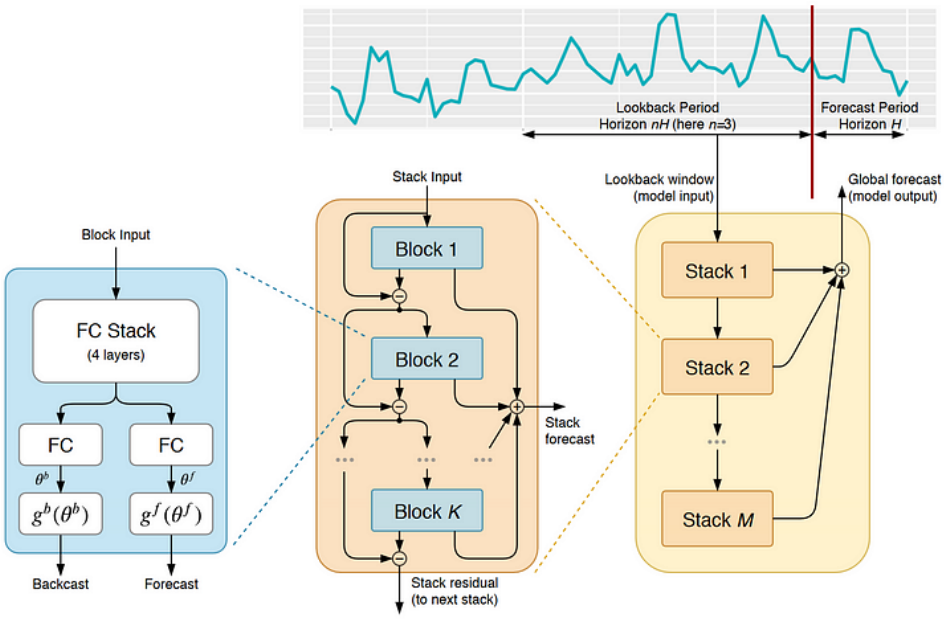
Pemodelan *Exponential smoothing* membutuhkan pemisahan komponen utama runtun waktu (*time series decomposition*) menjadi tiga bagian yaitu komponen tren, komponen musiman, dan komponen kesalahan (*error component*) (Hyndman, dkk, 2002). Berbagai kombinasi dari komponen-komponen ini menghasilkan kombinasi model ES yang berbeda. Dalam kerangka kerja yang dijelaskan oleh Hyndman, dkk, (2002) diajukan pendekatan pemilihan model otomatis (*automatic model selection approach*) untuk memilih model *Exponential smoothing* (ES) yang paling sesuai dengan data runtun waktu yang diamati. Proses ini dilakukan melalui dua tahap utama, yaitu memaksimalkan *likelihood* (kemungkinan) untuk setiap model ES dan memilih model yang menghasilkan *Akaike's Information Criterion* (AIC) terendah.

Pada kompetisi *M5 forecasting,* kerangka kerja pemilihan model (*model selection*) *Exponential smoothing* (ES) yang digabungkan pendekatan *Bottom-Up reconciliation* (ES_bu) digunakan sebagai model dasar yang dijadikan *Benchmarks* (Makridakis, dkk, 2022). Model ini berhasil mengalahkan lima puluh model *benchmark* lainnya dan hanya 415 dari 5507 tim (7,5%) yang berhasil mengalahkan model *benchmark* ini. Oleh karena itu, metode pendekatan ES_bu akan berfungsi sebagai tolak ukur (*benchmark*) yang kompetitif dalam penelitian ini.

Model ARIMA (*Autoregressive Integrated Moving Average*) banyak digunakan dalam peramalan deret waktu karena berbasis regresi linear. Model ini sangat efektif untuk data yang memiliki autokorelasi, karena peramalan yang dihasilkan ARIMA didasarkan pada nilai historis masa lalu dari data itu sendiri. Model ARIMA terdiri dari tiga komponen utama: *autoregressive* (AR), *integration* (I), dan *moving average* (MA). Komponen AR memprediksi nilai masa depan berdasarkan observasi masa lalu, komponen MA memprediksi nilai masa depan menggunakan residu atau kesalahan dari observasi sebelumnya, dan komponen I digunakan untuk membuat data menjadi stasioner dengan menghilangkan tren atau musiman. Pemilihan ketiga parameter dari model ARIMA dilakukan menggunakan *Akaike Information Criterion* (AIC). Kombinasi dari ketiga komponen ini memungkinkan ARIMA fleksibel pada berbagai jenis data runtun waktu.

2.2.4.2 Model N-BEATS.

N-BEATS (*Neural Basis Expansion Analysis for interpretable Time Series forecasting*) merupakan arsitektur *deep learning* yang dirancang khusus untuk meramalkan runtun waktu (Oreshkin, dkk, 2019). N-BEATS dapat secara efektif meramalkan berbagai jenis data runtun waktu tanpa memerlukan pendekatan *feature engineering* yang kompleks dan spesifik untuk setiap runtun waktu (Shaikh, dkk, 2022). Arsitektur dari model N-BEATS dapat dilihat pada Gambar 2.2.



Gambar 2.2 *Architecture* N-BEATS (Oreshkin, dkk, 2019)

Arsitektur N-BEATS membagi data runtun waktu menjadi dua bagian (Gambar 2.2 bagian kanan atas). Bagian pertama adalah periode pengamatan mundur (*lookback period*) dengan panjang nH dan bagian kedua merupakan periode peramalan (*forecast period*) dengan panjang H. *Lookback period* ini akan digunakan dalam model untuk melakukan peramalan, sedangkan *forecast period* (berisi nilai sebenarnya) digunakan untuk melakukan evaluasi peramalan oleh model. Model N-BEATS terdiri dari tumpukan lapisan atau *stacks* (kotak kuning di sebelah kanan). Masing-masing lapisan ini dibangun dari kumpulan blok-blok (kotak oranye di tengah), dan konstruksi dari setiap blok dapat dilihat pada kotak biru di sebelah kiri.

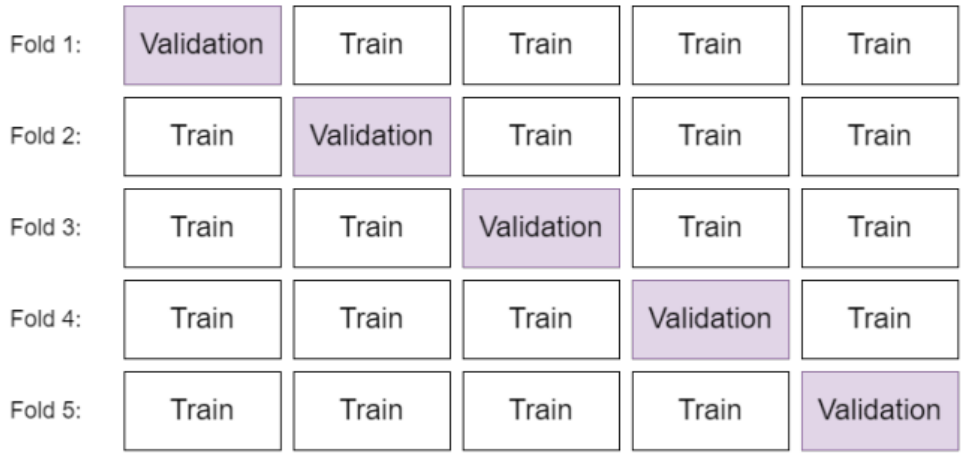
Setiap blok terdiri dari empat lapisan yang sepenuhnya saling terhubung. Jaringan yang saling terhubung ini menghasilkan dua keluaran yaitu peramalan (*forecast*) dan *backcast*. *Forecast* merupakan prediksi nilai masa depan, dan backcast merupakan residu dari hasil peramalan yang akan diteruskan sebagai masukan pada blok berikutnya. Dalam arsitektur ini, hanya blok pertama yang mendapatkan rangkaian input aktual, blok berikutnya kemudian mendapatkan residu yang berasal dari blok pertama. Oleh karena itu hanya informasi yang tidak tertangkap oleh blok pertama yang diteruskan ke blok berikutnya (setiap blok berusaha menangkap informasi yang mungkin terlewatkan oleh blok sebelumnya). Kombinasi dari setiap prediksi pada masing-masing blok ini kemudian membentuk ramalan akhir.

2.2.5 Pemilihan dan Evaluasi Model.

Terdapat dua tugas utama dalam *supervised learning*. Tugas yang pertama adalah melakukan pencarian *hyperparameter* terbaik untuk jenis model tertentu dan tugas yang kedua adalah melakukan perbandingan terhadap model-model yang berbeda untuk mencari model terbaik (Raschka, dkk, 2020).

Pendekatan yang paling sederhana dalam melakukan pemilihan dan evaluasi model adalah dengan membagi data yang tersedia menjadi tiga subset data yang berbeda, yaitu *training set, validation set,* dan *test set*. *Training set* digunakan untuk melakukan pelatihan pada model, *validation set* digunakan untuk menemukan *hyperparameter* terbaik, dan *error* (kesalahan prediksi) pada *test set* digunakan untuk melakukan validasi terhadap kinerja model. Pendekatan ini dikenal sebagai *three-way holdout method* (Raschka, dkk, 2020). *Three-way holdout method* memiliki kelemahan ketika digunakan pada dataset yang kecil (Raschka, dkk, 2020). Karena hanya *kesalahan* pada *validation* set yang digunakan untuk menemukan *hyperparameter* terbaik, maka ada kemungkinan model yang dilatih hanya optimal pada data validasi saja, jika pembagian set validasi tidak menggambar distribusi data secara keseluruhan dengan baik. Masalah ini dapat diatasi menggunakan metode validasi silang *k-fold*. Contoh validasi silang silang *k-fold* dapat dilihat pada Gambar 2.3.

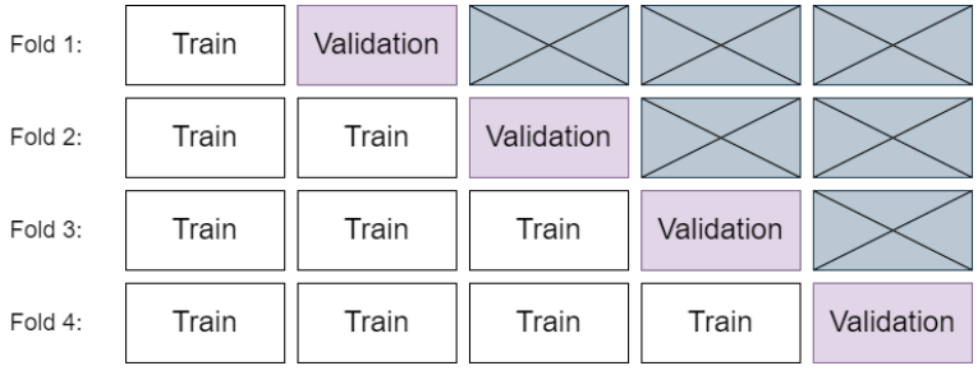
Metode validasi silang *k-fold* dimulai dengan melakukan pembagian data menjadi $k-$ subset terpisah. Pada Gambar 2.3, set data dibagi menjadi 5 *fold* artinya set data dibagi menjadi 5 bagian yang sama besar, sehingga model dapat dilatih dan diuji sebanyak 5 iterasi. Pada setiap iterasi, satu *fold* dijadikan sebagai set validasi (*validation set*), sementara empat *fold* lainnya digunakan sebagai set pelatihan (*train set*). Pada Gambar 2.3, *train set* merujuk pada sub set data yang digunakan untuk melatih model. Data ini digunakan untuk mengajarkan model bagaimana mengenali pola dan hubungan dalam data, serta mengoptimalkan parameter model untuk meminimalkan kesalahan prediksi. Sebaliknya, *validation set* adalah sub set data yang digunakan untuk mengevaluasi kinerja model setelah proses pelatihan. Data validasi memberikan indikasi mengenai seberapa baik model dapat melakukan generalisasi pada data yang belum pernah dilihat sebelumnya.



Gambar 2.3 Ilustrasi dari validasi silang 5-fold

Metode validasi silang k-*fold* bergantung pada asumsi bahwa semua titik data independen sehingga pembagian data pada setiap *fold* dilakukan tanpa mempertimbangkan hubungan temporal. Namun, dalam peramalan runtun waktu, semua titik data memiliki hubungan temporal. Penggunaan validasi silang k-*fold* dapat menyebabkan pelanggaran struktur temporal data dimana data dari masa depan dapat digunakan dalam pelatihan model sebelum model diuji pada data tersebut. Oleh karena itu, validasi silang k-*fold* bukan pilihan yang layak untuk peramalan runtun waktu karena memungkinkan informasi pada masa depan

diketahui oleh model saat ini (Lainder dan Wolfinger, 2022). Pendekatan paling sederhana untuk mengatasi masalah ini adalah menggunakan metode pemisahan runtun waktu (*time series split*) (Lainder dan Wolfinger, 2022). Ilustrasi pendekatan *time series split* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Ilustrasi dari *time series split*

Metode validasi pemisahan runtun waktu (*time series split*) membagi data secara berurutan untuk melatih dan menguji model. Berbeda dengan k-*fold validation* yang membagi data secara acak atau merata. Metode *time series split* mempertahankan urutan waktu dan memastikan bahwa data yang lebih awal digunakan untuk melatih model, sementara data yang lebih baru digunakan untuk menguji model. Sebagai contoh, semisal sebuah organisasi retail memiliki data penjualan produk harian selama satu tahun kemudian dilakukan 5-*Fold Time Series Split* dengan horizon peramalan 14 hari, data dibagi menjadi lima *fold* berurutan. masing-masing *fold* terdiri dari 73 hari. Pada setiap iterasi, model dilatih pada data dari *fold* sebelumnya dan diuji pada data dari *fold* berikutnya. Dengan mempertimbangkan horizon peramalan 14 hari, maka dalam Iterasi 1, model dilatih pada data dari hari 1 hingga 59 dan diuji pada hari 60 hingga 73. Pada Iterasi 2, model dilatih pada data dari hari 1 hingga 130 dan diuji pada hari 131 hingga 146. Proses ini diulang hingga Iterasi 5.

Metode validasi pemisahan runtun waktu (*time series split*) memiliki kelemahan. Kelemahan ini muncul karena setiap lipatan (*fold*) dalam metode ini berisi jumlah data yang berbeda, proses optimasi *hyperparameter* untuk satu lipatan mungkin tidak menghasilkan hasil yang optimal pada lipatan lainnya. Selain itu titik

data awal, yang merupakan titik data paling tidak penting digunakan beberapa kali dalam proses validasi silang. Sebaliknya, data paling baru hanya digunakan sekali. Hal ini dapat mengakibatkan bobot yang tidak seimbang pada titik data tertentu dalam pengukuran kinerja model. Meskipun demikian, metode ini merupakan metode validasi yang lebih kokoh dibandingkan dengan metode *three-way holdout.*

2.2.6 Metrik Akurasi.

Dalam mengevaluasi kinerja dari berbagai model peramalan, terdapat beberapa metrik yang digunakan. Metrik-metrik ini meliputi *Mean Absolute Error* (MAE), *Root Mean Squared Error* (RMSE), dan *Mean Absolute Percentage Error* (MAPE). MAE merepresentasikan rata-rata absolut antara nilai sebenarnya dengan nilai peramalan, semakin mendekati nol nilai MAE, semakin baik hasil peramalan. Formula dari MAE adalah

$$MAE = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|, \tag{2.9}$$

Keterangan:

n = jumlah total pengamatan,

y_t = nilai sebenarnya pada waktu t,

$\hat{y}_t$ = nilai yang diprediksi pada waktu t,

sedangkan RMSE merepresentasikan akar dari rata-rata kuadrat perbedaan absolut antara nilai sebenarnya dengan nilai peramalan, semakin mendekati nol nilai RMSE, semakin baik hasil peramalan. Formulasi dari RMSE adalah

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(|y_t - \hat{y}_t|)^2}, \tag{2.10}$$

Keterangan:

n = jumlah total pengamatan,

y_t = nilai sebenarnya pada waktu t,

$\hat{y}_t$ = nilai yang diprediksi pada waktu t,

selanjutnya formulasi dari MAPE didefinisikan sebagai

$$MAPE = \frac{100\%}{n}\sum_{t=1}^{n}\left|\frac{y_t - \hat{y}_t}{y_t}\right|, \tag{2.11}$$

Keterangan:

n = jumlah total pengamatan,

y_t = nilai sebenarnya pada waktu t,

$\hat{y}_t$ = nilai yang diprediksi pada waktu t,

nilai MAPE memberikan ukuran persentase dari kesalahan prediksi terhadap nilai sebenarnya, semakin kecil nilai MAPE, semakin baik hasil peramalan. Masalah utama pada penggunaan MAPE adalah ketika (y_t) memiliki nilai nol, pembaginya menjadi nol, dan hasil pembagian menjadi tidak dapat didefinisikan (*undefined*). *Symmetric Mean Absolute Percentage Error* (SMAPE) mengatasi masalah ini dengan cara menghitung kesalahan prediksi relatif terhadap rata-rata dari nilai aktual dan nilai prediksi, bukan hanya nilai aktual saja. Semakin kecil nilai SMAPE, semakin bagus hasil peramalan.

$$SMAPE = \frac{100\%}{n} \sum_{t=1}^{n} \frac{|y_t - \hat{y}_t|}{|y_t| + |\hat{y}_t|} \tag{2.12}$$

Keterangan:

n = jumlah total pengamatan,

y_t = nilai sebenarnya pada waktu t,

$\hat{y}_t$ = nilai yang diprediksi pada waktu t,

Interpretasi dari MAPE dan SMAPE dapat dilihat pada tabel 2.2.

Tabel 2.2 Interpretasi MAPE dan SMAPE (Maiseli, 2019).

Nilai MAPE	Nilai SMAPE	Evaluasi Kinerja Prediktif
<10%	<10%	Peramalan sangat akurat
10-20%	10-20%	Peramalan bagus
20-50%	20-50%	Peramalan wajar
>50%	>50%	Peramalan tidak akurat

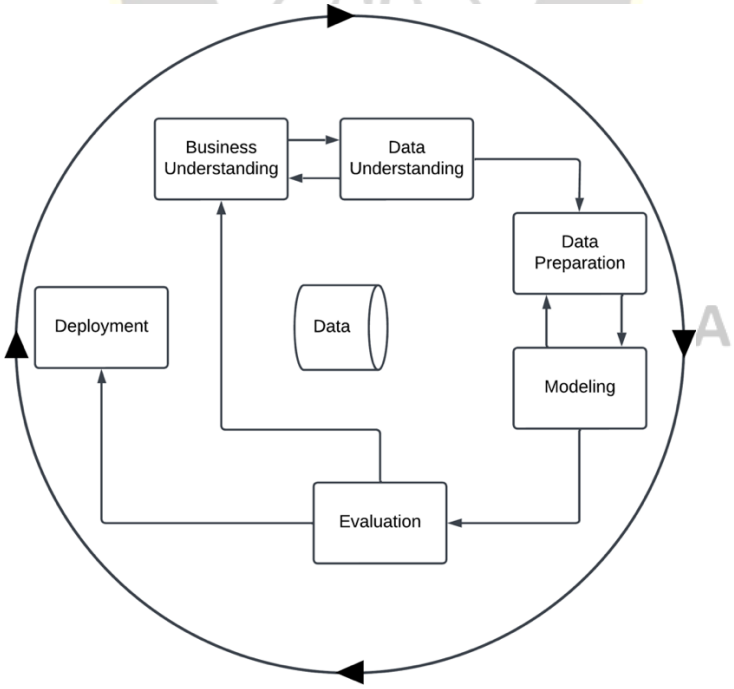
2.2.7 Manajemen Persediaan.

Manajemen persediaan merupakan proses pengelolaan barang atau produk yang tersedia dalam stok perusahaan (Ganesha, dkk, 2020). Proses ini mencakup pengelolaan aliran barang yang masuk dan keluar dari persediaan perusahaan agar

tetap seimbang dan optimal (Muchaendepi, dkk, 2019). Salah satu aspek kunci dalam manajemen persediaan adalah analisis *trend*. Melalui analisis *trend*, perusahaan dapat memperhitungkan pola dan perubahan permintaan pasar, sehingga dapat lebih akurat meramalkan kebutuhan persediaan di masa depan. Hal ini memungkinkan perusahaan memastikan bahwa barang atau produk yang dijual mencukupi untuk memenuhi pesanan pelanggan, sehingga meningkatkan kepuasan pelanggan dan mengoptimalkan kinerja bisnis serta mengurangi biaya yang berlebihan akibat kelebihan stok (Muchaendepi, dkk, 2019).

2.2.8 Metodologi CRISP-DM.

Konsep-konsep yang telah diuraikan sebelumnya kemudian diimplementasikan menggunakan metodologi CRISP-DM (*Cross-Industry Standard Process for Data Mining*). Metodologi CRISP-DM mengorganisir penelitian ke dalam enam tahap. Pembagian tahapan ini bertujuan untuk memudahkan pemahaman terhadap proses penelitian dan memberikan arahan yang jelas dalam merencanakan dan melaksanakan penelitian (Schröer, dkk, 2021). Gambar 2.5 menunjukkan tahapan model CRISP-DM.



Gambar 2.5 CRISP-DM *(Cross-Industry Standard Process for Data Mining)*
(Chapman Pete, dkk, 2000)

Tanda panah pada Gambar 2.5 menunjukkan aliran serta ketergantungan antara setiap fase atau tahapan dalam CRISP-DM. Posisi data yang berada di bagian tengah pada diagram CRISP-DM memiliki makna bahwa setiap tahap dalam CRISP-DM baik dari tahap *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation*, dan *Deployment* memiliki ketergantungan terhadap ketersediaan dan kualitas data (Chapman Pete, dkk, 2000). Penjelasan lengkap mengenai keseluruhan tahapan-tahapan dalam metodologi CRISP-DM adalah sebagai berikut:

2.2.8.1 *Business Understanding.*

Tahap awal dan paling krusial dalam CRISP-DM adalah pemahaman bisnis. Tahap ini bertujuan untuk memahami tujuan proyek dari sudut pandang bisnis, kemudian mengubah pandangan bisnis ini menjadi definisi masalah penelitian.

2.2.8.2 *Data Understanding.*

Tahap pemahaman data dimulai dengan pengumpulan data awal. Tahap ini bertujuan untuk mengidentifikasi, mengumpulkan, dan menganalisis kumpulan data yang dapat membantu mencapai tujuan proyek penelitian.

2.2.8.3 *Data Preparation.*

Tahap persiapan data melibatkan semua kegiatan yang diperlukan untuk membuat *dataset* final (data yang akan dimasukkan ke dalam model) dari data mentah awal. Tahap ini mencakup pemilihan tabel, dan atribut, serta transformasi dan pembersihan data untuk alat pemodelan.

2.2.8.4 *Modelling.*

Pada tahap ini, berbagai teknik pemodelan dipilih dan diterapkan, serta parameter-parameter dari setiap model disesuaikan untuk mendapatkan model yang optimal.

2.2.8.5 *Evaluation.*

Tahap evaluasi melibatkan analisis mendalam terhadap hasil dari proses pemodelan yang dilakukan sebelumnya. Tujuan tahap ini adalah untuk memahami kualitas model yang telah dibuat dan menyesuaikannya dengan tujuan penelitian yang telah ditetapkan sebelumnya.

2.2.8.6 *Deployment.*

Tahap *Deployment* adalah tahap pembuatan laporan atau presentasi berdasarkan pengetahuan yang diperoleh dari evaluasi yang telah dilakukan pada tahap sebelumnya.