

**SKRIPSI**

**SISTEM PREDIKSI *DWELLING TIME CONTAINERS* DI  
TERMINAL PETI KEMAS (STUDI KASUS PELABUHAN  
SOEKARNO HATTA MAKASSAR)**

**Disusun dan diajukan oleh:**

**TIARA YANIA IFANI LAKITA  
D121 18 1022**



**PROGRAM STUDI SARJANA TEKNIK INFORMATIKA  
FAKULTAS TEKNIK  
UNIVERSITAS HASANUDDIN  
GOWA  
2024**

**LEMBAR PENGESAHAN SKRIPSI**

**SISTEM PREDIKSI DWELLING TIME CONTAINERS DI  
TERMINAL PETIKEMAS (STUDI KASUS PELABUHAN  
SOEKARNO HATTA MAKASSAR)**

Disusun dan diajukan oleh

**Tiara Yania Ifani Lakita  
D121181022**

Telah dipertahankan di hadapan Panitia Ujian yang dibentuk dalam rangka Penyelesaian  
Studi Program Sarjana Program Studi Teknik Informatika  
Fakultas Teknik Universitas Hasanuddin  
Pada tanggal 7 Juni 2024  
dan dinyatakan telah memenuhi syarat kelulusan

Menyetujui,

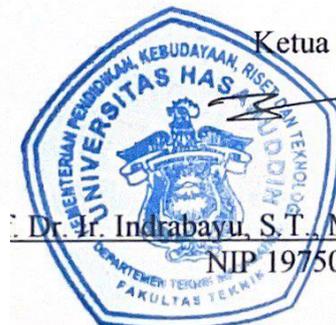
Pembimbing Utama,

Pembimbing Pendamping,

  
Prof. Dr. Ir. Indrabayu, S.T., M.T.,  
M.Bus.Sys., IPM., ASEAN.Eng.  
NIP 19750716 200212 1 004

  
A. Ais Prayogi Alimuddin,  
S.T., M.Eng.  
NIP 19830510 201404 1 001

Ketua Program Studi,



  
Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM., ASEAN.Eng.  
NIP 19750716 200212 1 004



## PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini ;

Nama : Tiara Yania Ifani Lakita  
NIM : D121181022  
Program Studi : Teknik Informatika  
Jenjang : S1

Menyatakan dengan ini bahwa karya tulisan saya berjudul

Sistem Prediksi Dwelling Time Containers di Terminal Petikemas (Studi Kasus  
Pelabuhan Soekarno Hatta Makassar)

Adalah karya tulisan saya sendiri dan bukan merupakan pengambilan alihan tulisan orang lain dan bahwa skripsi yang saya tulis ini benar-benar merupakan hasil karya saya sendiri.

Semua informasi yang ditulis dalam skripsi yang berasal dari penulis lain telah diberi penghargaan, yakni dengan mengutip sumber dan tahun penerbitannya. Oleh karena itu semua tulisan dalam skripsi ini sepenuhnya menjadi tanggung jawab penulis. Apabila ada pihak manapun yang merasa ada kesamaan judul dan atau hasil temuan dalam skripsi ini, maka penulis siap untuk diklarifikasi dan mempertanggungjawabkan segala resiko.

Segala data dan informasi yang diperoleh selama proses pembuatan skripsi, yang akan dipublikasi oleh Penulis di masa depan harus mendapat persetujuan dari Dosen Pembimbing.

Apabila dikemudian hari terbukti atau dapat dibuktikan bahwa sebagian atau keseluruhan isi skripsi ini hasil karya orang lain, maka saya bersedia menerima sanksi atas perbuatan tersebut.

Gowa, 7 Juni 2024

Yang Menyatakan



Tiara Yania Ifani Lakita



## ABSTRAK

**TIARA YANIA IFANI LAKITA.** *Sistem Prediksi Dwelling Time Containers di Terminal Petikemas (Studi Kasus Pelabuhan Soekarno Hatta Makassar)* (dibimbing oleh Indrabayu dan A. Ais Prayogi Alimuddin)

Kinerja logistik di Indonesia masih tertinggal dibandingkan dengan negara tetangga seperti Singapura, Malaysia, Thailand, Vietnam, dan Filipina. Indeks Kinerja Logistik (LPI) menunjukkan bahwa kinerja logistik Indonesia belum efisien. Data terbaru tahun 2023 menunjukkan bahwa Indonesia memiliki kinerja yang lebih buruk dibandingkan dengan negara-negara tersebut dalam semua kategori LPI. Salah satu masalah yang signifikan adalah *dwelling time*, yaitu lamanya waktu bongkar muat kontainer di pelabuhan. Rata-rata *dwelling time* di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar melebihi standar pemerintah yang ditetapkan sebesar 3 hari. Hal ini mempengaruhi produktivitas perusahaan pengiriman barang melalui kontainer. Untuk mengatasi masalah ini, diperlukan pengujian variabel-variabel tertentu dalam pengembangan model sistem prediksi *dwelling time* kontainer di Terminal Petikemas (Studi Kasus: Pelabuhan Soekarno Hatta Makassar)", yang dapat membantu operator terminal dalam mengelola sumber daya secara efisien dan menerapkan kebijakan yang tepat. Variabel-variabel tersebut antara lain Pelabuhan Asal, Status Kontainer, Komoditas, Jenis Kontainer, Ukuran Kontainer, *Gross* dan *Dwelling Time* sebanyak 8.181 baris data. Metode prediksi yang digunakan dalam penelitian ini adalah ANN dan SVR dengan matriks perhitungan *error* menggunakan RMSE dan MAE. Hasil penelitian didapatkan RMSE 1,85 dan MAE 1,3 pada metode SVR serta RMSE 1,73 dan MAE 1,25 pada metode ANN. Hal ini menunjukkan bahwa kedua metode belum dapat memprediksi *dwelling time* dengan akurat dikarenakan kualitas data yang dimiliki.

Kata kunci: *Dwelling Time, Container, SVR, ANN*



## ABSTRACT

**TIARA YANIA IFANI LAKITA.** *Prediction System for Dwelling Time of Containers at Container Terminal (Case Study: Soekarno Hatta Port in Makassar) (supervised by Indrabayu and A. Ais Prayogi Alimuddin)*

*The logistics performance in Indonesia still lags behind neighboring countries such as Singapore, Malaysia, Thailand, Vietnam, and the Philippines. The Logistics Performance Index (LPI) indicates that Indonesia's logistics performance is not yet efficient. The latest data from 2023 shows that Indonesia performs worse than these countries in all LPI categories. One of the significant problem is dwelling time, whis is the length of time for loading and unloading containers at the port. The average dwelling time at the Makassar Soekarno Hatta Port Container Terminal exceeds the government standard set at 3 days. This affects the productivity of companies shipping goods via containers. To overcome this problem, it is necessary to test certain variables in developing a container dwelling time prediction system model at the Container Terminal (Case Study: Soekarno Hatta Port, Makassar)", which can help terminal operators manage resources efficiently and implement appropriate policies. These variables include Port of Origin, Container Status, Commodity, Container Type, Container Size, Gross and Dwelling Time totaling 8,181 rows of data. The prediction method used in this research is ANN and SVR with an error calculation matrix using RMSE and MAE. Results research obtained RMSE 1.85 and MAE 1.3 for the SVR method and RMSE 1.73 and MAE 1.25 for the ANN method. This shows that the two methods cannot predict dwelling time accurately due to the quality of the data they have.*

*Keywords: Dwelling Time, Container, SVR, ANN*



## DAFTAR ISI

LEMBAR PENGESAHAN SKRIPSI .....	iv
PERNYATAAN KEASLIAN.....	iv
ABSTRAK .....	i
ABSTRACT .....	iv
DAFTAR ISI.....	v
DAFTAR GAMBAR .....	vii
DAFTAR TABEL.....	viii
DAFTAR SINGKATAN DAN ARTI SIMBOL .....	ix
DAFTAR LAMPIRAN.....	x
KATA PENGANTAR .....	xi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah.....	2
1.3 Tujuan Penelitian/Perancangan .....	3
1.4 Manfaat Penelitian/Perancangan .....	3
1.5 Ruang Lingkup/Asumsi perancangan.....	4
BAB II TINJAUAN PUSTAKA.....	5
2.1 Petikemas (Container) .....	5
2.2 <i>Dwelling Time</i> .....	5
2.3 Jenis Kontainer .....	6
2.4 Ukuran Kontainer .....	6
2.5 Status Kontainer .....	7
2.6 Komoditi .....	7
2.7 <i>Gross</i> .....	7
2.8 Pelabuhan Asal .....	8
2.9 <i>Knowledge Discovery in Database</i> .....	8
2.10 <i>Machine Learning</i> .....	16
2.11 Peramalan ( <i>forecasting</i> ).....	19
2.12 <i>Support Vector Regression</i> .....	20
2.13 <i>Artificial Neural Network</i> .....	25
2.14 RMSE .....	30
2.15 Penelitian Terkait.....	31
BAB 3 METODE PENELITIAN/PERANCANGAN .....	35
3.1 Tahapan Penelitian .....	35
3.2 Waktu dan Lokasi Penelitian .....	35
3.3 Instrumen Penelitian .....	35
3.4 Teknik Pengambilan Data .....	36
3.5 Perancangan Sistem.....	36
3.6 Prediksi/Peramalan Menggunakan <i>Support Vector Machine</i> .....	44
3.7 Prediksi/Peramalan Menggunakan <i>Artificial Neural Network</i> .....	48
BAB 4 HASIL DAN PEMBAHASAN.....	53
4.1 Dataset .....	53
4.2 Penerapan Algoritma <i>Support Vector Regression</i> .....	57
4.3 Penerapan Algoritma <i>Artificial Neural Network</i> .....	61
4.4 Pembahasan .....	64



BAB 5. KESIMPULAN DAN SARAN ..... 70  
5.1 Kesimpulan..... 70  
5.2 Saran ..... 71  
DAFTAR PUSTAKA ..... 72



## DAFTAR GAMBAR

Gambar 1 Proses Knowledge Discovery in Database.....	9
Gambar 2 Map pengembangan produk App .....	17
Gambar 3 Hyperplane dan margin pada SVR.....	21
Gambar 4 Insensitive zone .....	22
Gambar 5 Ilustasi <i>hyperplane</i> pada dimensi yang lebih tinggi .....	23
Gambar 6 Jaringan Syaraf dengan lapisan tunggal .....	27
Gambar 7 Jaringan Syaraf Tiruan lapisan banyak .....	27
Gambar 8 Jaringan Saraf dengan lapisan kompetitif yang memiliki $-\eta$ .....	28
Gambar 9 Struktur ANN .....	29
Gambar 10 Sampel data Inbound List.....	37
Gambar 11 Sampel data Terminal In & Out .....	37
Gambar 12 Informasi nilai null pada setiap feature .....	40
Gambar 13 Median gross untuk setiap ukuran kontainer.....	40
Gambar 14 Banyaknya data hasil data <i>cleaning</i> .....	42
Gambar 15 Banyaknya data hasil <i>outlier handling</i> .....	43
Gambar 16 Hasil <i>GridSearch</i> SVR.....	45
Gambar 17 <i>Bias term</i> dari metode SVR.....	46
Gambar 18 Informasi persebaran data ( <i>data distribution</i> ).....	54
Gambar 19 Hasil uji ANOVA dataset.....	57
Gambar 20 Hasil <i>tuning</i> parameter SVR .....	59
Gambar 21 Hasil perhitungan nilai error saat evaluasi model SVR .....	61
Gambar 22 Hasil perhitungan nilai error saat evaluasi model ANN .....	64



## DAFTAR TABEL

Tabel 1 Rumus dari setiap tipe kernel pada SVR .....	14
Tabel 2 Rumus dari setiap tipe kernel pada SVR .....	24
Tabel 3 Data yang dipilih dari sumber data <i>Inbound List</i> .....	38
Tabel 4 Data yang dipilih dari sumber data <i>Inbound List</i> .....	38
Tabel 5 Hasil penggabungan dataset .....	39
Tabel 6 Hasil pembersihan <i>missing value</i> .....	41
Tabel 7 Hasil <i>data cleaning</i> .....	42
Tabel 8 Hasil <i>outlier handling</i> .....	43
Tabel 9 Hasil transformasi data .....	44
Tabel 10 <i>Support vectors</i> dari metode SVR .....	45
Tabel 11 Sampel data support vector, Lagrange multiplier, and target value .....	46
Tabel 12 Sampel data yang dipakai untuk melakukan prediksi .....	53
Tabel 13 Sampel data yang memiliki perbedaan pola kombinasi data .....	56
Tabel 14 Informasi korelasi data ( <i>data correlation</i> ) .....	56
Tabel 15 Pembagian data training dan data testing pada metode SVR .....	58
Tabel 16 Hasil <i>training</i> pada prediksi menggunakan metode SVR .....	59
Tabel 17 Hasil prediksi menggunakan metode SVR .....	60
Tabel 18 Hasil scaling data pada metode ANN .....	62
Tabel 19 Pembagian data <i>training</i> dan data testing pada metode ANN .....	62
Tabel 20 Hasil <i>training</i> pada prediksi dengan menggunakan metode ANN .....	63
Tabel 21 Hasil prediksi menggunakan metode ANN .....	63



## DAFTAR SINGKATAN DAN ARTI SIMBOL

Lambang/Singkatan	Arti dan Keterangan
LPI	<i>Logistik Performance Indeks</i>
INSA	<i>Indonesian National Shipowners Association</i>
AIMP	<i>Artificial Intelligence and Multimedia Processing</i>
SVR	<i>Support Vector Regression</i>
ANN	<i>Artificial Neural Network</i>
ISO	<i>Internasional Standar Organization</i>
VGM	<i>Verified Gross Mass</i>
KDD	<i>Knowledge Discovery In Database</i>
ML	<i>Machine Learning</i>
ReLU	<i>Rectified Linear Unit</i>
NB	<i>Naïve Bayes</i>
CDT	<i>Container Dwelling Time</i>
ANOVA	<i>Analysis of Varians</i>
.csv	<i>Comma Separated Values File</i>
RBF	<i>Radial Basis Function</i>
C	Parameter reguralisasi
RMSE	<i>Root Mean Square Error</i>
MAE	<i>Mean Absolute Error</i>



## DAFTAR LAMPIRAN

Lampiran 1 Dataset .....	76
Lampiran 2 <i>Source Code</i> .....	77
Lampiran 3 Dataset Hasil <i>Preprocessing</i> .....	94
Lampiran 4 Hasil Prediksi.....	95



## KATA PENGANTAR

Segala puji dan syukur penulis panjatkan kepada Allah SWT atas segala rahmat dan karunia-Nya, sehingga penulis dapat menyelesaikan tugas akhir dengan judul “Sistem Prediksi *Dwelling Time Containers* di Terminal Petikemas (Studi Kasus Pelabuhan Soekarno Hatta Makassar)”.

Penelitian ini dilatarbelakangi pengamatan penulis terhadap beberapa ketidaksesuaian lamanya waktu singgah kontainer di Pelabuhan Soekarno-Hatta, yang mana belum sesuai dengan aturan dengan aturan dari pemerintah. Sehingga terjadi berbagai masalah logistik yang dapat menghambat jalannya perekonomian di jalur laut. Maka dari itu, penulis bermaksud untuk membuat model prediksi *dwelling time* kontainer di Pelabuhan Soekarno Hatta, dengan harapan dapat membantu operator terminal maupun pihak-pihak terkait untuk dapat mengelola kebijakan yang tepat untuk Pelabuhan yang bersangkutan.

Penulis menyadari bahwa penyusunan dan penulisan tugas akhir ini tidak dapat terselesaikan dengan baik tanpa adanya bantuan, bimbingan, serta dukungan dari berbagai pihak. Oleh karena itu, penulis ingin menyampaikan ucapan terima kasih banyak kepada :

1. Keluarga penulis, Bapak Abubakar A.J. Lakita dan Ibu Titik Priyati selaku kedua orang tua penulis, Tika Septiana Lakita dan Tisa Aprilia Ramadhani Lakita selaku kakak kandung penulis yang senantiasa memberikan doa dan dukungan yang tiada hentinya, serta selalu sabar dan tanpa kenal lelah dalam mendidik penulis selama ini.
2. Bapak Prof. Dr. Ir. Indrabayu, S.T., M.T., M.Bus.Sys., IPM, ASEAN. Eng. selaku pembimbing I dan Bapak A. Ais Prayogi Alimuddin, S.T., M.Eng. selaku pembimbing II, yang senantiasa menyediakan waktu, tenaga, pikiran dan perhatian yang luar biasa dalam mengarahkan penulis untuk menyelesaikan tugas akhir.
3. Segenap staf dan dosen Departemen Teknik Informatika, Fakultas Teknik, Universitas Hasanuddin yang telah membantu untuk kelancaran penyelesaian tugas akhir.
4. Bapak Asrullah yang telah senantiasa membantu penulis selama pengambilan data penelitian di Komando Pusat Otoritas Terminal Petikemas Makassar.
5. Segenap keluarga AIMP *Research Group* Universitas Hasanuddin yang telah memberikan bantuan selama penelitian dan diskusi terkait penyusunan tugas akhir serta memberikan motivasi di masa-masa sulit penyelesaian skripsi.
6. Maulana Bagaswara Marsidi yang selalu memberi bantuan, nasihat dan dukungan yang tulus selama penyelesaian tugas akhir, serta Lilo yang selalu menemani dan menghibur dalam setiap kesempatan.
7. Segenap teman seperjuangan tugas akhir, Indah, Dee, Caca, Lisyia, Aqilah, Lisyia, Mage, kak Imad yang telah membantu memberikan pertolongan, semangat, serta hiburan di masa-masa sulit penyelesaian tugas akhir. Serta pihak-pihak lain yang tidak sempat disebutkan dan tanpa sadar telah menjadi inspirasi maupun membantu penulis dalam menyelesaikan tugas akhir.



# BAB I PENDAHULUAN

## 1.1 Latar Belakang

Indonesia berada pada posisi dimana kinerja logistiknya masih tertinggal dari beberapa negara seperti Singapura, Malaysia, Thailand, Vietnam dan Filipina. Diukur dari komponen Logistik Performance Index (LPI), World Bank menyebutkan bahwa kinerja logistik Indonesia belum efisien. Berdasarkan data terakhir pada tahun 2023, dari keenam kategori dalam LPI, kinerja Indonesia lebih buruk dibandingkan kelima negara tersebut pada semua kategori. Dan dalam beberapa kategori seperti pengiriman internasional, kompetensi logistik, pelacakan dan penelusuran serta ketepatan waktu, Indonesia mengalami penurunan signifikan jika dibandingkan dengan tahun 2018.

Jika dilihat dari Logistics Performance Index (LPI), permasalahan utama tingginya biaya logistik nasional disebabkan dari masalah infrastruktur yang berkontribusi terhadap kelancaran barang di pelabuhan. Hal ini merupakan hambatan di bidang logistik Indonesia yang berdampak pada melemahnya daya saing nasional (Utami, 2015). *Indonesian National Shipowners Association (INSA)* menyebutkan kunci penurunan biaya logistik terdapat pada peningkatan efisiensi dan efektivitas distribusi barang sektor angkutan laut melalui pelabuhan. Sedangkan rendahnya daya saing infrastruktur, memberikan kontribusi terhadap kurang lancarnya arus distribusi barang. Kondisi infrastruktur di pelabuhan khususnya pada Pelabuhan Soekarno Hatta yang ada sekarang ini dinilai masih kurang memadai.

Lemahnya dukungan sektor logistik nasional tersebut memicu berbagai permasalahan dalam distribusi barang dikarenakan kurangnya efisiensi pelayanan kepabeanan serta infrastruktur terutama terkait masalah lamanya waktu bongkar muat barang di pelabuhan (*dwelling time*). *Dwelling time* merupakan ukuran waktu yang dibutuhkan kontainer sejak kontainer dibongkar dari waktu kapal sandar

) sampai dengan keluar dari kawasan pelabuhan (*gate out*). Selama ini, *dwelling time* di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar mencapai 5-6 hari, yang mana belum memenuhi standar yang ditetapkan oleh



pemerintah yakni maksimal 3 hari. Sedangkan permasalahan perusahaan yang bergerak di bidang jasa pengiriman barang melalui kontainer adalah produktivitas proses bongkar dan proses muat kontainer di pelabuhan yang dipengaruhi proses *dwelling time*. Tingginya nilai *dwelling time* terjadi karena beberapa faktor salah satu yang paling dominan adalah ketidaksiapan pelabuhan dalam mengantisipasi arus barang dan masalah operasional lainnya.

Berkaitan dengan aturan pemerintah, permasalahan mengenai *dwelling time* ini perlu dikaji agar dapat mengevaluasi proses bongkar muat kontainer. Hal ini penting dilakukan agar tidak terjadi penumpukan kapal yang ingin berlabuh di pelabuhan Soekarno Hatta Makassar. Karena penumpukan kapal dapat menjadi reaksi berantai untuk kapal-kapal setelahnya sehingga membuat jalannya operasional dari seluruh pihak terkait menjadi tidak teratur.

Di sisi lain, terdapat berbagai parameter menyangkut kontainer yang diasumsikan dapat mempengaruhi *dwelling time* antara lain hari dan bulan keluar, pelabuhan asal, ukuran dan jenis kontainer, serta *yard occupation ratio* terminal. Maka dari itu diperlukan untuk menguji berbagai parameter-parameter yang ada untuk memprediksi *dwelling time*. Hal ini diharapkan dapat menjadi bahan masukan yang dapat membantu operator terminal untuk dapat mengelola sumber daya mereka secara efisien dan menerapkan kebijakan yang tepat pada waktu dan tempat yang sesuai. Untuk itu diusulkan judul “**Sistem Prediksi *Dwelling Time Containers* di Terminal Petikemas (Studi Kasus Pelabuhan Soekarno Hatta Makassar)**” sebagai bentuk evaluasi pengelolaan kontainer di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar.

## 1.2 Rumusan Masalah

Berdasarkan pada latar belakang masalah di atas, maka perumusan masalah dalam penelitian ini adalah sebagai berikut:

- a. Bagaimana tahapan prediksi *dwelling time* dengan metode-metode *forecasting* menggunakan parameter-parameter yang diolah dari Terminal Petikemas Pelabuhan Soekarno Hatta Makassar?



- b. Bagaimana perbandingan hasil prediksi *dwelling time* dengan metode-metode *forecasting* di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar?
- c. Apakah parameter-parameter yang digunakan kompatibel untuk memprediksi *dwelling time* di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar?

### 1.3 Tujuan Penelitian/Perancangan

Tujuan yang diharapkan dalam penelitian ini adalah sebagai berikut:

- a. Menjelaskan tahapan prediksi *dwelling time* dengan metode-metode *forecasting* menggunakan parameter-parameter yang diolah dari Terminal Petikemas Pelabuhan Soekarno Hatta Makassar.
- b. Menganalisa dan membandingkan hasil prediksi metode-metode *forecasting* menggunakan parameter-parameter yang diolah Terminal Petikemas Pelabuhan Soekarno Hatta Makassar .
- c. Menyimpulkan kompatibilitas parameter-parameter yang ada terhadap prediksi *dwelling time* di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar.

### 1.4 Manfaat Penelitian/Perancangan

Dengan dilakukannya penelitian ini, diharapkan dapat memberikan manfaat sebagai berikut:

- a. Akademisi
  - i. Penelitian ini diharapkan dapat menjadi referensi bagi akademisi maupun peneliti berikutnya dalam melakukan penelitian yang terkait dengan tema yang diteliti.
- b. Perusahaan
  - i. Penelitian ini diharapkan dapat menjadi bahan masukan yang berkaitan dengan judul yang diteliti.
  - ii. Diharapkan dapat menjadi acuan bagi operator terminal dapat mengelola sumber daya mereka secara efisien dan menerapkan kebijakan yang tepat pada waktu dan tempat yang sesuai.



## 1.5 Ruang Lingkup/Asumsi perancangan

- a. Data yang digunakan adalah data Terminal Petikemas Makassar, menggunakan dengan data yang dimiliki oleh sistem terapan baru perekaman data-data kapal kontainer di terminal petikemas yang bersangkutan.
- b. Pembangunan sistem akan menggunakan metode SVR dan ANN, dan menghasilkan model yang akan dipakai untuk memprediksi *dwelling time*
- c. Variabel-variabel penelitian terdiri dari:
  - i. Ukuran Kontainer
  - ii. Tipe/Jenis Kontainer
  - iii. Status Kontainer
  - iv. Komoditas
  - v. Pelabuhan Asal
  - vi. *Gross*
  - vii. *Dwelling Time*.



## BAB II TINJAUAN PUSTAKA

### 2.1 Petikemas (Container)

Petikemas adalah peti berbentuk empat persegi panjang yang dirancang khusus dengan ukuran tertentu terbuat dari besi maupun aluminium serta memiliki pintu di salah satu sisinya serta dapat digunakan berulang kali juga digunakan sebagai tempat untuk menyimpan sekaligus mengangkut muatan yang ada di dalamnya dan telah ditetapkan berdasarkan standar internasional (ISO).

Filosofi dibalik petikemas adalah membungkus atau membawa muatan dalam peti-peti yang sama dan membuat semua kendaraan dapat mengangkutnya sebagai satu kesatuan, baik kendaraan itu berupa kapal laut, kereta api, truk, atau angkutan lainnya, dan dapat membawa secara cepat, dan efisien atau bila mungkin dari pintu ke pintu (*door to door*) (Afiffudin, 2019).

### 2.2 *Dwelling Time*

*Dwelling time* memiliki beberapa pengertian berdasarkan berbagai sumber karena definisi dwell time sendiri dapat berasal dari aspek manapun khususnya aspek pengangkutan. *Dwelling time* merupakan gambaran umum kecepatan proses pelayanan logistik dalam Pelabuhan sejak dilakukan pembongkaran barang impor (*discharge*) sampai dengan pengeluaran barang impor dari pelabuhan (Ramiaji, 2017).

Menurut *World Bank* (2011), *dwelling time* adalah waktu yang dihitung mulai dari suatu petikemas dibongkar muat dari kapal sampai petikemas tersebut meninggalkan terminal pelabuhan melalui pintu utama. Sedangkan bea & cukai berpendapat bahwa *dwelling time* adalah waktu berapa lama petikemas barang impor ditimbun di tempat penimbunan sementara di pelabuhan sejak dibongkar dari kapal sampai dengan barang impor keluar dari tempat penimbunan sementara (Azizah, 2019).



### 2.3 Jenis Kontainer

Jenis kontainer adalah media kemas yang dibuat dengan menyesuaikan fungsi, ukuran serta kegunaannya. Menurut Edy Hidayat pada tahun 2009 dalam bukunya, beberapa jenis petikemas antara lain:

- a. *General Cargo* adalah petikemas yang dipakai untuk mengangkut muatan umum (*General Cargo*).
- b. *Thermal Container* adalah petikemas yang dilengkapi dengan pengaturan suhu untuk muatan tertentu.
- c. *Tank Container* adalah tangkai yang ditempatkan dalam kerangka petikemas yang dipergunakan untuk muatan cair (*bulk liquid*) maupun gas (*bulk gas*).
- d. *Dry Bulk Container* adalah kontainer umum yang dipergunakan khusus untuk mengangkut muatan curah (*bulk cargo*).
- e. *Platform Container* adalah petikemas yang terdiri dari lantai dasar.

### 2.4 Ukuran Kontainer

Menurut Edy Hidayat pada tahun 2009, agar pengoperasian petikemas dapat berjalan dengan baik, maka semua pihak yang terlibat harus menyetujui agar ukuran-ukuran petikemas harus sama dan sejenis serta mudah diangkut. Adapun Badan *Internasional Standar Organization* (ISO) telah menetapkan ukuran-ukuran dari petikemas sebagai berikut (Afiffudin, 2019):

- a. *Container 20' Dry Freight (20 Feet)*

Ukuran Luarnya : 20' [p] x 8' [l] x 8,6' [t] atau 6.058 x 2.438 x 2.591M.

Ukuran Dalamnya : 5.919 x 2.340 x 2.380 M.

Kapasitasnya : *Cubic Capacity* : 33 Cbm. Pay Load : 22.1 Ton.

- b. *Container 40' Dry Freight (40 Feet)*

Ukuran Luarnya : 40' [p] x 8' [l] x 8,6' [t] atau 12.192 x 2.438 x 2.591M.

Ukuran Dalamnya : 12.045 x 2.309 x 2.379 M.

Kapasitasnya : *Cubic Capacity* : 67.3 Cbm.

*Pay Load* : 27.396 Ton

- c. *Container 40' High Cube Dry*

Ukuran Luarnya : 40' [p] x 8' [l] x 9,6' [t] atau 12.192 x 2.438 x 2.926M.



Ukuran Dalamnya : 12.056 x 2.347 x 2.684 M.

Kapasitasnya : *Cubic Capacity* : 76 Cbm.

*Pay Load* : 29.6 Ton.

## 2.5 Status Kontainer

Status kontainer adalah status atau keadaan kontainer yang dimuat. Dalam hal ini ada 2 status kontainer yang umumnya didokumentasikan, yaitu (Anggia, 2019):

- a. *Full*, yaitu keadaan dimana kontainer terisi penuh dengan berbagai komoditas.
- b. *Empty*, yaitu keadaan kontainer yang kosong karena tidak ada komoditas yang dimuat. Biasanya kontainer ini dimuat di kapal untuk dikembalikan setelah pembongkaran dan tidak ada komoditas yang dimuat kembali. Adapun untuk bongkaran kontainer *empty* biasanya langsung dipindahkan oleh pemilik kontainer ke depo kontainer, karena menghindari biaya penimbunan di CY yang cukup tinggi dan progresif.

## 2.6 Komoditi

Menurut UU RI No. 10 Tahun 2011 tentang Perdagangan Berjangka Komoditi, komoditi adalah semua barang, jasa, hak dan kepentingan lainnya, dan setiap derivatif dari Komoditi, yang dapat diperdagangkan dan menjadi subjek Kontrak Berjangka, Kontrak Derivatif Syariah, dan/atau Kontrak Derivatif lainnya.

## 2.7 Gross

Berat Kotor Petikemas (*Gross Weight*) adalah jumlah keseluruhan berat kotor dari kemasan petikemas, yaitu berat kontainer ditambah dengan berat komoditi atau isi di dalamnya. Contoh berat barang 500 Kg dan berat kontainer 1000 Kg maka G.W : 1.500 Kg.

Adapun yang dinamakan Berat Kotor Petikemas Terverifikasi (*Verified Gross Mass/ VGM*) sebagaimana yang ditetapkan dalam Peraturan Menteri Perdagangan Republik Indonesia Nomor Pm 53 Tahun 2018 Tentang Kelaikan dan Berat Kotor Petikemas Terverifikasi. VGM adalah jumlah keseluruhan berat kotor dari kemasan petikemas yang diperoleh melalui salah satu



metode yang ditetapkan (Peraturan Menteri Perhubungan Republik Indonesia Nomor PM 25 Tahun 2022 tentang Kelaikan Petikemas dan Berat Kotor Petikemas Terverifikasi).

## 2.8 Pelabuhan Asal

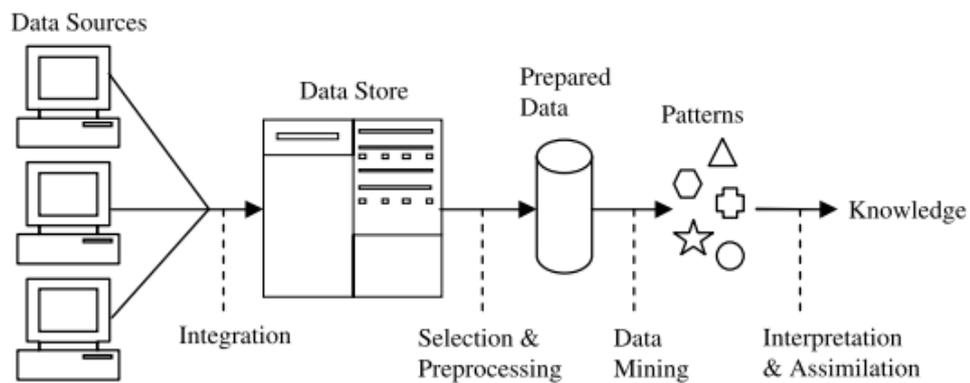
Pelabuhan adalah daerah perairan yang terlindung terhadap gelombang, yang dilengkapi dengan fasilitas terminal laut meliputi dermaga dimana kapal dapat bertambat untuk bongkar muat barang, kran-kran (*Cren*) untuk bongkar muat barang, gudang laut dan tempat-tempat menyimpan dimana kapal membongkar muatannya, dan gudang-gudang dimana barang-barang dapat disimpan dengan waktu yang lebih lama selama menunggu pengirim ke daerah tujuan atau pengapalan (Triatmojo, 2010).

Adapun pelabuhan asal atau *original port* atau *port of loading* yang disebutkan dalam penelitian ini adalah pelabuhan dimana kontainer pertama kali dimuat atau pelabuhan asal kontainer yang bersangkutan. Selain itu ada namanya *port of discharge* yang merupakan pelabuhan tempat kontainer yang bersangkutan dibongkar.

## 2.9 Knowledge Discovery in Database

*Knowledge Discovery in Database* (KDD) merupakan metode untuk memperoleh pengetahuan dari *database* yang ada. Dalam *database* terdapat tabel - tabel yang saling berhubungan / berelasi. Hasil pengetahuan yang diperoleh dalam proses tersebut dapat digunakan sebagai basis pengetahuan (*knowledge base*) untuk keperluan pengambilan keputusan. Istilah *Knowledge Discovery in Database* (KDD) dan *data mining* seringkali digunakan secara bergantian untuk menjelaskan proses penggalian informasi tersembunyi dalam suatu basis data yang besar. Sebenarnya kedua istilah tersebut memiliki konsep yang berbeda, tetapi berkaitan satu sama lain, dan salah satu tahapan dalam keseluruhan proses KDD adalah *data mining* (Mardi, 2014).





Gambar 1 Proses Knowledge Discovery in Database

Sumber: Bramer, 2020

Jika mengacu pada Gambar 1, *Knowledge Discovery in Database* menyangkut seluruh proses ekstraksi pengetahuan, termasuk bagaimana data disimpan dan diakses, bagaimana menggunakan algoritma yang efisien dan terukur untuk menganalisis dataset besar, bagaimana menafsirkan dan memvisualisasikan hasil, dan bagaimana memodelkan dan mendukung interaksi antara manusia dan mesin. Ini juga menyangkut dukungan untuk mempelajari dan menganalisis domain aplikasi. Proses KDD secara garis besar dapat dijelaskan sebagai berikut.

### 2.9.1 Data Collection

*Data Collection* adalah proses mengumpulkan dan mengukur informasi tentang variabel yang ditargetkan dalam sistem yang mapan, yang kemudian memungkinkan seseorang untuk menjawab pertanyaan yang relevan dan mengevaluasi hasil. Pengumpulan Data pada penelitian ini adalah proses pengumpulan data dari Biro yang diperlukan sebagai dataset. Proses pengumpulan data ini dilakukan dengan datang langsung ke Biro dengan melakukan wawancara dan meminta data secara jamaah ke bagian administrasi (Yunitasari dkk., 2022).

### 2.9.2 Data Selection dan Preprocessing

Data yang sudah ada di *database* seringkali tidak semuanya dibutuhkan, maka dibutuhkan penyeleksian data untuk data yang benar-benar relevan dan an dalam proses selanjutnya. Data hasil seleksi yang akan digunakan oses *data mining*, disimpan dalam suatu berkas terpisah dari basis data nal (Wahyudi dkk., 2022).



Sedangkan *preprocessing* adalah proses perubahan bentuk data yang tidak terstruktur menjadi data yang terstruktur agar sesuai kebutuhan untuk diproses selanjutnya. Data yang telah melewati *preprocessing* adalah data yang lebih terstruktur (Veriane dkk., 2023).

Ada beberapa langkah yang dapat dilakukan dalam tahap *preprocessing*, di antaranya sebagai berikut (Assena, 2020):

- a. *Data Cleaning*, yaitu proses untuk mengisi *missing value* atau isian yang hilang atau kurang pas, mengoreksi data yang tidak konsisten, dan mendeteksi redundansi data yaitu penumpukan data atau duplikasi data
- b. *Data Integration*, yaitu suatu proses untuk menggabungkan data dari beberapa *file* sumber. *Data integration* ini hanya dilakukan apabila data yang akan diolah bersumber dari beberapa *file* sumber.
- c. *Data Transformation*, yaitu proses mengubah suatu data supaya mendapatkan data yang lebih berkualitas atau sesuai dengan kebutuhan. Dalam proses ini peneliti hanya akan melakukan proses *transformation* data pada *file* pertama dan *file* kedua.
- d. *Data Reduction*, yaitu proses untuk mengurangi atau mereduksi sejumlah data yang tidak dibutuhkan. *Data reduction* sangat berguna untuk mendapatkan atribut dan data yang akan digunakan sesuai keperluan.

### 2.9.3 Data Mining

Menurut *Gartner Group*, *data mining* adalah proses menemukan hubungan baru yang mempunyai arti, pola dan kebiasaan dengan memilah-milah sebagian besar data yang disimpan dalam media penyimpanan dengan menggunakan teknologi pengenalan pola seperti teknik statistik dan matematika (Larose dkk., 2005). Data yang diproses dengan teknik *data mining* akan menciptakan suatu pengetahuan ilmu baru yang bersumber dari data lama, hasil yang diperoleh dari pemrosesan data tersebut bisa digunakan untuk menentukan keputusan di masa depan (Santosa, 2007). *Data mining* merupakan bagian dari proses penemuan

uan dari basis data *Knowledge Discovery in Databases*.

*ata mining* dibagi menjadi beberapa kelompok berdasarkan tugas yang dilakukan (Rosmini dkk., 2018), yaitu:



a. Deskripsi

Para peneliti dan analisis secara sederhana mencoba menemukan cara untuk menggambarkan pola dan kecenderungan trend yang tersembunyi dalam data. Deskripsi dari pola dan kecenderungan tersebut seringkali memberikan kemungkinan penjelasan untuk suatu pola tertentu.

b. Estimasi

Estimasi adalah keseluruhan proses yang memerlukan serta menggunakan *estimator* untuk menghasilkan sebuah *estimation* dari suatu parameter. Sedangkan menurut Kamus Besar Bahasa Indonesia (KBBI) Estimasi adalah perkiraan, penilaian atau pendapat. Ini menunjukkan bahwa istilah estimasi dapat digunakan secara umum untuk menyatakan perkiraan, penilaian atau pendapat mengenai sesuatu (Nainel dkk., 2020).

Teknik untuk melakukan estimasi terhadap sebuah data baru yang tidak memiliki keputusan berdasarkan histori data yang telah ada, dimana variabel target estimasi lebih ke arah numerik daripada ke arah kategori. Model dibangun menggunakan *record* lengkap yang menyediakan nilai dari variabel target sebagai nilai prediksi. Selanjutnya, pada peninjauan berikutnya estimasi nilai dari variabel target dibuat berdasarkan nilai dari variabel prediksi. Contohnya melakukan estimasi tekanan darah sistolik pada pasien rumah sakit berdasarkan umur pasien, jenis kelamin, indeks berat badan, dan level sodium darah. Hubungan antara tekanan darah sistolik dan nilai variabel prediksi dalam proses pembelajaran akan menghasilkan model estimasi. Model estimasi yang dihasilkan dapat digunakan untuk memprediksi kasus baru lainnya (Pradnyana dkk., 2018).

c. Prediksi

Prediksi memiliki kemiripan dengan klasifikasi dan estimasi, namun dalam prediksi nilai dari hasil akan ada di masa yang akan datang. Beberapa algoritma dan teknik yang digunakan dalam klasifikasi dan estimasi dapat pula digunakan (untuk keadaan yang tepat) untuk prediksi (Santoso, 2007).

lasifikasi

Suatu teknik dengan melihat pada kelakuan dan atribut dari kelompok yang telah didefinisikan. Teknik ini dapat memberikan klasifikasi



pada data baru dengan memanipulasi data yang ada yang telah diklasifikasi dan dengan menggunakan hasilnya untuk memberikan sejumlah aturan. Salah satu contoh yang mudah dan populer adalah dengan decision tree yaitu salah satu teknik klasifikasi yang paling populer karena mudah untuk interpretasi seperti Algoritma C4.5, ID3 dan lain-lain. Contoh pemanfaatannya misalnya pada bidang akademik terkait klasifikasi siswa yang layak masuk ke dalam kelas unggulan atau akselerasi di sekolah tertentu (Pradnyana at al., 2018).

Dalam klasifikasi variabel, tujuan bersifat kategorik. Sebagai contoh penggolongan dataset terbagi ke dalam dua kelas (*binary*) ataupun lebih dari dua kelas (*multiclass*).

e. *Clustering*

Teknik untuk mengelompokkan data dan membentuk kelas objek-objek yang memiliki kemiripan. Kluster adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record-record* dalam kluster lain. Proses klusterisasi berbeda dengan proses klasifikasi yaitu tidak adanya variabel target dalam pengklasteran. Proses klusterisasi mencoba untuk melakukan pembagian terhadap keseluruhan data menjadi kelompok-kelompok yang memiliki kemiripan (homogen), dimana kemiripan data dalam satu kelompok tinggi (maksimal) dan kemiripan data dengan data pada kelompok lain rendah (minimal). Contoh klusterisasi misalkan proses mendapatkan kelompok-kelompok konsumen untuk target pemasaran dari suatu produk bagi perusahaan yang tidak memiliki dana pemasaran yang besar (Pradnyana dkk., 2018).

*Clustering* merupakan teknik pengelompokan *record* data, pengamatan atau kasus dalam kelas yang memiliki kemiripan. *Cluster* adalah kumpulan *record* yang memiliki kemiripan satu dengan yang lainnya dan memiliki ketidakmiripan dengan *record* lain dalam *cluster*.

f. Asosiasi



Teknik untuk mengenali kelakuan dari kejadian-kejadian khusus atau proses dimana hubungan asosiasi muncul pada setiap kejadian. Adapun teknik pemecahan masalah yang sering digunakan seperti Algoritma

Apriori. Contoh pemanfaatan Algoritma apriori yaitu pada bidang *marketing* ketika sebuah Minimarket melakukan tata letak produk yang dijual berdasarkan produk-produk mana yang paling sering dibeli konsumen, selain itu seperti tata letak buku yang dilakukan pustakawan di perpustakaan (Pradnyana dkk., 2018).

Mengidentifikasi hubungan antara berbagai peristiwa yang terjadi pada satu waktu. Dalam dunia bisnis lebih umum disebut sebagai analisis keranjang belanja.

Menurut Sihombing pada tahun 2012, *data mining* memiliki kelebihan sebagai alat analisis seperti :

- a. *Data mining* ini mampu menangani data-data dalam jumlah besar dan juga kompleks.
- b. *Data mining* juga dapat menangani data-data dengan berbagai macam tipe atribut.
- c. *Data mining* mampu mencari dan mengolah data secara otomatis. Disebut semi otomatis karena dalam beberapa teknik *data mining*, diperlukan parameter yang harus di *input* oleh *user* secara manual.
- d. *Data mining* dapat menggunakan pengalaman ataupun kesalahan terdahulu untuk meningkatkan kualitas dan hasil analisa sehingga mendapat hasil yang terbaik.

Menurut Sihombing pada tahun 2012, *data mining* memiliki kelemahan dalam pencarian data tidak mencari secara individualis, tetapi set individualis, atau dengan kata lain dikelompokkan dengan kriteria-kriteria tertentu (Tarigan dkk., 2022).

## 2.10 ANOVA Table

Analisis varians (*analysis of variance*) atau ANOVA adalah suatu metode analisis statistika yang termasuk ke dalam cabang statistika inferensi. Uji dalam menggunakan uji F karena dipakai untuk pengujian lebih dari 2 sampel. (*Analysis of variances*) digunakan untuk melakukan analisis komparasi abel. Teknik analisis komparatif dengan menggunakan tes “t” yakni mencari perbedaan yang signifikan dari dua buah mean hanya efektif bila



jumlah variabelnya dua. Untuk mengatasi hal tersebut ada teknik analisis komparatif yang lebih baik yaitu *Analysis of Variances* yang disingkat ANOVA. ANOVA digunakan untuk membandingkan rata-rata populasi bukan ragam populasi. Adapun asumsi dasar yang harus terpenuhi dalam analisis varian adalah (Santoso 2008):

1. Kenormalan

Distribusi data harus normal, agar data berdistribusi normal dapat ditempuh dengan cara memperbanyak jumlah sampel dalam kelompok.

2. Kesamaan variansi

Setiap kelompok hendaknya berasal dari populasi yang sama dengan variansi yang sama pula. Bila banyaknya sampel sama pada setiap kelompok maka kesamaan variansinya dapat diabaikan. Tapi bila banyak sampel pada masing-masing kelompok tidak sama maka kesamaan variansi populasi sangat diperlukan.

3. Pengamatan bebas

Sampel hendaknya diambil secara acak (*random*), sehingga setiap pengamatan merupakan informasi yang bebas.

Tabel 1 Contoh tabel ANOVA

	df	SS	MS	F	Significance F
Regression	1	2650706,899	2650707	1,606271	0,214454864
Residual	31	51156927,4	1650223	-	-
Total	32	53807634,3	-	-	-

Sumber: Budi, 2020

Tabel 1 menunjukkan bagaimana contoh penampakan tabel ANOVA. Tabel ANOVA (*Analysis of Variance*) menguji penerimaan (*acceptability*) model dari perspektif statistik dalam bentuk analisis sumber keragaman. ANOVA ini sering juga diterjemahkan sebagai analisis ragam (Budi, 2020).

Dari tabel ANOVA tersebut diungkapkan bahwa keragaman data aktual terikat (Jumlah penduduk miskin) bersumber dari model regresi dan dari Dalam pengertian sederhana untuk kasus ini adalah variasi (turun-naiknya atau kecilnya) jumlah penduduk miskin disebabkan oleh variasi dari luas



daerah (model regresi) serta dari faktor-faktor lainnya yang mempengaruhi jumlah penduduk miskin yang tidak dimasukkan dalam model regresi (residual).

*Degree of Freedom* (df) atau derajat bebas dari total adalah  $n-1$ , dimana  $n$  adalah banyaknya observasi. Dalam hal ini banyaknya observasi adalah 33 maka derajat bebas total adalah 32. Derajat bebas dari model regresi adalah 1, karena ada satu variabel bebas dalam model ini (luas daerah). Derajat bebas untuk residual adalah sisanya yaitu derajat bebas total – derajat bebas regresi =  $33 - 1 = 32$ .

Kolom SS (*Sum of Square*) atau jumlah kuadrat untuk *regression* diperoleh dari penjumlahan kuadrat dari prediksi variabel terikat (Jumlah penduduk miskin) dikurangi dengan nilai rata-rata jumlah penduduk miskin dari data sebenarnya. Jadi secara manual mencari terlebih dahulu rata-rata permintaan dari data asli. Kemudian masing-masing prediksi jumlah penduduk miskin dikurangi dengan rata-rata tersebut kemudian dikuadratkan. Selanjutnya, seluruh hasil perhitungan tersebut dijumlahkan. Pada data di atas nilai dari SS *regression* adalah 2650706,899.

Kolom SS untuk residual diperoleh dari jumlah pengkuadratan dari residual. Hasil perhitungan tersebut kemudian dijumlahkan. Berdasarkan output di atas maka nilai SS total adalah 53807634,3 Hasil ketika SS tersebut memiliki arti dimana apabila SS total yang diperoleh adalah 53807634,3 yang memiliki arti, variasi dari jumlah penduduk miskin yang dikuadratkan adalah sebesar nilai tersebut. Bervariasinya jumlah penduduk miskin disebabkan oleh sebagian berasal dari variabel bebas (luas daerah) yaitu sebesar 2650706,899 (regresi). Kemudian sisanya sebesar 51156927,4 disebabkan oleh variabel lain yang juga mempengaruhi jumlah penduduk miskin tetapi tidak dimasukkan dalam model (residual).

Jika membandingkan (bagi) antara SS regresi dengan SS total, maka akan didapatkan proporsi dari total variasi jumlah penduduk miskin yang disebabkan oleh variasi luas daerah. Praktikan mencoba membagi antara nilai SS regresi dengan SS total yaitu  $2650706,899 / 53807634,3 = 0,049$ . hasil tersebut sama dengan hasil dari  $R^2$  atau koefisien determinasi yang telah dibahas di atas.



anjutnya kolom berikutnya dari ANOVA adalah kolom MS (*Mean of* atau rata-rata jumlah kuadrat. Ini adalah hasil bagi antara kolom SS dengan  $\text{df}$ . Berdasarkan hasil di atas besarnya MS adalah regresi dan residual

berturut-turut adalah 2650707 dan 1650223. Dari perhitungan MS ini, selanjutnya dengan membagi antara MS Regresi dengan MS Residual didapatkan nilai F. Nilai F ini yang dikenal dengan F hitung dalam pengujian hipotesa dibandingkan dengan nilai F tabel. Jika F hitung  $>$  F tabel, maka dapat dinyatakan bahwa secara simultan (bersama-sama) luas daerah berpengaruh signifikan terhadap jumlah penduduk miskin. Setelah melakukan pembagian antara MS regresi dengan residual didapatkan hasil nilai F hitung yaitu 1,606271 seperti pada hasil analisis. Selain itu, dapat juga membandingkan antara taraf nyata dengan p-value (dalam istilah Excel adalah *Significance F*). Jika taraf nyata  $>$  dari *p-value* maka kesimpulannya sama dengan di atas.

## 2.11 Machine Learning

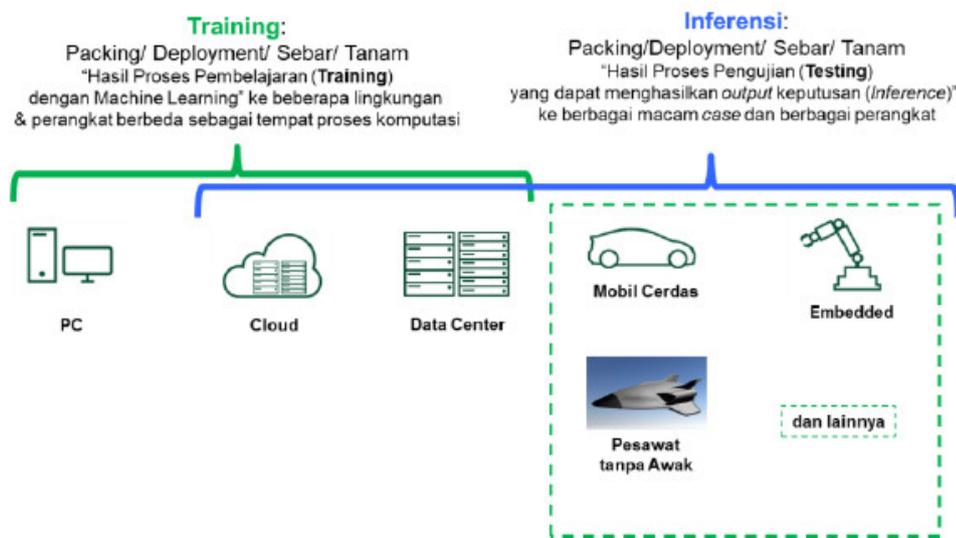
*Machine Learning* (ML) atau pembelajaran mesin merupakan pendekatan dalam AI yang banyak digunakan untuk menggantikan atau menirukan perilaku manusia untuk menyelesaikan masalah atau melakukan otomatisasi. Sesuai namanya, ML mencoba menirukan bagaimana proses manusia atau makhluk cerdas belajar dan menggeneralisasi.<sup>1</sup> Setidaknya ada dua aplikasi utama dalam ML yaitu, klasifikasi dan prediksi. Ciri khas dari ML adalah adanya proses pelatihan, pembelajaran, atau *training*. Oleh karena itu, ML membutuhkan data untuk dipelajari yang disebut sebagai data *training*. Klasifikasi adalah metode dalam ML yang digunakan oleh mesin untuk memilah atau mengklasifikasikan objek berdasarkan ciri tertentu sebagaimana manusia mencoba membedakan benda satu dengan yang lain. Sedangkan prediksi atau regresi digunakan oleh mesin untuk menerka keluaran dari suatu data masukan berdasarkan data yang sudah dipelajari dalam *training*. Metode ML yang paling populer yaitu Sistem Pengambil Keputusan, *Support Vector Machine* (SVM) dan Neural Network.

*Machine Learning* atau Mesin Pembelajaran adalah cabang dari AI yang fokus belajar dari data (learn from data), yaitu fokus pada pengembangan sistem yang mampu belajar secara “mandiri” tanpa harus berulang kali diprogram manusia. ML

hkan Data yang valid sebagai bahan belajar (ketika proses *training*) digunakan ketika testing untuk hasil *output* yang optimal (Cholissodin,



Hasil pengembangan produk berbasis AI (*Machine Learning* dan Teknik lainnya misal dengan Optimasi, etc) ini harapannya dapat lebih memberikan kemudahan dan langsung dapat diterapkan di masyarakat luas atau bahkan masuk ke industri dalam skala nasional dan internasional. Gambar 2 berikut merupakan ilustrasi Map produk untuk *apply* ke masyarakat.



Gambar 2 Map pengembangan produk App

Sumber: Cholissodin, 2012

Penelitian terkini mengungkapkan bahwa machine learning terbagi menjadi tiga kategori: *Supervised Learning*, *Unsupervised Learning*, *Reinforcement Learning* (Somvanshi & Chavan, 2016). Berikut penjelasan kategori machine learning secara lebih detail.

### 2.11.1 Supervised Learning

Teknik yang digunakan oleh *supervised learning* adalah metode klasifikasi dimana kumpulan data sepenuhnya diberikan label untuk mengklasifikasikan kelas yang tidak dikenal (Thupae dkk., 2018).

Kumpulan sampel digunakan untuk meringkas karakteristik distribusi ukuran perilaku dalam setiap jenis aplikasi sehingga membentuk model perilaku dari data

(k., 2011). *Supervised learning* dikelompokkan lebih lanjut dalam masalah klasifikasi dan regresi. Masalah klasifikasi adalah ketika variabel *output* berbentuk kategori seperti merah atau biru atau penyakit dan tidak ada penyakit. Sedangkan



masalah regresi adalah ketika variabel *output* adalah nilai riil, seperti *dollar* atau berat (Brownlee, 2016).

### 2.11.2 Unsupervised Learning

Teknik yang digunakan oleh *unsupervised learning* sering disebut cluster dikarenakan tidak ada kebutuhan untuk pemberian label dalam kumpulan data dan hasilnya tidak mengidentifikasi contoh di kelas yang telah ditentukan.

Dalam jenis pembelajaran *unsupervised learning*, sistem disediakan dengan beberapa *input* sampel tetapi tidak ada output yang hadir. Karena tidak ada output yang diinginkan di sini kategorisasi dilakukan sehingga algoritma membedakan dengan benar antara kumpulan data. Ini adalah tugas mendefinisikan fungsi untuk menggambarkan struktur yang tersembunyi dari data yang tidak berlabel (Somvanshi & Chavan, 2016). *unsupervised learning* dikelompokkan lebih lanjut dalam masalah *clustering* dan asosiasi. Masalah pengelompokan (*clustering*) adalah tempat untuk menemukan pengelompokan yang melekat dalam data, seperti mengelompokkan pelanggan berdasarkan pada perilaku pembelian. Sedangkan masalah asosiasi adalah aturan yang menggambarkan sebagian besar data yang ada, seperti orang yang membeli A juga cenderung membeli B (Brownlee, 2016).

### 2.11.3 Reinforcement Learning

*Reinforcement Learning* biasanya berada antara *supervised learning* dan *unsupervised learning*, teknik ini bekerja dalam lingkungan yang dinamis dimana konsepnya harus menyelesaikan tujuan tanpa adanya pemberitahuan dari komputer secara eksplisit jika tujuan tersebut telah tercapai (Das & Nene, 2017).

*Reinforcement learning* berasal dari teori belajar hewan. Pembelajaran ini tidak memerlukan pengetahuan sebelumnya, dapat secara mandiri mendapatkan kebijakan opsional dengan pengetahuan yang diperoleh melalui coba-coba dan terus berinteraksi dengan lingkungan yang dinamis (Qiang & Zhongli, 2011).

Masalah *reinforcement learning* diselesaikan dengan mempelajari pengalaman melalui trial-and-error (Mahmud dkk., 2018). Algoritma reinforcement terkait dengan algoritma pemrograman dinamis yang sering digunakan menyelesaikan masalah optimisasi (Mitchell, 1997).



## 2.12 Peramalan (*forecasting*)

Peramalan merupakan suatu kegiatan untuk memperkirakan suatu kejadian apa yang akan terjadi pada masa yang akan datang. Peramalan dapat dikatakan sebagai awal dari sebuah proses pengambilan keputusan. Sebelum melakukan sebuah peramalan, hendaknya harus diketahui dahulu apa maksud dan tujuan dari melakukan peramalan. Pada dasarnya, peramalan memperkirakan suatu keadaan dimasa yang akan datang berdasarkan keadaan masa lalu dan sekarang yang diperlukan untuk menetapkan kapan suatu peristiwa akan terjadi, sehingga tindakan yang tepat dapat dilakukan. Contohnya adalah permintaan terhadap suatu produk pada periode waktu yang akan datang. Pada hakikatnya, sebuah peramalan hanya merupakan suatu perkiraan terhadap suatu objek. Akan tetapi, dengan menggunakan teknik peramalan yang tepat maka peramalan akan menjadi lebih dari sekedar perkiraan. Pada umumnya, hasil peramalan yang baik dapat dilihat dari kecilnya nilai kesalahan meramal atau *forecast error* yang dapat diukur dengan menggunakan *Mean Absolute Deviation*, *Mean Square Error*, dan *Mean Absolute Percentage Error* (Iswahyudi,2016).

Dua hal pokok yang harus diperhatikan dalam proses peramalan yang akurat dan bermanfaat :

- a. Pengumpulan data yang relevan berupa informasi yang dapat menghasilkan peramalan yang akurat.
- b. Pemilihan teknik peramalan yang tepat yang akan memanfaatkan informasi data yang diperoleh semaksimal mungkin.

Apabila dilihat sifat ramalan yang telah disusun, maka peramalan dapat dibedakan atas 2 macam, yaitu:

- a. Peramalan Kualitatif

Pada peramalan kualitatif, hasil peramalan sangat tergantung pada analis atau peramal yang menyusunnya. Hal ini dikarenakan hasil peramalan berdasarkan pemikiran yang bersifat *judgement* atau pendapat.

- b. Peramalan Kuantitatif

Pada peramalan kuantitatif, hasil peramalan yang dibuat sangat tergantung pada metode peramalan yang digunakan. Penggunaan metode berbeda akan menghasilkan hasil yang berbeda walaupun data yang



digunakan sama. Setiap metode yang digunakan harus mempertimbangkan tingkat keakuratan hasil ramalan sehingga tidak terjadi penyimpangan yang terlalu signifikan antara hasil ramalan dengan kenyataan yang terjadi di lapangan.

Metode peramalan kuantitatif dibedakan menjadi dua bagian, yaitu sebagai berikut:

- a). Metode peramalan yang didasarkan pada penggunaan pola hubungan antara variabel yang diperkirakan dengan variabel waktu yang merupakan deret waktu atau *Time Series*. Contohnya metode *smoothing*, regresi, dekomposisi
- b). Metode peramalan yang berdasarkan pada penggunaan analisa pola hubungan antara variabel yang diperkirakan dengan variabel lain yang mempengaruhinya, yang bukan waktu yang disebut Causal. Contoh metode regresi, korelasi, ekonometrik, *input-output*

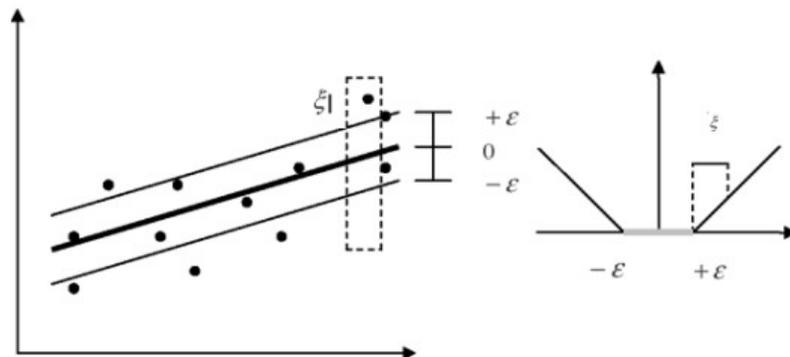
### 2.13 Support Vector Regression

*Support Vector Regression* (SVR) adalah metode pengembangan dari *Support Vector Machine* (SVM) yang digunakan pada permasalahan regresi. SVR termasuk dalam algoritma *supervised learning* yang digunakan untuk memprediksi nilai variabel kontinu. Perbedaan SVM dan SVR adalah SVM digunakan untuk mencari *hyperplane* (fungsi pemisah) yang terbaik di antara 2 objek yang tidak terbatas jumlahnya dengan cara memaksimalkan jarak (margin) antara dua objek yang berbeda sedangkan SVR digunakan untuk menemukan suatu fungsi yang memiliki deviasi paling besar  $\epsilon$  dari target aktual  $y_i$ . Menurut Smola & Scholkopt pada tahun 2004, SVR bertujuan untuk menemukan fungsi  $f(x)$  sebagai suatu *hyperplane* (garis pemisah) berupa fungsi regresi, pada semua input data yang memiliki deviasi paling besar  $\epsilon$  dari target aktual  $y_i$  untuk semua data training dan membuat galat sekecil mungkin

Keunggulan SVR adalah kemampuan untuk mengatasi masalah data dengan trik kernel. Dalam SVR terdapat parameter *epsilon* ( $\epsilon$ ), *cost* ( $C$ ), *gamma* ( $\gamma$ ) yang ditentukan nilai optimalnya menggunakan metode *grid search*.



*Grid search* merupakan kombinasi parameter yang diujikan pada suatu model SVR untuk mencari nilai error dalam klasifikasi.



Gambar 3 Hyperplane dan margin pada SVR

Sumber: Smola & Scholkopt, 2004

Pada Gambar 3 garis hitam tebal merupakan *hyperplane* sedangkan untuk dua garis yang mengapitnya adalah *soft margin*. Jarak antara *hyperplane* dan *soft margin* adalah sebesar  $\varepsilon$  dan titik-titik yang berada pada  $+\varepsilon$  sampai  $-\varepsilon$  merupakan support vector, namun untuk titik yang melewati soft margin dibutuhkan adanya variabel slack  $\xi$ .

Ide dasar penggunaan metode SVR adalah misalkan terdapat  $n$  set data *training*  $x_i y_i$ , dengan  $x_i \in R^d$  adalah vektor input dari data ke- $i$  dimana  $i=1,2,\dots,n$ , dan  $d$  adalah dimensi dan  $y_i$  adalah nilai target. Persamaan fungsi regresi secara umum dapat ditulis persamaan (1) berikut (Smola & Scholkopt, 2004):

$$f(x) = w^T \varphi(x) + b \quad (1)$$

Keterangan :

$\varphi(x)$  = titik di dalam feature space  $F$ , hasil dari pemetaan  $x$  di dalam input space

$f(x)$  = fungsi regresi

$w$  = vektor bobot, yang mempunyai dimensi 1

$b$  = bias

$x$  = input atau vektor input

Sedangkan koefisien  $w$  dan  $b$  mempunyai fungsi untuk meminimalkan fungsi perti pada persamaan (2) di bawah ini:

$$R = \min \frac{1}{2} \|w\|^2 + C \frac{1}{l} \left( \sum_1^l L_\varepsilon(y_i, f(x_i)) \right) \quad (2)$$



Keterangan :

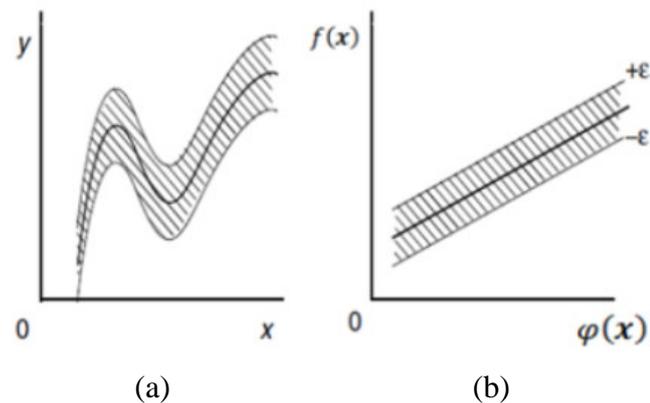
$R$  = Risk Function (Fungsi Resiko)

$L_\varepsilon$  = Loss Function yang bertipe  $\varepsilon$ -insentive loss function

$\|w\|$  = normalisasi  $w$

$\varepsilon$  = epsilon yaitu deviasi atau derajat toleransi terhadap error

$C$  = nilai penalti jika deviasi lebih besar dari batas error



Gambar 4 *Insensitive zone* (a) *original input space*, dan (b) *feature space*

Sumber: Yasin dkk., 2016

Pada gambar 4, gambar *Insensitive zone* (a) *original input space*, dan (b) *feature space*  $D(x, y) = \pm\varepsilon$  adalah jarak terjauh *support vector* dari *hyperplane*, kemudian disebut margin. Memaksimalkan margin akan meningkatkan probabilitas data ke dalam radius  $\pm\varepsilon$ . Jarak dari *hyperplane*  $D(x, y)$  ke data  $(x, y)$  adalah  $|D(x, y)|/\|w^*\|$ , dimana:

$$w^* = (1 - w^T)^T \quad (3)$$

Diasumsikan bahwa jarak maksimum data terhadap *hyperplane* adalah  $\delta$ , maka estimasi yang ideal akan terpenuhi dengan:

$$\begin{aligned} \frac{|D(x, y)|}{\|w^*\|} &\leq \delta \\ |D(x, y)| &\leq \delta \|w^*\| \\ \delta \|w^*\| &= \varepsilon \end{aligned} \quad (4)$$



h karena itu untuk memaksimalkan margin  $\delta$ , diperlukan  $\|w^*\|$  yang  
t. Optimasi penyelesaian masalah dengan bentuk Quadratic Programming:

$$\min \frac{1}{2} \|w\|^2 \quad (5)$$

dengan syarat

$$y_i - w^T \varphi(x_i) - b \leq \varepsilon \text{ untuk } i = 1, \dots, l$$

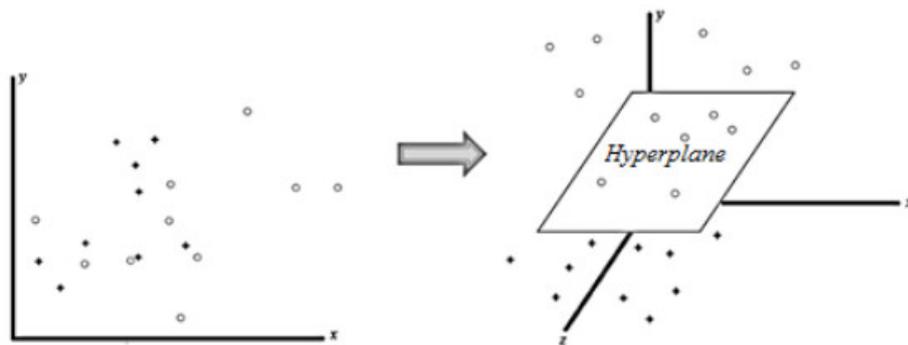
$$w^T \varphi(x_i) - y_i + b \leq \varepsilon \text{ untuk } i = 1, \dots, l$$

Faktor  $\|w\|^2$  dinamakan regulasi. Meminimalkan  $\|w\|^2$  akan membuat suatu fungsi setipis (flat) mungkin, sehingga bisa mengontrol kapasitas fungsi (function capacity).

Diasumsikan bahwa semua titik ada dalam rentang  $f(x) \pm \varepsilon$  (feasible), dalam hal ketidaklayakan (*infesibility*), dimana ada beberapa titik yang mungkin keluar dari rentang  $f(x) \pm \varepsilon$  maka ditambahkan variabel slack  $\xi$  dan  $\xi^*$  untuk mengatasi masalah pembatasan yang tidak layak (infeasible constraints) dalam problem optimasi (Santosa, 2007).

### 2.13.1 Fungsi Kernel

Menurut Furi, Jondri, & Saepudin pada tahun 2015 dalam dunia nyata permasalahan jarang yang bersifat linear dan kebanyakan bersifat non linear. Untuk menyelesaikan permasalahan non linear ini digunakan fungsi kernel. Untuk memecahkan masalah linear dalam ruang berdimensi tinggi, yang harus dilakukan adalah mengganti *inner product* ( $x_i$  dan  $x_j$ ) dengan fungsi kernel yang sesuai.



Gambar 5 Ilustrasi *hyperplane* pada dimensi yang lebih tinggi

Sumber: Athoillah, 2015

Ilustrasi dari konsep kernel pada Gambar 5 di atas yaitu, gambar pada sisi kiri menunjukkan data dari dua kelas berbeda pada ruang input berdimensi dua yang dipisahkan secara linear. Sedangkan pada gambar sebelah kanan terlihat *feature space* mampu memetakan setiap data pada ruang input ke ruang fitur



berdimensi lebih tinggi, sehingga kedua kelas data tersebut bisa dipisahkan secara linear oleh *hyperplane*.

Untuk membantu mengatasi permasalahan non linear pada dimensi tinggi yang harus dilakukan yaitu mengganti *inner product* ( $x_i$  dan  $x_j$ ) dengan fungsi kernel. Karena kinerja dari metode SVR ditentukan oleh jenis fungsi kernel dan parameter yang digunakan. Adapun menurut Bhavsar & Mahesh pada tahun 2012, fungsi-fungsi kernel yang sering digunakan pada metode SVR adalah sebagai berikut.

Tabel 2 Rumus dari setiap tipe kernel pada SVR

No.	Tipe Kernel	Formula
1.	Linear	$K(x_i, x_j) = (x_i^T x_j)$
2.	Polynomial	$K(x_i, x_j) = (x_i, x_j + 1)^P$
3.	Radial Basis Function	$K(x_i, x_j) = \exp(-\gamma(x_i, x_j)^2)$

Sumber: Bhavsar & Mahesh, 2012

dengan:

$x_i, x_j$  = Vektor dari dua dataset

$P$  = Derajat polinomial

$\gamma$  = Gamma

Pemilihan fungsi kernel yang tepat merupakan hal yang penting untuk menentukan *feature space*. Kernel linear merupakan fungsi kernel paling sederhana. Kernel linear biasanya digunakan pada data set yang datanya sudah terpisah secara linear. Kernel polinomial adalah sebuah persamaan yang terdiri dari variabel dan koefisien yang memiliki suku banyak dan sering digunakan jika data training dinormalisasi. Sedangkan kernel radial atau Gaussian merupakan kernel yang digunakan untuk menyelesaikan masalah pada data yang tidak terpisah secara linear.

### 2.13.2 Grid Search Optimization



masalah yang sering terjadi ketika menggunakan metode SVR adalah menentukan parameter model yang optimal. Salah satu cara optimasi yang

dapat digunakan untuk menentukan parameter terbaik pada metode SVR adalah optimasi *Grid Search* atau *Grid Search Optimization*. *Grid Search Optimization* ini akan membagi jangkauan parameter yang akan dioptimalkan ke dalam grid dan melintasi semua titik untuk mendapatkan parameter yang optimal (Yasin dkk, 2014). Dalam aplikasinya, *Grid Search* dipadukan dengan matriks kinerja dan diukur menggunakan *cross validation* pada data training. *Grid Search Optimization* akan melatih pasangan-pasangan parameter dan akan menentukan/memilih pasangan parameter yang optimal dengan akurasi terbaik atau rata-rata galat yang terkecil dari uji *cross validation Grid Search*.

Menurut Leidiyana pada tahun 2013, *cross validation* adalah pengujian standar yang dilakukan untuk memprediksi *error rate*. Data *training* dibagi secara *random* ke dalam beberapa bagian dengan perbandingan yang sama kemudian *error rate* dihitung bagian demi bagian, selanjutnya dihitung rata-rata seluruh *error rate* untuk mendapatkan *error rate* keseluruhan. Dalam *cross validation*, dikenal validasi *leave-one-out* (LOO). Dalam LOO, data dibagi kedalam 2 subset, subset 1 berisi N-1 data training dan satu data sisanya untuk testing (Santosa, 2007).

Salah satu metode *cross validation* yang umum digunakan adalah *k-fold validation*. Menurut Jiawei dkk pada tahun 2011, prosedur dari metode *cross validation* adalah sebagai berikut:

1. Membagi data menjadi k bagian dengan ukuran yang sama.
2. K-1 bagian dijadikan data *training* dan satu bagian dijadikan data testing.
3. Proses ini dilakukan sebanyak k pengulangan pada setiap kombinasi data *training* dan data testing. Nilai akurasi dari setiap iterasi dirata-ratakan untuk mendapatkan estimasi nilai akurasi akhir.

## 2.14 Artificial Neural Network

*Artificial Neural Network* (ANN) merupakan sebuah sistem cerdas yang digunakan untuk mengolah informasi yang merupakan perkembangan dari generalisasi model matematika. Prinsip kerja ANN terinspirasi dari prinsip kerja jaringan saraf (*neural network*) manusia. Para ilmuwan menciptakan algoritma yang bekerja menyerupai pola kerja saraf (neuron) tersebut, maka



digunakanlah nama *Artificial Neural Network*, atau dalam Bahasa dengan biasa disebut Jaringan Saraf Tiruan (JST).

Meniru cara kerja yang sama dengan otak, input atau masukan diumpamakan sebagai neuron, kemudian masukan-masukan tersebut dikalikan dengan suatu nilai (bobot) dan diolah dengan fungsi tertentu (fungsi aktivasi sehingga menghasilkan suatu keluaran (Stergiou dan Siganos, 2006). Secara kausal, ANN bekerja dengan mempresentasikan dinamika suatu sistem ke dalam suatu model matematika dengan cara menentukan arsitektur jaringan, metode pembelajaran jaringan, dan akurasi model jaringan. Dengan interpretasi tersebut akan diketahui pola perilaku sistem yang dapat digunakan untuk meramalkan keluaran tertentu pada periode ke depan.

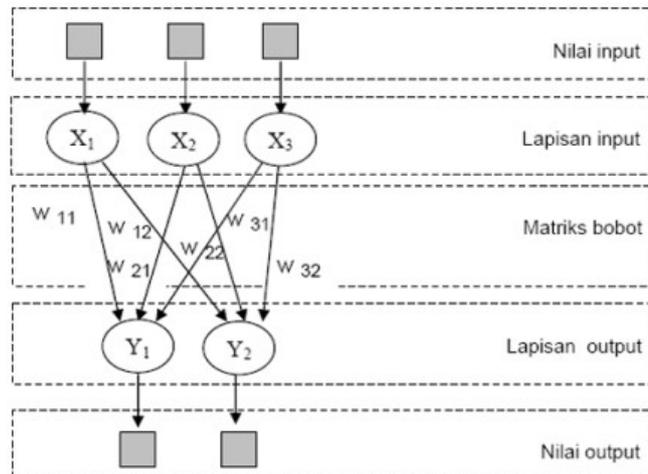
### 2.14.1 Arsitektur Artificial Neural Network

Pada ANN, neuron diasumsikan dapat dikelompokkan dalam layer seperti yang telah dijelaskan sebelumnya. Umumnya, terdapat 3 struktur ANN yaitu *single layer network* dan *multilayer network* (berdasarkan banyak lapisan) serta *competitive layer network*.

#### a. Jaringan dengan lapisan tunggal (*single layer network*)

Jaringan lapis tunggal terdiri dari satu layer dari bobot yang saling terhubung. Jaringan ini hanya menerima *input* atau informasi yang masuk kemudian secara langsung akan mengolahnya menjadi sinyal keluaran (*output*) tanpa harus melalui lapisan tersembunyi (*hidden layer*). Pada Gambar 6, terlihat neuron-neuron pada kedua lapisan saling berhubungan. Seberapa besar hubungan antara 2 neuron tersebut ditentukan oleh bobot yang bersesuaian. Semua unit *input* akan dihubungkan dengan setiap unit *output*.



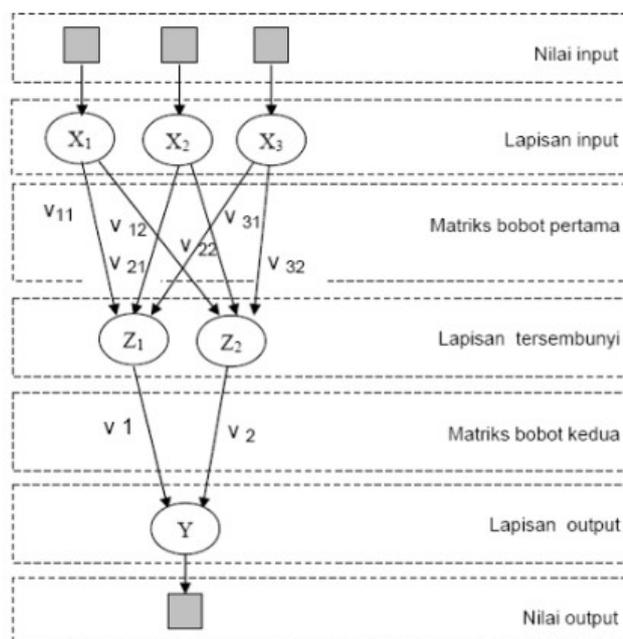


Gambar 6 Jaringan Saraf dengan lapisan tunggal

Sumber: Wuryandari & Afrianto, 2012

b. Jaringan dengan banyak lapisan (*multilayer network*)

ANN dengan tipe ini memiliki lebih dari satu layer yang diberi nama *hidden layer*. Gambar 7 menunjukkan jaringan dengan *multilayer*, dimana semua layer yang terletak di antara *input layer* dan *output layer* merupakan *hidden layer*, karena jumlah *hidden layer* yang bisa lebih dari satu layer. ANN lapis jamak ini menggunakan fungsi aktivasi nonlinear yang dapat digunakan untuk menyelesaikan berbagai tipe permasalahan yang lebih rumit dan kompleks daripada lapisan tunggal

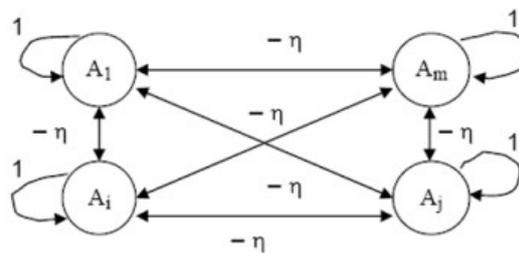


### Gambar 7 Jaringan Saraf Tiruan lapisan banyak

Sumber: Wuryandari & Afrianto, 2012

#### c. Jaringan dengan lapisan kompetitif (*competitive layer network*)

Pada jaringan ini sekumpulan neuron bersaing untuk mendapatkan hak menjadi aktif. Umumnya hubungan antar neuron pada lapisan kompetitif ini tidak diperlihatkan pada diagram arsitektur. Gambar 8 berikut menunjukkan salah satu contoh arsitektur jaringan dengan lapisan kompetitif yang memiliki bobot  $-\eta$ .



Gambar 8 Jaringan Saraf dengan lapisan kompetitif yang memiliki  $-\eta$

Sumber: Wuryandari & Afrianto, 2012

#### 2.14.2 Fungsi Pelatihan

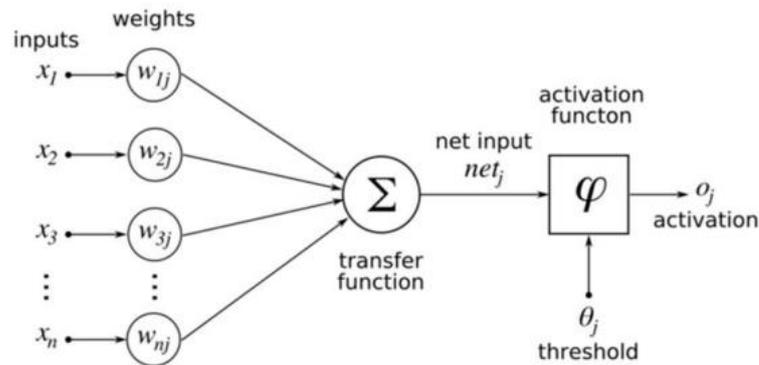
Fokus utama dalam peramalan menggunakan metode ANN adalah untuk melatih *network* untuk dapat memahami *pattern* nilai yang dimiliki dari sebuah kasus berdasarkan data historis untuk dapat memprediksi nilai yang akan datang. Fungsi pelatihan berfungsi sebagai algoritma yang digunakan pada ANN untuk melatih *network* dalam memahami *pattern* tersebut. Terdapat banyak fungsi pelatihan yang tersedia dan dapat digunakan, hal ini menjadikan sebagian orang bingung untuk menentukan fungsi pelatihan apa yang paling baik untuk digunakan. Namun berdasarkan uji coba yang dilakukan MATLAB didapatkan dua fungsi pelatihan yang dipilih dalam penelitian ini. Kedua fungsi tersebut adalah Lavenberg-Marquardt (LM), BFGS Quasi-Newton (BFG). Kedua fungsi pelatihan diatas telah diuji dan dibuktikan lebih baik dari fungsi lainnya oleh MATLAB

memecahkan kasus dengan tipe permasalahan baik *function approximation* maupun *pattern recognition*.



### 2.14.3 Fungsi Aktivasi

Fungsi aktivasi merupakan fungsi yang digunakan untuk mengolah inputan informasi. Sebagai gambaran, fungsi aktivasi dapat dilihat pada Gambar 9 di bawah ini.



Gambar 9 Struktur ANN

Sumber: Wuryandari & Afrianto, 2012

Dapat dilihat pada struktur ANN pada Gambar 9 di atas, fungsi transfer berfungsi mengubah sejumlah inputan ( $x_1, x_2, x_3, \dots, x_n$ ) yang memiliki bobot ( $W_{1j}, W_{2j}, \dots, W_{nj}$ ) dan bias menjadi nilai input ( $net_j$ ) bagi fungsi aktivasi. Selanjutnya fungsi aktivasi memproses nilai input untuk dibandingkan dengan threshold yang ditentukan dan mengaktivasi nilai menjadi output ( $o_j$ ). Beberapa fungsi aktivasi yang digunakan pada ANN adalah:

Selain Fungsi Tanda, terdapat beberapa fungsi lain yang praktis untuk digunakan oleh neuron, yaitu:

a. *Step Function*

Fungsi aktivasi *Step* dan *Sign* disebut juga *hard limit function*, sering digunakan oleh neuron untuk tugas-tugas pengambilan keputusan (*decision making*) dalam pengklasifikasian dan pengenalan pola.

b. *Linear Function*

Fungsi aktivasi *Linear* menyediakan *output* yang setara dengan *input* neuron yang diberikan bobot. Biasanya digunakan untuk aproksimasi linear.

$$f(x) = x \quad (6)$$

*sigmoid Function*



Fungsi *Sigmoid* mentransformasi *input*, yang bisa memiliki nilai di antara positif dan negatif tak terhingga, menjadi nilai yang dapat dinalar diantara 0 dan 1, biasanya untuk jaringan *back-propagation*.

1. *Binary Sigmoid Function*

$$f(x) = \frac{1}{1+e^{(-kx)}} \quad (7)$$

2. *Bipolar Sigmoid Function*

$$f(x) = \frac{1-e^{(-kx)}}{1+e^{(-kx)}} \quad (8)$$

d. *Fungsi Rectified Linear Unit (ReLU)*

*Rectified Linear Unit (ReLU) function* memiliki kelebihan dalam Network yang diinisiasi secara *random*, hanya 50% dari *hidden layer* yang akan di aktivasi. ReLU Function dirumuskan sebagai berikut.

$$f(x) = \max(0, x) \quad (9)$$

e. *Softmax*

*Softmax Funtion* atau disebut juga *softmax regression* adalah bentuk *logistic regression* yang *input value*-nya di normalisasi ke dalam *probability distribution*, dirumuskan sebagai,

$$f_i(\vec{x}) = \frac{e^{(x_i)}}{\sum_{j=1}^J e^{(x_j)}} \text{ untuk } i = 1, \dots, J \quad (10)$$

## 2.15 RMSE

Setiap hasil dari proses prediksi (biasanya untuk istilah perkiraan jangka pendek) atau peramalan (biasanya untuk istilah perkiraan jangka panjang) diperlukan proses evaluasi untuk mengetahui performa dari model peramalan yang diterapkan (Cholissodin dkk., 2020).

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (P_i - A_1)^2} \quad (11)$$

Ket:



nilai *root mean square error*  
 aktual  
 hasil prediksi

$n$  = jumlah data

## 2.16 Penelitian Terkait

Berikut beberapa penelitian terkait terhadap penelitian yang dilakukan:

### 2.16.1 Estimating the Determinant Factors of Container Dwell Seaports

Pada penelitian Nadereh dkk tahun 2012 ini, dilakukan identifikasi faktor penentu CDT (*container dwelling time*) dan menggambarkan alat komputasi yang sesuai untuk memperkirakan CDT berdasarkan serangkaian faktor tertentu dimana operator terminal biasanya mengumpulkan data yang relevan. Artikel ini membandingkan kinerja tiga algoritma *data mining* untuk memperkirakan CDT yaitu Naive Bayes, decision tree, dan NB-decision tree hybrid. Dengan menggunakan model dengan kinerja terbaik, data terminal sampel digunakan untuk mengukur bagaimana perubahan faktor penentu CDT berdampak pada CDT, kapasitas pekarangan, dan pendapatan terminal. Hasilnya mengungkapkan bahwa dampak dari perubahan faktor penentu CDT dapat cukup besar untuk mempengaruhi kapasitas terminal dan pendapatan yang diperoleh dari biaya kelebihan waktu berlabuh. Algoritma yang berkinerja terbaik adalah *decision tree*, yang kemudian hasilnya digunakan untuk mengukur dampak perubahan penentu CDT pada CDT. Meskipun tidak ada kesimpulan umum yang dapat diturunkan mengenai pentingnya faktor apa pun dan dampaknya terhadap CDT, pendekatan generik yang dikembangkan dapat digunakan dalam kombinasi dengan data dari terminal tertentu untuk membantu menemukan cara potensial untuk mengelola CDT secara efektif dan menentukan dampak yang diantisipasi pada terminal. Kapasitas pekarangan dan pendapatan. Penelitian ini menyediakan alat yang berguna bagi anggota komunitas pelabuhan, perdagangan, dan transportasi untuk mengevaluasi kebijakan yang tepat guna meningkatkan pengoperasian fasilitas yang merupakan mata rantai penting dalam rantai pasokan dan sangat penting bagi ekonomi global. Penelitian di masa depan dapat dibangun berdasarkan pekerjaan

memprediksi CDT menggunakan kumpulan data yang lebih komprehensif tentang faktor penentu CDT, seperti informasi pengirim dan penerima



### 2.16.2 Development of Models Predicting Dwell Time of Import Containers in Port Container Terminals – An Artificial Neural Networks Application

Pada penelitian Ioanna dkk tahun 2015 ini, tujuan umumnya adalah untuk mengusulkan pengembangan kerangka metodologis yang menggabungkan berbagai faktor yang mempengaruhi *Dwell Time (DT)* petikemas di terminal petikemas. Terminal petikemas dianggap sebagai elemen kunci dari rantai logistik karena merupakan penghubung antara moda transportasi laut dan pedalaman. Peramalan beban kerja sangat penting dalam hal kedatangan truk untuk menghindari kemacetan dan kelancaran integrasi terminal petikemas dalam rantai pasokan. Operator terminal, cenderung membuat keputusan penumpukan berdasarkan sebagian besar faktor seperti berat, ukuran, dan jenis petikemas. Metodologi yang disarankan memerlukan pengumpulan data agregat dan penerapan *Artificial Neural Networks (ANN)* untuk mengidentifikasi faktor penentu *Dwell Time (DT)*. Hasil pertama dari ANN menunjukkan bahwa faktor terpenting yang mempengaruhi akurasi model secara signifikan adalah sebagai berikut: ukuran dan jenis petikemas, hari dan bulan pembongkaran petikemas, pelabuhan asal kapal dan komoditas yang diangkut. Dari perbandingan model Jaringan Saraf Tiruan yang berbeda, hasilnya dapat disimpulkan bahwa semakin banyak informasi yang tersedia semakin baik akurasi model. Oleh karena itu, operator terminal disarankan untuk mengumpulkan sebanyak mungkin informasi tentang setiap petikemas yang mereka tangani agar dapat digunakan untuk pengembangan model prediktif.

### 2.16.3 Analisis Prediksi Dwell Time Of Import Containers Menggunakan Artificial Neural Network Pada Terminal Petikemas Semarang

Penelitian Sakinah pada tahun 2019 ini bertujuan untuk mengidentifikasi dan menjelaskan faktor faktor berpengaruh terhadap *dwell time* dan cara menangani *dwell time* yang dipengaruhi *month of discharge*, *day of discharge*, *container's size*, *container's type*, *yard occupancy* dengan di Terminal Petikemas Semarang. Metode yang digunakan dalam penelitian ini adalah metode *artificial neural network (ANN)*. Metode ini mengadopsi prinsip kerja otak manusia dalam memecahkan dan mampu menghasilkan hasil prediksi yang akurat melalui iannya dalam meminimalkan error pada model. Menurut Sarakolaei dkk. in 2012, ANN dapat memberikan hasil lebih baik, saat data dan variabel



independen dan dependen tidak ada hubungan matematis. Hasil dari penelitian ini menunjukkan bahwa faktor yang paling penting mempengaruhi secara signifikan akurasi model adalah sebagai berikut: *day of discharge, container's type, month of discharge, container's size, yard occupancy*.

#### **2.16.4 Prediksi Waktu Sandar Kapal di Pelabuhan Batu Ampar, Kota Batam, Provinsi Kepulauan Riau**

Penelitian Candra dkk. tahun 2019 ini melakukan prediksi waktu sandar kapal (*dwelling time*) di pelabuhan Batu Ampar dengan bantuan *data mining*. Pelabuhan Batu Ampar merupakan pelabuhan barang terbesar di Kota Batam yang memiliki lalu lintas terbaik untuk kegiatan ekspor dan kegiatan penting. Waktu tunggu masih menjadi masalah dalam layanan pelabuhan. Waktu tunggu merupakan salah satu indikator efisiensi pengelolaan pelabuhan. Rata-rata waktu tunggu pelabuhan Batu Ampar untuk kegiatan bongkar pada Triwulan I tahun 2015 adalah 7 hari, sedangkan kegiatan bongkar muat adalah 5 hari. Hal inilah yang membuat pelabuhan Batu Ampar masih banyak dikeluhkan sehingga mengakibatkan banyaknya antrian kapal. Untuk itu diperlukan analisis untuk menghasilkan model yang dapat memberikan gambaran waktu tunggu di pelabuhan dan mengevaluasi model analitik yang dibangun. Analisis data sekunder pelabuhan Batu Ampar menggunakan *data mining*. Metode *data mining* dilakukan dengan menggunakan metode pembelajaran terbimbing yaitu regresi berganda dan pohon keputusan. Tujuan umum dari regresi berganda adalah untuk lebih jauh tentang hubungan antara variabel independen atau prediktor dan variabel atau kriteria dependen. Pohon keputusan yang digunakan untuk mengekstrak data port menggunakan klasifikasi ini. Klasifikasi pohon keputusan dapat menemukan data yang mengandung kelas objek yang terkumpul dengan baik, sehingga kelas dapat diinterpretasikan secara lengkap dalam konteks teori substantif. Dua metode evaluasi model dilakukan untuk dua hasil pemodelan yang dibangun. Uji Analisis Varians (ANOVA) digunakan untuk mengevaluasi model regresi berganda, sedangkan untuk model pohon keputusan dievaluasi dengan matriks konfusi. Hasil data menunjukkan lama kapal bongkar/muat dengan tiga variabel jenis , bendera, dan volume. Dengan menggunakan regresi berganda, dihasilkan prediksi waktu berlabuh kapal. Hasil evaluasi, model dibuat signifikan.



Dengan tingkat kepercayaan 95% model prediksi yang dibuat akan mewakili nilai sebenarnya. Untuk pohon keputusan, model evaluasi yang dibuat sudah sesuai dengan presisi sebesar 84,50%.



## BAB 3 METODE PENELITIAN/PERANCANGAN

### 3.1 Tahapan Penelitian

Pada penelitian ini mengusulkan pembuatan sistem prediksi *dwelling time*. Penelitian ini akan membangun model menggunakan 2 metode, yaitu *Support Vector Regression (SVR)* dan *Artificial Neural Network (ANN)*. Set data pada penelitian ini menggunakan penggabungan data *Inbound List* dan *Terminal In & Out*. Data *Inbound List* antara lain Pelabuhan Asal, Status Kontainer, Jenis Kontainer, dan *Gross Kontainer*. Data *Terminal In & Out* yaitu Ukuran Kontainer dan *Dwelling Time*. Data penelitian diambil pada dari bulan November 2022 hingga bulan Mei 2023, berfokus pada 3 kapal kontainer setiap bulannya yang digunakan untuk membangun model. Adapun disisihkan data bulan Juni 2023 digunakan untuk testing model. Selanjutnya, dibandingkan hasil prediksi model dari kedua metode untuk didapatkan hasil yang optimal.

### 3.2 Waktu dan Lokasi Penelitian

Penelitian ini dilakukan sejak disetujuinya proposal penelitian ini yaitu pada bulan Desember 2021 hingga proses pelaporan hasil penelitian ini pada bulan Desember 2023. Penelitian ini dilakukan di Laboratorium *Artificial Intelligence and Multimedia Processing (AIMP)* Departemen Teknik Informatika, Fakultas Teknik Universitas Hasanuddin yang terletak di Bontomarannu, Kabupaten Gowa, Sulawesi Selatan.

### 3.3 Instrumen Penelitian

Instrumen penelitian yang digunakan pada penelitian ini meliputi:

a. *Software*:

- Windows 10x64
- PyCharm professional
- Google Colab
- Google Chrome Version 105.0.5195.127 (Official Build) (64-bit)



### 3.4 Teknik Pengambilan Data

Data yang digunakan pada penelitian ini merupakan data primer yaitu data yang diperoleh langsung dari sumber daya yang berhubungan dengan penelitian. Data tersebut adalah data informasi detail kontainer Pelabuhan Soekarno Hatta Makassar. Data ini diperoleh secara langsung dari Terminal Petikemas Makassar sudah dalam bentuk digital (.xlsx) namun terpisah dalam 2 source data, yaitu data *Inbound List* dan data *Terminal In & Out*. Sehingga harus dilakukan penggabungan terlebih dahulu pada tahap *pre-processing*. Keseluruhan total data setelah digabungkan yaitu sebanyak 8.665 data.

Data-data yang diperoleh dari Terminal Petikemas Makassar antara lain detail kapal yang mengangkut serta detail kontainer seperti pelabuhan asal, status kontainer, jenis kontainer, ukuran kontainer, komoditas, *gross* dan *dwelling time*.

### 3.5 Perancangan Sistem

Sistem ini dibangun menggunakan 2 metode, yaitu *forecasting* dengan algoritma SVR (*Support Vector Regression*) dan ANN (*Artificial Neural Network*) untuk meramalkan besaran *dwelling time* dari *input* tertentu.

Sistem yang dirancang awalnya akan melakukan seleksi data untuk menentukan parameter-parameter yang berpotensi menjadi parameter prediksi *dwelling time*. Kemudian data yang telah diseleksi akan melalui tahapan *preprocessing* terlebih dahulu agar data siap diolah untuk dipakai dalam sistem. Kemudian setelah data siap diolah maka akan dilakukan pembangunan sistem prediksi dengan peramalan menggunakan metode *forecasting*. Dan diakhiri dengan melakukan evaluasi dari hasil prediksi data *dwelling time* yang ada.

Untuk penelitian ini, digunakan data dari bulan November 2022 hingga bulan Mei 2023 untuk membangun model prediksi dan data bulan Juni 2023 untuk mengevaluasi model prediksi.

#### 3.5.1 Input Data



Langkah pertama adalah memasukkan dataset ke dalam sistem. Namun, sebelumnya, dilakukan perubahan untuk seluruh dataset ke dalam format data *Separated Values File* (.csv). Atribut yang bersumber dari Terminal

Petikemas Makassar terbilang lengkap namun tidak semua yang akan dipakai dalam penelitian ini. Karena sumber data yang dimiliki memiliki *file* yang terpisah, maka terlebih dahulu dilakukan penggabungan kedua sumber data yang akan dipakai untuk selanjutnya diseleksi lebih lanjut. Gambar 10 dan 11 menunjukkan data *Inbound List* dan *Terminal In & Out*.

No.	Stowage	Cargo	Op. Type	DS	Plan	POW	Position	Class	Out	Out Carrier	Thau	Direct	OPR	POB	POL	POD	NPOD	FPOD	FDEST	B/I	F/M	Comm.	IS
1	0170502	BEAU2717978	Container			GOT		Import	G	0556			TNT		IDJKT	IDMAK				031/TTTRUKT/208/22	FULL	GE	23
2	0210502	BEAU282022	Container			GOT		Import	G	1444			TNT		IDJKT	IDMAK				055/TTTRUKT/208/22	FULL	GE	23
3	0220482	BEAU5068814	Container			GOT		Import	G	1223			TNT		IDJKT	IDMAK				009/TTTRUKT/208/22	FULL	GE	48
4	0100384	BMOU5455046	Container			GOT		Import	G	0556			TNT		IDJKT	IDMAK				024/TTTRUKT/208/22	FULL	GE	48
5	0140782	BMOU6014430	Container			GOT		Import	G	0186			TNT		IDJKT	IDMAK				079/TTTRUKT/208/22	FULL	GE	48
6	0200065	CAU5220273	Container			GOT		Import	G	1715			TNT		IDJKT	IDMAK				051/TTTRUKT/208/22	FULL	RF	23
7	0150082	CICU8585990	Container			GOT		Import	G	0087			TNT		IDJKT	IDMAK				031/TTTRUKT/208/22	FULL	GE	23
8	0070108	CICU8587544	Container			GOT		Import	G	0934			TNT		IDJKT	IDMAK				072/TTTRUKT/208/22	FULL	GE	22
9	0210208	CICU8587950	Container			GOT		Import	G	1575			TNT		IDJKT	IDMAK				017/TTTRUKT/208/22	FULL	GE	22
10	0130082	CICU8589742	Container			GOT		Import	G	0272			TNT		IDJKT	IDMAK				035/TTTRUKT/208/22	FULL	GE	22
11	0130054	CICU8589204	Container			GOT		Import	G	0372			TNT		IDJKT	IDMAK				035/TTTRUKT/208/22	FULL	GE	22
12	0230482	CICU8589280	Container			GOT		Import	G	0259			TNT		IDJKT	IDMAK				041/TTTRUKT/208/22	FULL	GE	22
13	0130008	CICU8584748	Container			GOT		Import	G	0147			TNT		IDJKT	IDMAK				085/TTTRUKT/208/22	FULL	GE	22
14	0010008	CICU8585766	Container			GOT		Import	G	0309			TNT		IDJKT	IDMAK				088/TTTRUKT/208/22	FULL	GE	22
15	0110208	CICU8589130	Container			GOT		Import	G	0258			TNT		IDJKT	IDMAK				040/TTTRUKT/208/22	FULL	GE	22
16	0110808	CICU8589289	Container			GOT		Import	G	1387			TNT		IDJKT	IDMAK				073/TTTRUKT/208/22	FULL	GE	22
17	0050308	CICU8589280	Container			GOT		Import	G	0934			TNT		IDJKT	IDMAK				073/TTTRUKT/208/22	FULL	GE	22
18	0250304	CICU8587258	Container			GOT		Import	G	0251			TNT		IDJKT	IDMAK				040/TTTRUKT/208/22	FULL	GE	22
19	0130404	CICU8589282	Container			GOT		Import	G	0285			TNT		IDJKT	IDMAK				042/TTTRUKT/208/22	FULL	GE	22
20	0110308	CICU8589904	Container			GOT		Import	G	0813			TNT		IDJKT	IDMAK				071/TTTRUKT/208/22	FULL	GE	22
21	0150304	CICU8600389	Container			GOT		Import	G	1393			TNT		IDJKT	IDMAK				078/TTTRUKT/208/22	FULL	GE	22
22	0070304	CICU8600749	Container			GOT		Import	G	0838			TNT		IDJKT	IDMAK				016/TTTRUKT/208/22	FULL	GE	22

Gambar 10 Sampel data *Inbound List*

No.	Move Type	Cargo	Class	POW	Position	In TRSP			Out TRSP			F/M	Comm.	ISO	Length	OPR	POD	NPOD	FDEST	D/O	S/O	
						Type	In Carrier ID	In Carrier Name	In Carrier Voyage	Type	Out Carrier ID											Out Carrier Name
1	DS	BEAU2717978	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0556			FULL	GE	2201	20	TNT	IDMAK			
2	DS	BEAU282022	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	1444			FULL	GE	2201	20	TNT	IDMAK			
3	DS	BEAU5068814	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	1223			FULL	GE	4501	40	TNT	IDMAK			
4	DS	BMOU5455046	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0556			FULL	GE	4501	40	TNT	IDMAK			
5	DS	BMOU6014430	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0186			FULL	GE	4501	40	TNT	IDMAK			
6	DS	CAU5220273	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	1715			FULL	RF	2281	20	TNT	IDMAK			
7	DS	CICU8585990	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0087			FULL	GE	2201	20	TNT	IDMAK			
8	DS	CICU8587544	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0934			FULL	GE	2201	20	TNT	IDMAK			
9	DS	CICU8587950	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	1575			FULL	GE	2201	20	TNT	IDMAK			
10	DS	CICU8589742	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0272			FULL	GE	2201	20	TNT	IDMAK			
11	DS	CICU8589204	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0372			FULL	GE	2201	20	TNT	IDMAK			
12	DS	CICU8589280	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0259			FULL	GE	2201	20	TNT	IDMAK			
13	DS	CICU8584748	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0147			FULL	GE	2201	20	TNT	IDMAK			
14	DS	CICU8585766	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0309			FULL	GE	2201	20	TNT	IDMAK			
15	DS	CICU8589130	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0258			FULL	GE	2201	20	TNT	IDMAK			
16	DS	CICU8589289	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	1387			FULL	GE	2201	20	TNT	IDMAK			
17	DS	CICU8589280	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0934			FULL	GE	2201	20	TNT	IDMAK			
18	DS	CICU8587258	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0251			FULL	GE	2201	20	TNT	IDMAK			
19	DS	CICU8589282	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0285			FULL	GE	2201	20	TNT	IDMAK			
20	DS	CICU8589904	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	0813			FULL	GE	2201	20	TNT	IDMAK			
21	DS	CICU8600389	IM	GOT		V	TATE-0005	TANTO TERANG	208/22	G	1393			FULL	GE	2201	20	TNT	IDMAK			

Gambar 11 Sampel data *Terminal In & Out*

### 3.5.2 Data Selection

Setelah dataset diinput, kemudian atribut-atribut dataset yang tidak diperlukan akan diseleksi sehingga tersisa atribut yang relevan dengan penelitian. Selanjutnya data-data selain yang sudah diseleksi akan dibersihkan atau an pada tahapan data *cleaning*. Atribut-atribut yang diperlukan dalam ini antara lain:



abuhan asal

- status kontainer
- ukuran kontainer
- jenis kontainer
- komoditas

Tabel 3 Data yang dipilih dari sumber data *Inbound List*

No	Cargo	POL	F/M	Comm.	UT	GW
1.	CAIU2054482	IDJKT	FULL	GE	G	28900
2.	CAIU7172390	IDJKT	FULL	GE	G	NaN
3.	CAIU7550783	IDJKT	FULL	GE	G	30060
4.	CTPU2705679	IDJKT	FULL	GE	G	NaN
5.	CTPU2709611	IDJKT	FULL	GE	G	24140
6.	CTPU2710628	IDJKT	FULL	GE	G	28640
7.	CTPU2713632	IDJKT	FULL	GE	G	NaN
8.	CTPU2714006	IDJKT	FULL	GE	G	NaN
...	...	...	...	...	...	...
8992.	TGHU9643598	IDJKT	FULL	GE	G	9700
8993.	THRU5031481	IDJKT	FULL	RF	R	30000
8994.	TRHU8891185	IDJKT	FULL	GE	G	27300

Tabel 4 Data yang dipilih dari sumber data *Terminal In & Out*

No	Cargo	Length	Dwelling Time
1.	CAIU2054482	20	3.99
2.	CAIU7172390	40	2.3
3.	CAIU7550783	40	4.26
4.	CTPU2705679	20	1.96
5.	CTPU2709611	20	3.77
6.	CTPU2710628	20	8.02
7.	CTPU2713632	20	1.75
8.	CTPU2714006	20	1.39
...	...	...	...
8992.	TGHU9643598	40	0.91



<b>8993.</b>	THRU5031481	40	0.13
<b>8994.</b>	TRHU8891185	40	0.33

### 3.5.3 Data Preprocessing

Data yang telah diseleksi selanjutnya akan melalui tahap *preprocessing* agar data yang dihasilkan dapat lebih efisien dan optimal untuk diolah sistem. Berikut adalah tahap-tahap yang dilakukan pada *preprocessing*:

#### 3.5.3.1 Data Integration

Penelitian ini memiliki 2 sumber data yang berbeda, sehingga diperlukan integrasi data atau penggabungan data. Lebih detailnya, terdapat sumber data “*Inbound List*” dan “*Terminal In and Out*”. Pada kedua sumber data, terdapat masing-masing 12 data excel informasi muatan kapal kontainer, sehingga totalnya ada 24 file excel. Maka dilakuka penggabungan keseluruhan data untuk menjadi kesatuan dataset yang dapat diproses. Dalam melakukan penggabungan data, diperlukan atribut yang sama di kedua sumber agar baris datanya sesuai. Pada data yang dimiliki, terdapat atribut yang sama dalam kedua sumber data yaitu kode kontainer, namun karena sumber datanya sudah rapi dan keduanya memiliki baris dengan kunci (*key*) data yang sama, maka langsung dilakukan penggabungan kedua sumber data. Dimulai dengan menggabungkan data keseluruhan kapal pada sumber data “*Inbound List*” dan kemudian data keseluruhan kapal pada sumber data “*Terminal In and Out*”, dengan hasil pada Tabel 5 berikut.

Tabel 5 Hasil penggabungan dataset

No.	POL	F/M	Comm.	UT	GW	Length	Dwelling Time
1.	IDJKT	FULL	GE	G	28900	20	3.99
2.	IDJKT	FULL	GE	G	NaN	40	2.3
3.	IDJKT	FULL	GE	G	30060	40	4.26
4.	IDJKT	FULL	GE	G	NaN	20	1.96
	IDJKT	FULL	GE	G	24140	20	3.77
	IDJKT	FULL	GE	G	28640	20	8.02



7.	IDJKT	FULL	GE	G	NaN	20	1.75
8.	IDJKT	FULL	GE	G	NaN	20	1.39
...	...	...	...	...	...	...	...
8992.	IDJKT	FULL	GE	G	9700	40	0.91
8993.	IDJKT	FULL	RF	R	30000	40	0.13
8994.	IDJKT	FULL	GE	G	27300	40	0.33

### 3.5.3.2. Data Cleaning

Pada proses *cleaning* data, ada banyak data yang harus dibersihkan dari dataset seperti menghilangkan atribut yang tidak akan dipakai, menghilangkan *missing value* (*default Nan*) maupun menggantinya dengan median ataupun mean dari atribut tersebut .

Pertama, akan dibersihkan atribut yang tidak dipakai. Untuk ini, jumlah baris dataset tetap sama seperti sebelumnya karena proses ini hanya menghilangkan kolom yaitu kolom “Index” dan “Id”.

```
POL          0
F/M          0
Comm.        0
UT           0
GW           232
Length       0
Dwelling Time 28
dtype: int64
```

Gambar 12 Informasi nilai null pada setiap *feature*

Selanjutnya dilakukan deteksi untuk *missing value*, hasilnya menunjukkan terdapat *missing values* pada *Gross* dan *Dwelling Time*. Gambar 12 menunjukkan terdapat 232 data null pada *GW* dan 28 data null pada *Dwelling Time*. Untuk *Dwelling Time*, karena *missing value* nya terbilang sedikit (28 data), maka diputuskan untuk menghilangkan saja keseluruhan data tersebut. Kemudian untuk *Gross*, dilakukan penggantian null dengan mean dari *gross* sesuai ukuran kontainer.

```
Length
20    16867.841911
40    17057.474493
45    29666.666667
Name: GW, dtype: float64
```



Gambar 13 Median *gross* untuk setiap ukuran kontainer

Gambar 13 menunjukkan median *gross* untuk setiap kelompok ukuran kontainer untuk dilakukan penggantian null. Contohnya untuk *gross* yang memiliki ukuran kontainer 20 akan diganti dengan 16867.841911 kg, kemudian untuk *gross* yang memiliki ukuran kontainer 40 akan diganti dengan 17057,474493 kg dan untuk *gross* yang memiliki ukuran kontainer 45 akan diganti dengan 29666,666667 kg.

Setelah dilakukan pembersihan dari *missing value*, dataset yang tersisa berjumlah 8994 baris data. Tabel 6 berikut menunjukkan pembersihan *missing value*.

Tabel 6 Hasil pembersihan *missing value*

No.	POL	F/M	Comm.	UT	GW	Length	Dwelling Time
1.	IDJKT	FULL	GE	G	28900	20	3.99
2.	IDJKT	FULL	GE	G	17057.47449	40	2.3
3.	IDJKT	FULL	GE	G	30060	40	4.26
4.	IDJKT	FULL	GE	G	16867.84191	20	1.96
5.	IDJKT	FULL	GE	G	24140	20	3.77
6.	IDJKT	FULL	GE	G	28640	20	8.02
7.	IDJKT	FULL	GE	G	16867.84191	20	1.75
8.	IDJKT	FULL	GE	G	16867.84191	20	1.39
...	...	...	...	...	...	...	...
8992.	IDJKT	FULL	RF	R	30000	40	0.13
8993.	IDJKT	FULL	GE	G	27300	40	0.33
8994.	IDJKT	FULL	GE	G	21880	20	2.89

Setelah pengamatan lebih lanjut, disadari bahwa terdapat banyak anomali di dataset yang memiliki keunikan nilai ekstrim, seperti pada *Dwelling Time*. Sehingga dilakukan juga penyortiran agar rentang data yang ada tidak terlalu lebar.

Untuk *Dwelling Time*, dilakukan penghilangan seluruh data yang memiliki *Time* di atas 11,9. Nilai ini diambil dari rata-rata *dwelling time* di berdasarkan sumber data. Tabel 7 berikut merupakan sampel data yang alui tahapan data *cleaning*.



Tabel 7 Hasil *data cleaning*

No.	POL	F/M	Comm.	UT	GW	Length	Dwelling Time
1.	IDJKT	FULL	GE	G	28900	20	3.99
2.	IDJKT	FULL	GE	G	17057.47	40	2.3
3.	IDJKT	FULL	GE	G	30060	40	4.26
4.	IDJKT	FULL	GE	G	16867.84	20	1.96
5.	IDJKT	FULL	GE	G	24140	20	3.77
6.	IDJKT	FULL	GE	G	28640	20	8.02
7.	IDJKT	FULL	GE	G	16867.84	20	1.75
8.	IDJKT	FULL	GE	G	16867.84191	20	1.39
...	...	...	...	...	...	...	...
8663.	IDJKT	FULL	RF	R	30000	40	0.13
8664.	IDJKT	FULL	GE	G	27300	40	0.33
8665.	IDJKT	FULL	GE	G	21880	20	2.89

```
df_pre_pro10.shape
(8665, 7)
```

Gambar 14 Banyaknya data hasil *data cleaning*

Berdasarkan Gambar 14, hasilnya terdapat sebanyak total 8665 baris dataset yang telah melalui tahap *data cleaning* dan akan diproses lebih lanjut untuk mengoptimalkan hasil prediksi.

### 3.5.3.3. Outlier Handling

Dalam optimasi dataset, untuk menangani anomali yang banyak terdapat pada data *Dwelling Time*, dilakukan penghilangan *outlier* dengan menghitung batas bawah dan atas *outlier*. Kemudian dataset dibersihkan dari dataset yang berada di bawah batas bawah dan di atas batas atas *outlier*, sampai data *Dwelling Time* tidak

iliki *outlier* yang signifikan. Tabel 8 berikut merupakan hasil penanganan yang dilakukan. Pada tahapan ini dilakukan juga perubahan nama kolom 17.



Tabel 8 Hasil *outlier handling*

No.	Pelabuhan Asal	Status Kontainer	Komoditas	Jenis Kontainer	Gross	Ukuran Kontainer	Dwelling Time
1.	IDJKT	FULL	GE	G	28900	20	3.99
2.	IDJKT	FULL	GE	G	17057.47	40	2.3
3.	IDJKT	FULL	GE	G	30060	40	4.26
4.	IDJKT	FULL	GE	G	16867.84	20	1.96
5.	IDJKT	FULL	GE	G	24140	20	3.77
6.	IDJKT	FULL	GE	G	16867.84	20	1.75
7.	IDJKT	FULL	GE	G	16867.84	20	1.39
8.	IDJKT	FULL	GE	G	28380	20	5.69
...	...	...	...	...	...	...	...
8179.	IDJKT	FULL	RF	R	30000	40	0.13
8180.	IDJKT	FULL	GE	G	27300	40	0.33
8181.	IDJKT	FULL	GE	G	21880	20	2.89

```
df_clean2.shape
(8181, 7)
```

Gambar 15 Banyaknya data hasil *outlier handling*

Berdasarkan Gambar 15, setelah dilakukan *outlier handling*, tersisa 8.181 baris dataset yang kemudian akan ditransformasikan ke dalam tipe data yang dapat diterima oleh algoritma.

#### 3.5.3.4. Data Transformation

Pada transformasi data, dataset yang awalnya merupakan data objek atau kategorikal akan diubah menjadi data numerik atau data variabel kontinu. Hal ini dikarenakan algoritma untuk prediksi yang dipakai adalah pendekatan regresi yaitu SVR dan ANN Dimana kedua algoritma ini hanya menerima data numerik. Transformasi dilakukan dengan menggunakan label *encoding*, yaitu penyematan label pada setiap *unique itemset*. Tabel 9 berikut merupakan hasil transformasi data

1 8 dan merupakan tahap akhir preprocessing.



Tabel 9 Hasil transformasi data

No.	Pelabuhan Asal	Status Kontainer	Komo-ditas	Jenis Kontainer	Ukuran Kontainer	Gross	Dwelling Time
1.	0	1	0	1	20	28900	3.99
2.	0	1	0	1	40	17057.47	2.3
3.	0	1	0	1	40	30060	4.26
4.	0	1	0	1	20	16867.84	1.96
5.	0	1	0	1	20	24140	3.77
6.	0	1	0	1	20	16867.84	1.75
7.	0	1	0	1	20	16867.84	1.39
8.	0	1	0	1	20	28380	5.69
...	...	...	...	...	...	...	...
8179.	0	1	3	2	40	30000	0.13
8180.	0	1	0	1	40	27300	0.33
8181.	0	1	0	1	20	21880	2.89

### 3.6 Prediksi/Peramalan Menggunakan *Support Vector Machine*

Setelah mendapatkan *preprocessing* dataset, parameter yang dipilih digunakan sebagai parameter untuk memprediksi *dwelling ttime*. Pada tahapan ini, dilakukan beberapa tahap untuk membuat model SVR. Langkah-langkah prediksi dengan algoritma *Support Vector Regression* (SVR) ini dapat digambarkan sebagai berikut:

1. Pembagian Data Prediktor dan Data Target

Menentukan data prediktor (x) dan data target (y). Pada penelitian ini, data prediktor yaitu Pelabuhan Asal, Status Kontainer, Komoditas, Jenis Kontainer, Ukuran Kontainer, *Gross*, dengan data target yaitu *Dwelling Time*

2. Pembagian *data training* dan *data testing*

Selanjutnya dataset dibagi menjadi data *training* dengan porsi 80% dan data testing dengan porsi 20%

3. Penentuan Parameter untuk SVR

Menentukan kernel yaitu kernel “rbf” dan melakukan metode *Grid ch* yang akan digunakan untuk proses *training*. Hal ini berguna untuk mencari nilai optimal pada parameter yang digunakan pada kernel. Penentuan parameter dalam metode yang digunakan memegang peranan penting dalam



menghasilkan akurasi yang baik. Setiap kernel memiliki beberapa perbedaan parameter yang digunakan. Gambar 16 berikut menunjukkan hasil dari proses *GridSearch* SVR.

```
Best Parameters: {'C': 1, 'epsilon': 0.5, 'kernel': 'rbf'}
Mean Squared Error: 2.6004515675292645
```

Gambar 16 Hasil *GridSearch* SVR

#### 4. Pelatihan Dataset

Pelatihan data menggunakan parameter yang diusulkan *GridSearch*. Berdasarkan Gambar 16, parameter yang dibutuhkan untuk mengimplementasikan SVR dengan kernel rbf adalah “C=1”, epsilon=0,5. Untuk mengimplementasikan SVR, dibutuhkan *boundary line* atau *decision boundary*, yaitu dua garis yang ditarik di sekitar hyperplane pada jarak tertentu (epsilon).

Untuk memprediksi SVR menggunakan kernel RBF, dapat dijabarkan sebagai berikut.

$$f(x) = \sum_{i=1}^n \alpha_i \exp\left(-\frac{\|x-x_i\|^2}{2\sigma^2}\right) + b \quad (12)$$

Sumber: Javed dkk., 2009.

Keterangan:

$n$  = banyaknya *support vectors*

$\alpha_i$  = *Langrange multipliers*

$x_i$  = *support vectors*

$b$  = *bias term*

Tabel 10 berikut menunjukkan sampel *support vectors* dari metode SVR. Banyaknya *support vector* didapatkan dari *training* dengan mengambil *data points* terdekat dengan *hyperplane*.

Tabel 10 *Support vectors* dari metode SVR



	1	2	3	4	5	6
	-0.31299252	0.047931641	-0.196550377	-0.084042066	1.448266842	-0.900274384
	-0.31299252	0.047931641	-0.196550377	-0.084042066	-0.690028203	0.653995858
	-0.31299252	0.047931641	-0.196550377	-0.084042066	1.448266842	-1.029796904

4.	-0.31299252	0.047931641	-0.196550377	-0.084042066	-0.690028203	1.400045574
5.	-0.31299252	0.047931641	-0.196550377	-0.084042066	-0.690028203	-1.677409505
6.	-0.31299252	0.047931641	-0.196550377	-0.084042066	-0.690028203	-0.511706823
7.	3.286696499	0.047931641	-0.196550377	-0.084042066	-0.690028203	-0.130910614
8.	-0.31299252	0.047931641	-0.196550377	-0.084042066	-0.690028203	0.576282346
...	...	...	...	...	...	...
5022.	3.286696499	0.047931641	-0.196550377	-0.084042066	-0.690028203	0.850870089
5023.	3.286696499	0.047931641	-0.196550377	-0.084042066	-0.690028203	0.653995858
5024.	-0.31299252	0.047931641	-0.196550377	-0.084042066	1.448266842	1.032201617

Selanjutnya, *support vector*, *alpha*, dan *gamma* akan dipakai untuk menghasilkan *predicted value* ( $\hat{y}_i$ ). Setelah didapatkan *predicted value*, maka selanjutnya dilakukan perhitungan bias value sebagai perbandingan antara *predicted value* ( $\hat{y}_i$ ) dan actual value/target value ( $y_i$ ). Gambar 17 menunjukkan nilai bias term dari metode ini, yaitu 2,25534413.

→ Bias term (b): [2.25534413]

Gambar 17 Bias term dari metode SVR

Contoh perhitungan bias term, misalnya diambil 2 baris pertama support vectors dari dataset, seperti pada tabel 11.

Tabel 11 Sampel data support vector ( $x$ ), Lagrange multiplier ( $\alpha$ ), dan target value ( $y$ )

No.	1	2	3	4	5	6	$\alpha$	$y$
$x_1$	-0.31299252	0.047931641	-0.196550377	-0.084042066	1.448266842	-0.900274384	-1	0.64
$x_2$	-0.31299252	0.047931641	-0.196550377	-0.084042066	-0.690028203	0.653995858	1	3.23

$$b = \frac{1}{n} \sum_{i=1}^n (y_i - \sum_{j=1}^n \alpha_j K(x_j - x_i))$$

$$b = \frac{1}{n} \sum_{i=1}^n \left( y_i - \sum_{j=1}^n \alpha_j \exp \left( -\frac{\|x_j - x_i\|^2}{2\sigma^2} \right) \right) \quad (14)$$

Sumber: Javed, dkk.

... lerhanakan menjadi:

$$b = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i) \quad (15)$$



Keterangan:

$b$  = bias term

$\sigma$  = sigma

$y$  = target value

$\hat{y}$  = predicted value

Penyelesaian:

1. Kernel Value ( $K(x_j - x_i)$ )

- $K(x_1 - x_1) = \exp(0) = 1$
- $K(x_2 - x_1)$

dengan:

$$\begin{aligned} \|x_2 - x_1\|^2 &= (-0.31 - (-0,31))^2 + (0.05 - 0,05)^2 \\ &\quad + (-0.20 - (-0,20))^2 + (-0.08 - (-0,08))^2 \\ &\quad + (1.45 - (-0,70))^2 + (-0.90 - 0,65)^2 \end{aligned}$$

$$\|x_2 - x_1\|^2 = 0 + 0 + 0 + 0 + 4,62 + 2,40$$

$$\|x_2 - x_1\|^2 = 7,02$$

$$K(x_2 - x_1) = \exp\left(-\frac{\|x_j - x_i\|^2}{2\sigma^2}\right) = \exp\left(-\frac{7,02}{2(1)^2}\right) = \exp(-3,51) = 0,03$$

- $K(x_1 - x_2)$

dengan:

$$\begin{aligned} \|x_1 - x_2\|^2 &= (-0.31 - (-0,31))^2 + (0.05 - 0,05)^2 \\ &\quad + (-0.20 - (-0,20))^2 + (-0.08 - (-0,08))^2 \\ &\quad + ((-0,70) - 1.45)^2 + (0,65 - (-0.90))^2 \end{aligned}$$

$$\|x_1 - x_2\|^2 = 0 + 0 + 0 + 0 + 4,62 + 2,40$$

$$\|x_1 - x_2\|^2 = 7,02$$

$$K(x_1 - x_2) = \exp\left(-\frac{\|x_j - x_i\|^2}{2\sigma^2}\right) = \exp\left(-\frac{7,02}{2(1)^2}\right) = \exp(-3,51) = 0,03$$

- $K(x_2 - x_2) = \exp(0) = 1$

Predicted Value ( $\hat{y}_i$ )

- $\hat{y}_1 = \alpha_1 \cdot K(x_1 - x_1) + \alpha_2 \cdot K(x_2 - x_1) = -1(1) + 1(0,03) = -0,97$
- $\hat{y}_2 = \alpha_1 \cdot K(x_1 - x_2) + \alpha_2 \cdot K(x_2 - x_2) = -1(0,03) + 1(1) = 0,97$



### 3. Bias Term ( $b$ )

$$b = \frac{1}{2}((y_1 - \hat{y}_1) + (y_2 - \hat{y}_2)) = \frac{1}{2}((0,64 - (-0,97)) + (3,23 - 0,97)) = \frac{1}{2}(1,61 + 2,26) = 1,935$$

Jadi, bias term dari contoh perhitungan adalah 1,935

Pada algoritma SVR, operasinya dilakukan berurutan hingga semua perhitungan  $x$  selesai (dari  $x_1$  hingga  $x_n$ ), yang akhirnya didapatkan nilai fungsi  $f(x)$  untuk dimasukkan setiap nilai  $X_{\text{test/target value}}$  sehingga didapatkan hasil prediksi  $y_{\text{test/predicted value}}$ .

### 4. Evaluasi Model

Model SVR yang akan dihasilkan selanjutnya dievaluasi menggunakan data baru. Data baru yang dipakai adalah data bulan Juni 2023 sebanyak 1122 baris data. Dari proses ini dihasilkan hasil prediksi *dwelling time* berdasarkan data baru dari parameter bulan Juni yang kemudian dibandingkan dengan data aktual *dwelling time* bulan Juni 2023.

### 5. Perhitungan Galat

Pada tahap ini, dilakukan perhitungan model berdasarkan hasil prediksi 2023 yang dibandingkan dengan data aktual *dwelling time* bulan Juni 2023. Metode yang dipakai pada perhitungan ini adalah RMSE dan MAE.

## 3.7 Prediksi/Peramalan Menggunakan *Artificial Neural Network*

Selain SVR, akan dilakukan juga peramalan menggunakan *Artificial Neural Network* (ANN). Langkah-langkah prediksi dengan algoritma *Artificial Neural Network* (ANN) ini dapat digambarkan dalam *flowchart* prediksi berikut ini.

#### 1. Pembagian Data Prediktor dan Data Target

Dataset dibagi menjadi data prediktor ( $x$ ) dan data target ( $y$ ). Pada penelitian ini, data prediktor yaitu Pelabuhan Asal, Status Kontainer, Komoditas, Jenis Kontainer, Ukuran Kontainer, *Gross*, dengan data target yaitu *Dwelling Time*.



#### 2. Pembagian data *training* dan data *testing*

Selanjutnya dataset dibagi menjadi data *training* dengan porsi 80% dan *testing* dengan porsi 20%.

ntuan parameter untuk ANN

Kemudian ditentukan parameter yang akan digunakan saat *training* seperti layer dan matriks. Untuk membantu memastikan apakah parameter yang dipilih sudah sesuai dilakukan *tuning* parameter dengan metode *GridSearch*. Hal ini berguna untuk mencari nilai optimal pada parameter yang digunakan. Penentuan parameter dalam metode yang digunakan memegang peranan penting dalam menghasilkan akurasi yang baik. Algoritma *GridSearch* mencoba satu persatu kombinasi nilai parameter yang ada dengan menghitung MSE dari setiap kombinasi parameter dan mengambil kombinasi dengan nilai MSE tertinggi. Untuk itu, dihasilkan parameter yang paling optimal untuk mengimplementasikan ANN adalah ‘layers’: 3, ‘neurons’: 16, ‘optimizer’: ‘adam’.

#### 4. Pelatihan dataset

Pelatihan data dilakukan dengan menggunakan parameter yang diusulkan *GridSearch*. Tahapan ini akan menghasilkan model prediksi. Berikut informasi mengenai parameter pada pelatihan dataset.

##### a. Input Layer (First Dense Layer)

Layer pertama dari *training* ini terdiri dari:

- Units: 16
- Activation Function: ReLU (Rectified Linear Unit)
- Input Dimension: 1

Output dari layer ini dioperasikan seperti berikut.

$$o_1 = \text{ReLU}(W_1 \cdot X + b_1) \quad (16)$$

Sumber: Goodfellow dkk., 2016

Keterangan:

$o_1$  = output layer pertama

$W_1$  = *weight matrix* (matriks beban) layer pertama

$X$  = input data

$b_1$  = *bias vector* (vektor bias) layer pertama

Matriks berikut menunjukkan nilai *weight matrix* dari layer pertama dengan 16 unit, yang artinya terdapat 96 nilai *weight* yaitu 16 unit  $\times$  6 *input* (banyaknya *feature* yang dimasukkan sebagai input).



$$W = \begin{bmatrix} 0,53 & -0,58 & -0,27 & -0,14 & 0,13 & -0,32 & -0,52 & 0,45 & -0,32 & -0,10 & 0,27 & 0,28 & -0,38 & 0,31 & -0,47 & -0,29 \\ -0,08 & -0,12 & 0,01 & 0,38 & 0,58 & 0,47 & 0,28 & 0,24 & -0,42 & 0,65 & -0,24 & 0,31 & 0,21 & -0,20 & 0,02 & -0,02 \\ 0,14 & -0,53 & 0,48 & 0,26 & 0,17 & -0,52 & 0,29 & -0,28 & 0,03 & -0,30 & 0,40 & -0,17 & -0,10 & -0,21 & -0,25 & -0,35 \\ 0,21 & -0,31 & -0,49 & 0,28 & 0,21 & 0,23 & -0,21 & 0,03 & -0,23 & 0,15 & 0,29 & -0,08 & 0,01 & 0,40 & -0,03 & -0,09 \\ 0,54 & -0,35 & -0,14 & 0,30 & -0,46 & -0,07 & -0,09 & -0,27 & 0,35 & 0,44 & -0,67 & 0,49 & -0,52 & -0,12 & 0,12 & -0,45 \\ 0,48 & 0,18 & -0,17 & 0,32 & -0,43 & 0,11 & 0,12 & 0,26 & -0,74 & 0,39 & 0,15 & -0,28 & 0,24 & 0,27 & 0,04 & -0,43 \end{bmatrix}$$

Berikut merupakan nilai *bias vector* dari layer pertama dengan 16 unit, yang artinya terdapat 16 nilai *bias vector*.

$$b = \begin{bmatrix} -0,04 \\ 0,09 \\ -0,04 \\ 0,05 \\ 0,33 \\ 0,12 \\ 0,16 \\ 0,23 \\ 0,13 \\ 0,20 \\ 0,18 \\ 0,00 \\ 0,01 \\ -0,16 \\ 0,10 \\ 0,08 \end{bmatrix}$$

b. *Hidden Layer (Second Dense Layer):*

Layer kedua dari *training* ini terdiri dari:

- *Units: 8*
- *Activation Function: ReLU*

*Output* dari layer ini dioperasikan seperti berikut.

$$o_2 = \text{ReLU}(W_2 \cdot o_1 + b_2) \quad (13)$$

Sumber: Goodfellow dkk., 2016

Keterangan:

$o_2$  = output layer kedua

$W_2$  = *weight matrix* (matriks beban) layer kedua

$b_2$  = *bias vector* (vektor bias) layer kedua

Matriks berikut menunjukkan nilai *weight matrix* dari layer kedua dengan 8 unit, yang artinya terdapat 48 nilai *weight* yaitu 8 unit  $\times$  16 *input* (banyaknya *output* dari layer sebelumnya yang dijadikan *input* untuk layer setelahnya)



$$W = \begin{bmatrix} 0,54 & 0,05 & -0,21 & -0,19 & 0,52 & -0,20 & -0,11 & 0,55 \\ 0,02 & 0,28 & 0,29 & -0,13 & -0,19 & -0,31 & 0,30 & 0,36 \\ 0,02 & -0,35 & -0,37 & -0,58 & -0,32 & 0,34 & 0,51 & 0,02 \\ 0,06 & 0,47 & 0,37 & 0,32 & -0,18 & 0,12 & -0,55 & -0,43 \\ 0,29 & -0,33 & -0,12 & 0,43 & 0,00 & 0,18 & 0,29 & 0,00 \\ -0,09 & 0,58 & -0,30 & 0,02 & 0,39 & 0,33 & -0,21 & 0,23 \\ -0,23 & 0,43 & 0,32 & -0,35 & 0,58 & -0,41 & -0,12 & 0,30 \\ 0,48 & 0,48 & -0,14 & 0,43 & 0,46 & -0,33 & 0,40 & 0,38 \\ -1,02 & -0,35 & 0,06 & 0,19 & 0,49 & -1,06 & -0,19 & -0,25 \\ 0,40 & 0,57 & -0,34 & 0,50 & 0,49 & -0,58 & -0,03 & 0,01 \\ -0,20 & 0,42 & 0,30 & -0,03 & 0,00 & 0,32 & 0,48 & 0,23 \\ -0,13 & -0,50 & 0,37 & -0,01 & 0,55 & 0,06 & 0,50 & 0,45 \\ 0,06 & 0,41 & 0,22 & -0,13 & -0,17 & 0,30 & 0,09 & 0,49 \\ -0,25 & -0,20 & 0,03 & -0,49 & -0,31 & -0,31 & -0,15 & -0,05 \\ 0,08 & 0,43 & 0,29 & 0,32 & 0,21 & 0,09 & -0,59 & 0,28 \\ -0,50 & -0,05 & -1,29 & 0,45 & 0,45 & 0,23 & -0,22 & 0,33 \end{bmatrix}$$

Berikut merupakan nilai *bias vector* dari layer kedua dengan 8 unit, yang artinya terdapat 8 nilai *bias vector*.

$$b = \begin{bmatrix} -0,07 \\ 0,21 \\ -0,08 \\ 0,21 \\ 0,20 \\ 0,00 \\ 0,22 \\ 0,18 \end{bmatrix}$$

c. *Output Layer*

Layer terakhir dari *training* ini terdiri dari:

- *Units*: 1
- *Activation Function*: *Linear*

*Output* dari layer ini dioperasikan seperti berikut

$$\hat{y} = \text{ReLU}(W_3 \cdot o_2 + b_3) \quad (14)$$

Sumber: Goodfellow dkk., 2016

Keterangan:

$\hat{y}$  = *output* layer terakhir

$W_3$  = *weight matrix* (matriks beban) layer ketiga

$b_3$  = *bias vector* (vektor bias) layer ketiga

Matriks berikut menunjukkan nilai *weight matrix* dari layer ketiga dengan 1 unit, yang artinya terdapat 8 nilai *weight* yaitu 1 unit  $\times$  8 *input* (banyaknya *output* dari layer sebelumnya yang dijadikan *input* untuk layer setelahnya).



$$W = \begin{bmatrix} -0,61 \\ 0,39 \\ -0,62 \\ 0,76 \\ 0,48 \\ -0,76 \\ 0,64 \\ 0,42 \end{bmatrix}$$

Berikut merupakan nilai *bias vector* dari layer ketiga dengan 1 unit, yang artinya terdapat 1 nilai *bias vector*.

$$B = [0,19]$$

#### 5. Evaluasi Model

Model ANN yang akan dihasilkan selanjutnya dievaluasi menggunakan data baru. Data baru yang dipakai adalah data bulan Juni 2023 sebanyak 1122 baris data. Dari proses ini dihasilkan hasil prediksi *dwelling time* berdasarkan data baru dari parameter bulan Juni yang kemudian dibandingkan dengan data aktual *dwelling time* bulan Juni 2023.

#### 6. Perhitungan Galat

Pada tahap ini, dilakukan perhitungan model berdasarkan hasil prediksi 2023 yang dibandingkan dengan data aktual *dwelling time* bulan Juni 2023. Metode yang dipakai pada perhitungan ini adalah RMSE dan MAE.



## BAB 4. HASIL DAN PEMBAHASAN

### 4.1 Dataset

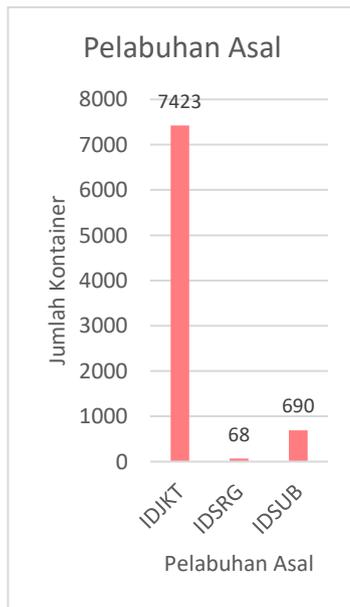
*Preprocessing* dataset yang akan dipakai pada metode SVR memiliki tahapan yang sama dengan yang akan dipakai pada metode ANN. Setelah dataset melalui proses *preprocessing*, dataset dianggap siap untuk dilakukan lebih lanjut dalam rangka ekstraksi pengetahuan. Tabel 12 menunjukkan sampel data dari dataset yang dipakai untuk *Support Vector Regression* (SVR) dan *Artificial Neural Network* (ANN).

Tabel 12 Sampel data yang dipakai untuk melakukan prediksi

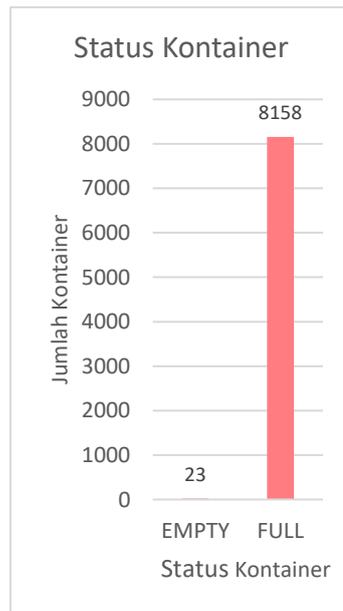
No.	Pelabuhan Asal	Status Kontainer	Komo-ditas	Jenis Kontainer	Ukuran Kontainer	Gross	Dwelling Time
1.	0	1	0	1	20	28900	3.99
2.	0	1	0	1	40	17057.47	2.3
3.	0	1	0	1	40	30060	4.26
4.	0	1	0	1	20	16868	1.96
5.	0	1	0	1	20	24140	3.77
6.	0	1	0	1	20	16868	1.75
7.	0	1	0	1	20	16868	1.39
8.	0	1	0	1	20	28380	5.69
...	...	...	...	...	...	...	...
8179.	0	1	3	2	40	30000	0.13
8180.	0	1	0	1	40	27300	0.33
8181.	0	1	0	1	20	21880	2.89

Sebelum diproses, beberapa informasi mengenai kondisi dataset dianalisis. Informasi mengenai kondisi dataset seperti persebaran data, korelasi data dan lainnya dapat dilihat pada gambar 18 berikut.

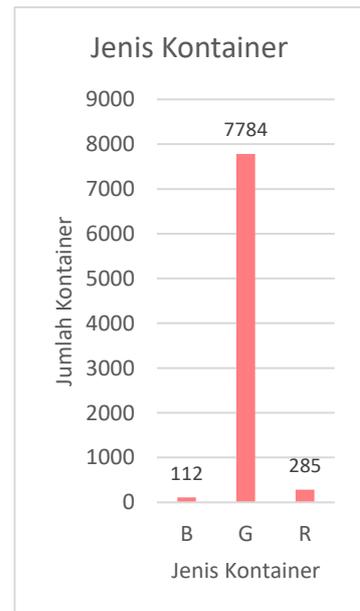




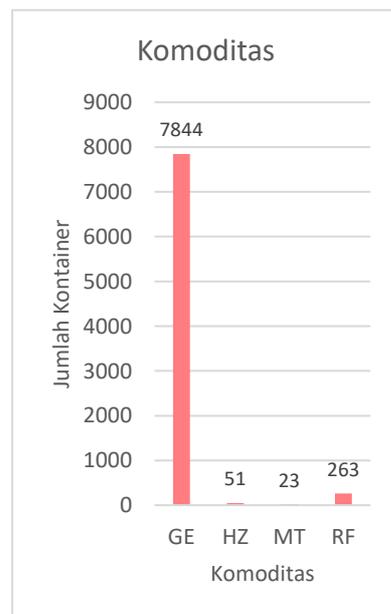
(a)



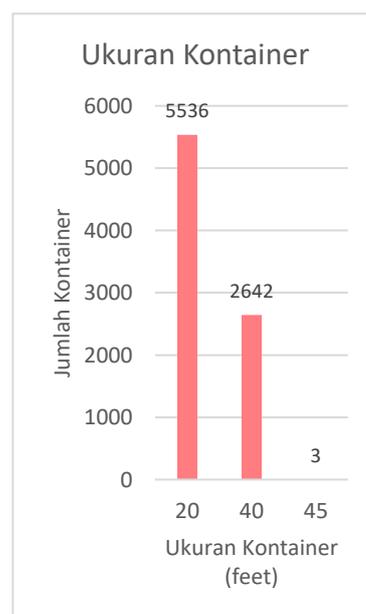
(b)



(c)

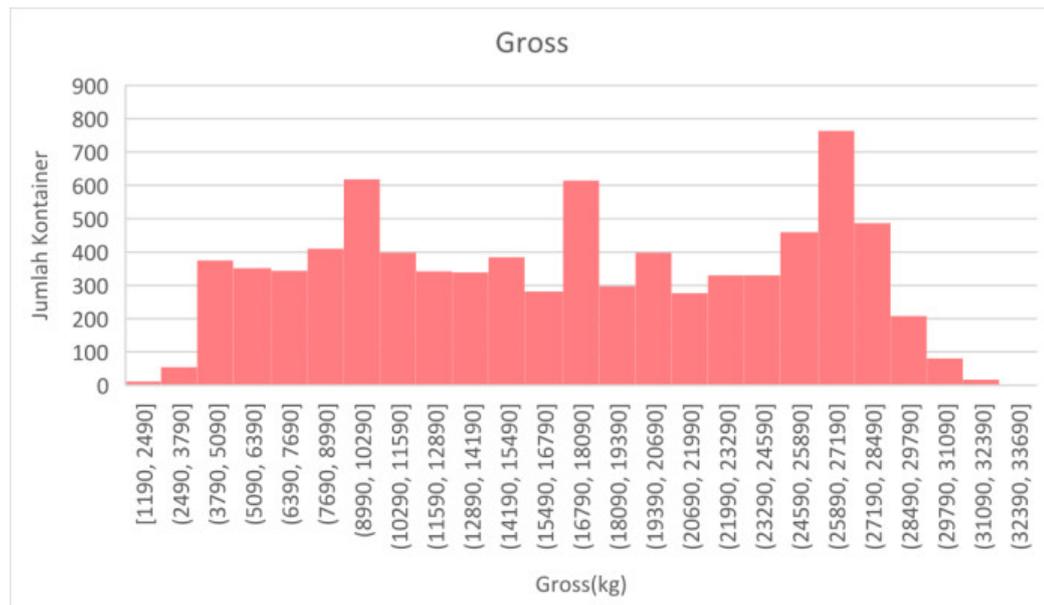


(d)

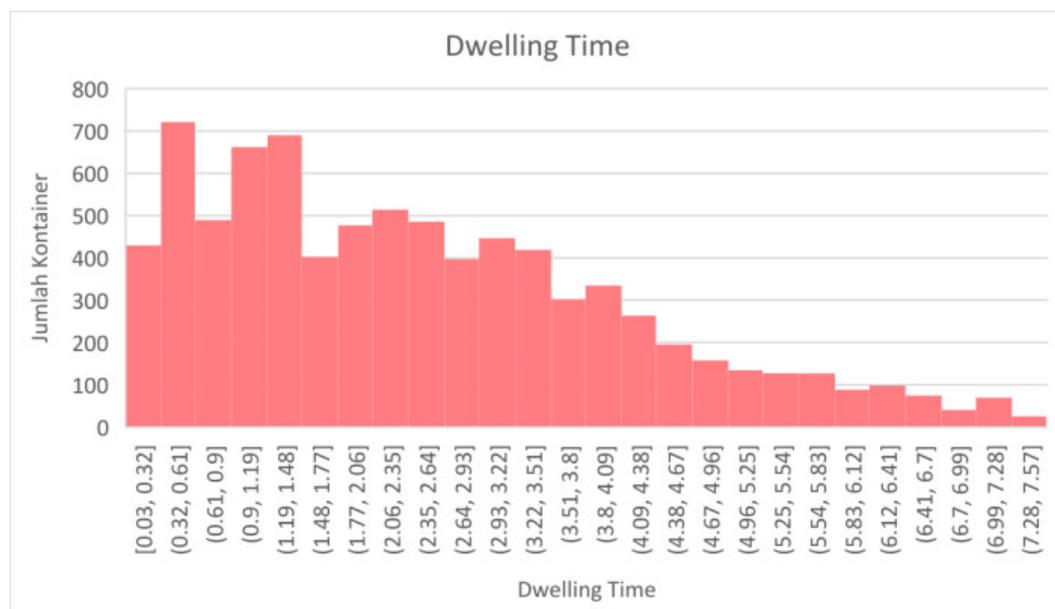


(e)





(f)



(g)

Gambar 18 Informasi persebaran data (*data distribution*) (a) Pelabuhan asal, (b) Status kontainer, (c) Komoditas (d) Jenis kontainer, (e) Ukuran kontainer, (f) *Gross*, (g) *Dwelling time*

Berdasarkan Gambar 18 mengenai informasi mengenai persebaran data (*tribution*), dapat dilihat bahwa persebaran data dari parameter Pelabuhan asal, Status Kontainer, Komoditas, dan Jenis Kontainer mendekati konstan,



kemudian variabel Ukuran Kontainer dan *Gross* yang sedikit variatif, serta variabel *Dwelling Time* yang cenderung menurun untuk setiap rentang nilainya.

Tabel 13 Sampel data yang memiliki perbedaan pola kombinasi data

Pelabuhan Asal	Status Kontainer	Komoditas	Jenis Kontainer	Ukuran Kontainer	Gross	Dwelling Time
0	1	0	1	40	30000	5.56
0	1	0	1	40	30000	0.8

Terlihat juga pola kombinasi data yang tidak konsisten, seperti pada Tabel 13 dimana pada kedua data memiliki nilai variabel yang sama namun memiliki nilai *dwelling time* yang berbeda jauh.

Tabel 14 Informasi korelasi data (*data correlation*)

	Pelabuhan Asal	Status Kontainer	Komoditas	Jenis Kontainer	Ukuran Kontainer	Gross	Dwelling Time
Pelabuhan Asal	1	-0.06559	0.051846	0.043131	-0.08354	0.028401	0.15952
Status Kontainer	-0.06559	1	-0.18491	-0.23703	-0.07678	0.067404	0.000786
Komoditas	0.051846	-0.18491	1	0.834591	0.005995	-0.02712	-0.13879
Jenis Kontainer	0.043131	-0.23703	0.834591	1	0.050101	-0.07202	-0.12713
Ukuran Kontainer	-0.08354	-0.07678	0.005995	0.050101	1	-0.00603	-0.07818
Gross	0.028401	0.067404	-0.02712	-0.07202	-0.00603	1	0.107306
Dwelling Time	0.15952	0.000786	-0.13879	-0.12713	-0.07818	0.107306	1

Selanjutnya, terlihat pada Tabel 14 untuk korelasi data semua variabel memiliki korelasi yang cenderung lemah positif maupun negatif. Sebagai contoh hubungan antara variabel *Gross* dan *Dwelling Time* yang lemah positif ( $r = 0,107306$ ), serta hubungan antara variabel Komoditas dan *Dwelling Time* yang lemah negatif ( $r = -0,13879$ ).

Sebelum melakukan prediksi, dilakukan uji ANOVA terhadap dataset yang akan dipakai. Hasil dari uji ANOVA dapat dilihat pada Gambar 19 berikut.



	sum_sq	df	F	PR(>F)
Pelabuhan_Asal	573.039026	1.0	211.680603	2.300681e-47
Status_Kontainer	19.579064	1.0	7.232506	7.174052e-03
Komoditas	125.756970	1.0	46.454622	1.004191e-11
Jenis_Kontainer	2.612941	1.0	0.965220	3.259042e-01
Ukuran_Kontainer	97.703813	1.0	36.091786	1.962936e-09
Gross	230.362166	1.0	85.095779	3.560162e-20
Residual	22127.776138	8174.0	NaN	NaN

Taraf signifikan variabel:  
Pelabuhan\_Asal: Signifikan/Passed ANOVA  
Status\_Kontainer: Signifikan/Passed ANOVA  
Komoditas: Signifikan/Passed ANOVA  
Jenis\_Kontainer: Tidak Signifikan/Failed ANOVA  
Ukuran\_Kontainer: Signifikan/Passed ANOVA  
Gross: Signifikan/Passed ANOVA

Gambar 19 Hasil uji ANOVA dataset

Berdasarkan hasil uji ANOVA pada Gambar 19, terlihat bahwa lima dari enam variabel yang akan dipakai lolos uji ANOVA karena memiliki *P-value* kurang dari 0,05, yaitu Pelabuhan Asal, Status Kontainer, Komoditas, Ukuran Kontainer, dan Gross. Sedangkan untuk satu variabel lainnya yaitu Jenis Kontainer tidak lolos uji karena memiliki *P-value* lebih dari 0,05. Ini menunjukkan bahwa tidak ada perbedaan *Dwelling Time* yang signifikan secara statistik di antara berbagai kategori variabel Jenis Kontainer.

## 4.2 Penerapan Algoritma *Support Vector Regression*

Penelitian ini menggunakan algoritma *Support Vector Regression* sebagai salah satu metode untuk memprediksi *Dwelling Time*. Adapun penerapan algoritma SVR dilakukan dalam beberapa tahapan, yaitu penginputan dataset, *feature scaling*, pembagian data *training* dan data *testing*, Pembangunan model SVR (penentuan kernel dan parameter menggunakan *Grid Search*), hingga pengujian model SVR dengan data baru dan perhitungan nilai galat atau error.

### 4.2.1 Feature Selection

Pemilihan fitur atau variabel prediktor pada penelitian ini dilakukan dengan menggunakan menggunakan seluruh fitur atau variabel penelitian sebagai input memprediksi target, yaitu Pelabuhan Asal, Status Kontainer, Komoditas, Kontainer, Jenis Kontainer, dan Ukuran Kontainer.



#### 4.2.2 Feature Scaling

Pada tahapan ini, dilakukan *scaling* data dengan pendekatan teknik *standardization* agar data *training* dan data *testing* berada pada skala atau rentang yang sama sehingga kompleksitas perhitungan berkurang dan mengurangi tingkat error dalam komputasi. Berikut rumus yang digunakan untuk standarisasi:

$$X_{stand} = \frac{x - \text{mean}(x)}{\text{standard deviation } x} \quad (19)$$

Sumber: Ali dkk., 2014

Keterangan:

$X_{stand}$  = standarisasi

$x$  = item data

Contoh perhitungan sampel

$$X(Gross_{x_1}) = \frac{28900 - 16990,45}{7723,59} = 1,542$$

#### 4.2.3 Pembagian Data Training dan Testing

Pada tahap ini, data dibagi menjadi 2 subset terpisah, yaitu data *training* dan data *testing*. Set data *training* digunakan untuk melatih model dengan memberikan fitur *input* dan nilai target yang sesuai, sehingga model dapat belajar pola dan hubungan dalam data. Set data *testing* kemudian digunakan untuk memprediksi data pembagian dan membandingkannya dengan data aktual yang telah ada. Pada penelitian ditentukan pembagian data *training* sebesar 0,8 dan data *testing* sebesar 0,2, yaitu sebanyak 80%  $X_{train}$  dan  $y_{train}$  serta sebanyak 20%  $X_{test}$  dan  $y_{test}$  dari total keseluruhan dataset. Tabel 15 merupakan pembagian data *training* dan data *testing*.

Tabel 15 Pembagian data *training* dan data *testing* pada metode SVR

Jumlah Data	Training	Testing
8181	6544	1637



#### 4.2.4 Menentukan Kernel

Untuk menentukan kernel yang akan dipakai, dilakukan tuning parameter menggunakan algoritma *GridSearch*. Pada *tuning* ini, ditentukan parameter-parameter yang akan dipakai yaitu kernel, epsilon, dan nilai C.

```
Best Parameters: {'svr_C': 0.1, 'svr_epsilon': 0.5, 'svr_kernel': 'rbf'}
```

Gambar 20 Hasil *tuning* parameter SVR

Berdasarkan hasil pada Gambar 20, parameter terbaik menurut *GridSearch* adalah  $C = 0,1$  ;  $\epsilon = 0,5$  dan kernel = rbf. Parameter-parameter ini yang kemudian akan digunakan untuk *training* data.

#### 4.2.6 Hasil Training

Untuk *training*, digunakan parameter dari hasil *tuning* parameter. Hasil *training* dataset menggunakan mencapai RMSE 1,612477643604 dan MAE 1,2583658658860322 dengan perhitungan akurasi 49,18496618106338%. Tabel 16 berikut adalah hasil *training* dengan menggunakan metode SVR.

Tabel 16 Hasil *training* pada prediksi menggunakan metode SVR

No	Actual ( $y_{test}$ )	Predicted ( $y_{pred}$ )
1.	1.05	1.960052
2.	1.2	1.657647
3.	2.68	1.629721
4.	1.62	1.894597
5.	0.61	1.921832
6.	2.93	2.475897
7.	0.11	3.478204
8.	1.13	1.606907
9.	4.58	3.133275
10.	1.27	1.901535
...	...	...
1633.	2.81	2.179319
1634.	1.99	2.492951



<b>1635.</b>	1.78	1.950357
<b>1636.</b>	3.13	1.597557
<b>1637.</b>	0.6	1.894597

#### 4.2.8 Hasil Prediksi

Setelah didapatkan parameter terbaik untuk model prediksi, dilakukan peramalan menggunakan dataset bulan Juni 2023 menggunakan dataset Pelabuhan Asal, Status Kontainer, Komoditas, Jenis Kontainer, Ukuran Kontainer, dan *Gross* sebagai prediktor dan dataset *Dwelling Time* sebagai target. Pada penelitian ini akan memprediksi *dwelling time* di bulan Juni 2023 dan dibandingkan dengan data aktualnya. Tabel 17 berikut hasil prediksi dari *testing* dataset bulan Juni 2023.

Tabel 17 Hasil prediksi menggunakan metode SVR

<b>No.</b>	<b><i>Actual</i> (y<sub>new</sub>)</b>	<b><i>Predicted</i> (y<sub>pred</sub>)</b>
<b>1.</b>	0.83	2.487124
<b>2.</b>	1.48	2.255108
<b>3.</b>	5.67	2.255111
<b>4.</b>	1.39	2.255219
<b>5.</b>	2.2	2.255219
<b>6.</b>	4.18	2.255219
<b>7.</b>	1.45	2.255119
<b>8.</b>	3.16	2.25519
<b>9.</b>	1.26	2.255377
<b>10.</b>	5.02	2.255108
...	...	...
<b>1118.</b>	5	1.651131667
<b>1119.</b>	0.91	1.601653917
<b>1120.</b>	0.13	1.529285812
<b>1121.</b>	0.33	2.325949902
<b>1122.</b>	2.89	2.406068351



#### 4.2.10 Perhitungan Nilai Error

Performansi atau kinerja dalam metode SVR dilakukan evaluasi menggunakan model dengan menghitung *error* menggunakan RMSE (*Root Mean Squared Error*), dan MAE (*Mean Absolute Error*). Gambar 21 berikut merupakan hasil perhitungan RMSE dan MAE dari prediksi *dwelling time*.

```
RMSE: 1.8462758393235088
MAE: 1.3150228024521156
Accuracy: 50.25402786427172
```

Gambar 21 Hasil perhitungan nilai error saat evaluasi model SVR

### 4.3 Penerapan Algoritma *Artificial Neural Network*

#### 4.3.1 Dataset

Dataset yang digunakan pada metode ANN ini sama dengan dataset yang digunakan pada metode SVR, agar bisa dibandingkan.

#### 4.3.2 Data Selection

Sama seperti pada metode SVR, pemilihan fitur atau variabel prediktor pada metode ini dilakukan dengan menggunakan seluruh fitur atau variabel penelitian sebagai *input* untuk memprediksi target, yaitu Pelabuhan Asal, Status Kontainer, Komoditas, *Gross Kontainer*, Jenis Kontainer, dan Ukuran Kontainer

#### 4.3.3 Data Scaling

Data scaling pada metode ini sama seperti data scaling pada metode SVR yaitu menggunakan pendekatan teknik *standardization* Berikut rumus yang digunakan untuk standarisasi.

$$X_{stand} = \frac{x - \text{mean}(x)}{\text{standard deviation } x} \quad (20)$$

Sumber: Ali dkk., 2014

Keterangan:

$X_{stand}$  = standarisasi

item data

berikut merupakan hasil standarisasinya.



Tabel 18 Hasil *scaling* data pada metode ANN

No.	Pelabuhan Asal	Status Kontainer	Komoditas	Jenis Kontainer	Ukuran Kontainer	Gross
1.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	1.542063726
2.	-0.315685273	0.053097255	-0.199369039	-0.096440164	1.446040729	0.008677961
3.	-0.315685273	0.053097255	-0.199369039	-0.096440164	1.446040729	1.69226205
4.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	-0.015875915
5.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	0.925732671
6.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	-0.015875915
7.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	-0.015875915
8.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	1.474733442
...	...	...	...	...	...	...
8179.	-0.315685273	0.053097255	5.323333346	4.46412044	1.446040729	1.684493171
8180.	-0.315685273	0.053097255	-0.199369039	-0.096440164	1.446040729	1.334893623
8181.	-0.315685273	0.053097255	-0.199369039	-0.096440164	-0.691181475	0.633104901

#### 4.3.4 Pembagian Data Training and Data Testing

Pada tahap ini, data dibagi menjadi 2 subset terpisah, yaitu data *training* dan data *testing*. Sama seperti pada metode SVR, pada metode ini ditentukan pembagian data *training* sebesar 0,8 dan data *testing* sebesar 0,2. Tabel 19 merupakan pembagian data *training* dan data *testing*

Tabel 19 Pembagian data *training* dan data *testing* pada metode ANN

Jumlah Data	Training	Testing
8903	7122	1781

#### 4.3.5 Hasil Training

Pada metode ANN, sejak awal telah dicoba *training* dataset dengan jumlah sebanyak 3.280 baris data, namun dihasilkan galat yang cukup besar sehingga tidak memuaskan yaitu RMSE 1,78 dan MAE 1,4 dengan perhitungan akurasi 52,13%

Sedangkan untuk training pada dataset yang berjumlah 8.181 baris data, digunakan parameter dari hasil tuning parameter. Kemudian didapatkan hasil *training* dataset mencapai RMSE 1,5767235963472337 dan MAE 3324031492 dengan perhitungan akurasi 48,442205551223296%. Tabel 19 menunjukkan hasil *training* dengan menggunakan metode ANN.



Tabel 20 Hasil *training* pada prediksi dengan menggunakan metode ANN

No.	Actual ( $y_{test}$ )	Predicted ( $y_{pred}$ )
1.	1.05	2.306303263
2.	1.2	2.116555452
3.	2.68	1.952313185
4.	1.62	2.264714241
5.	0.61	2.38733077
6.	2.93	2.710497141
7.	0.11	3.76769042
8.	1.13	1.803342581
9.	4.58	3.381915808
10.	1.27	2.519475222
...	...	...
1633.	2.81	2.376702309
1634.	1.99	2.620886564
1635.	1.78	2.457422495
1636.	3.13	1.868123293
1637.	0.6	2.264714241

#### 4.3.6 Hasil Prediksi

Setelah di dapatkan parameter terbaik untuk model prediksi, pada skenario 1 dilakukan peramalan menggunakan dataset bulan Juni 2023 menggunakan dataset Pelabuhan Asal, Status Kontainer, Komoditas, Jenis Kontainer, Ukuran Kontainer, dan *Gross* sebagai prediktor dan dataset *Dwelling Time* sebagai target. Pada penelitian ini akan memprediksi *dwelling time* di bulan Juni 2023 dan dibandingkan dengan data aktualnya. Tabel 21 berikut hasil prediksi dari *testing* dataset bulan Juni 2023.

Tabel 21 Hasil prediksi menggunakan metode ANN

No.	Actual ( $y_{test}$ )	Predicted ( $y_{pred}$ )
1.	0.83	1.06008



2.	1.48	2.760643
3.	5.67	2.771447
4.	1.39	2.691529
5.	2.2	2.691529
6.	4.18	2.691529
7.	1.45	2.786246
8.	3.16	2.748313
9.	1.26	2.664038
10.	5.02	2.760643
...	...	...
1118.	4.26	2.899278
1119.	4.25	2.879517
1120.	1.11	0.883685
1121.	1.86	0.940234
1122.	2.75	2.574736

#### 4.3.7 Perhitungan Nilai Error

Performansi atau kinerja dalam metode SVR dilakukan evaluasi menggunakan model dengan menghitung *error* menggunakan RMSE (*Root Mean Squared Error*), dan MAE (Mean Absolute Error). Gambar 22 berikut merupakan hasil perhitungan RMSE dan MAE dari prediksi *dwelling time*.

```
Root Mean Squared Error (RMSE): 1.7630370598814256
Mean Absolute Error (MAE): 1.2679273941502003
R-squared (R^2): 0.049740613980378545
Accuracy: 52.03559915317957
```

Gambar 22 Hasil perhitungan nilai error saat evaluasi model ANN

## 4.4 Pembahasan

### 4.4.1 Analisis Hasil Prediksi SVR dan ANN



Pada analisis regresi dengan tabel ANOVA, didapatkan hasil seperti pada [9]. Berdasarkan analisis dari sumber yang dibaca, ada beberapa alasan hal terjadi, di antaranya:

bedaan jauh jumlah data dari setiap kategori pada variabel

Distribusi dan jumlah observasi dalam setiap kategori variabel Jenis Kontainer berpotensi mempengaruhi *p-value* yang diperoleh dari uji ANOVA. Karena ANOVA mengasumsikan bahwa sampel yang dibandingkan memiliki varian dan ukuran yang serupa. Ketika ukuran atau frekuensi sampel berbeda secara signifikan antar kelompok, hal ini dapat mempengaruhi sensitivitas tes untuk mendeteksi perbedaan.

- Asosiasi yang sangat lemah

Variabel tersebut mempunyai hubungan yang lemah dengan variabel terikatnya yaitu *Dwelling Time*. Koefisien variabelnya mungkin mendekati nol, dan variabilitas yang diamati dalam variabel terikat tidak dapat dijelaskan secara signifikan oleh perubahan dalam variabel bebas tersebut.

- Nonlinearitas

Hubungan antara variabel independen dan variabel dependen tidak linier. Jika hubungan tersebut tidak dapat ditangkap dengan baik oleh model linier, koefisiennya tidak signifikan secara statistik.

Untuk tahapan prediksi, berdasarkan hasil yang diperoleh, pada metode SVR didapatkan rata-rata error pada data *training* yaitu RMSE 1,61 dan MAE 1,26 serta pada data *testing* yaitu sebesar RMSE 1,85 dan MAE 1,32. Sedangkan pada metode ANN didapatkan rata-rata error pada data *training* yaitu RMSE 1,58 dan 1,27 serta pada data *testing* yaitu sebesar RMSE 1,76 dan MAE 1,27. Dapat dilihat error yang terjadi sangatlah besar sehingga membuat prediksinya dapat dikatakan kurang akurat.

Sebelumnya, telah dicoba skenario prediksi dengan menggunakan 3000an data dan menghasilkan error yang besar. Pada penelitian ini, seperti yang telah dijelaskan pada sub bab “4.3. Penerapan Algoritma *Artificial Neural Network*”, dilakukan penambahan data sehingga menjadi total 8181 baris setelah *preprocessing* yang mana jauh lebih banyak daripada sebelumnya. Dengan banyak data tersebut, didapatkan akurasi yang lebih sedikit dibandingkan dengan dataset yang lebih sedikit, walaupun pada matriks RMSE dan MAE menunjukkan nilai

g lebih kecil (dikarenakan rasio jumlah data dalam perhitungan galat). Hal ibuktikan banyaknya data pada kasus prediksi ini tidak cukup aruhi kenaikan akurasi prediksi.



Jika dilihat dari hasil *training* metode SVR dan ANN, terlihat pada tahapan *testing* data menggunakan data yang telah ada, metode SVR memiliki lebih sedikit error daripada metode ANN. Berdasarkan sumber yang dibaca, hal ini dikarenakan jika dataset memiliki noise atau outliers, SVR bisa lebih menggeneralisir dataset dibandingkan ANN. Kemudian juga walaupun dataset yang ada memiliki jumlah 8181 data, namun dataset yang dimiliki masih dikatakan kurang banyak untuk memprediksi dataset yang memiliki pola berantakan. Pada hal ini, SVR akan bekerja lebih baik pada dataset dengan jumlah sedikit dibandingkan ANN. Ada beberapa hal yang membuat SVR bisa lebih bekerja lebih baik dalam tahap ini, antara lain:

- Kesederhanaan model: Kesederhanaan model dan pemfokusan pada pencarian margin yang optimal pada SVR dapat menghasilkan generalisasi yang lebih baik (tidak terlalu rentan terhadap kompleksitas data), terutama ketika kumpulan datanya kecil atau terdapat *noise*. Hal ini dikarenakan SVR hanya mengambil sebagian dari data pelatihan (*support vector*) yang berada pada batas margin ( $\epsilon$ ). SVR berfokus pada menangkap pola-pola penting dalam data tanpa terlalu dipengaruhi oleh *outlier* atau *noise* (Ottens, 2024).
- Regularisasi: SVR biasanya menggunakan teknik regularisasi, seperti penggunaan parameter regularisasi ( $C$ ) dan fungsi kernel, untuk mengontrol kompleksitas model dan mencegah *overfitting*, dengan memberikan penalti pada koefisien yang besar dalam fungsi keputusan, sehingga mendorong batasan keputusan yang lebih mulus (Salaudin dkk., 2023).

Sedangkan pada model *testing* pada data baru, error pada ANN lebih sedikit dibandingkan SVR, yang berarti ANN dapat memprediksi lebih baik pada data baru yang belum pernah dilihat sebelumnya. Hal ini dikarenakan ANN lebih mampu beradaptasi dalam menangani kompleksitas dan ketidak-linearitas data yang baru dibandingkan SVR. Ada beberapa hal yang membuat ANN bisa lebih bekerja lebih baik pada tahap ini, antara lain:

- Fungsi aktivasi non-linear: ANN menggunakan fungsi aktivasi non-linear seperti *ReLU*, *sigmoid*, *tanh*) di dalam lapisannya, sehingga memungkinkan untuk beradaptasi serta menangkap hubungan kompleks dan non-linear antara label masukan dan keluaran pada data baru, terutama jika data baru



memiliki pola yang belum terwakili dengan baik saat pelatihan, sehingga bekerja lebih baik dibandingkan dengan SVR yang mengandalkan batasan keputusan linier dengan *hyperplane* tunggal (Baheti, 2021).

- Adaptabilitas: ANN menggunakan algoritma optimasi berulang, seperti *backpropagation*, untuk menyesuaikan bobot/*weight* ( $w$ ) dan *bias* ( $b$ ) jaringan selama pelatihan, memungkinkan mereka untuk terus memperbarui parameternya berdasarkan data pelatihan agar lebih sesuai dengan kompleksitas dan kumpulan data baru yang ditemui (Baheti, 2021).

Dataset yang ada juga memiliki banyak anomali seperti ketidak-konsistenan pola kombinasi data, ketidak-teraturan/keserampangan data, ketidak-merataan distribusi data, dan banyaknya itemset yang masing-masingnya hanya memiliki sedikit data. Salah satu contoh akibat ketidak-konsistenan data seperti adanya 2 kombinasi parameter dengan nilai yang sama namun memiliki *dwelling time* yang sangat jauh berbeda (0,8 dan 5,56 hari) dikarenakan perbedaan lamanya administrasi antara kedua data *dwelling time* yang tidak direkam oleh pihak perusahaan. Hal ini juga sangat mempengaruhi hasil prediksi walaupun telah dilakukan *preprocessing* sedemikian rupa untuk mengurangi anomali-anomali yang ada agar errornya berkurang.

Untuk itu, berdasarkan hasil prediksi yang ada, kasus dataset pada penelitian ini belum mampu menghasilkan prediksi *dwelling time* yang akurat. Dan hasil prediksi akan semakin bagus jika semakin banyak anomali yang dibersihkan.

#### 4.4.2 Analisis Perbandingan dengan Penelitian Terkait

Jika dibandingkan dengan penelitian terkait, terdapat dua penelitian yang juga memprediksi *dwelling time*, walaupun memiliki nilai galat yang masih cukup besar. Pada penelitian pertama yang berjudul “*Development of Models Predicting the Dwell Time of Containers in Port Container Terminals*” oleh Kourouniotti dkk., hasil prediksi menggunakan metode ANN mendapatkan akurasi sebesar 65,17%.

Berdasarkan analisis dari sumber yang dibaca, hal ini disebabkan karena penelitian



memiliki dua variabel yang tidak dimiliki penelitian ini yaitu, tanggal an dan keberangkatan kapal serta tanggal *custom inspection*. Kedua ni berbanding lurus dengan *dwelling time* karena sama-sama merupakan

satuan waktu. Selain itu penelitian ini juga memprediksi *dwelling time* dengan mengubah nilai *dwelling time* nya menjadi kategori waktu pada metode ANN, yaitu kategori waktu 1 hingga 10 hari, Dimana hal ini dapat mempengaruhi akurasi prediksi karena kompleksitas dari nilai nilai kontinu (nilai regresi) lebih tinggi daripada nilai kategorik. Rentang nilai dari variabel yang diprediksi menjadi semakin luas, misalnya sistem memprediksi *dwelling time* 3,2 sedangkan nilai yang seharusnya adalah 3,9. Pada regresi jelas terdapat galat yang cukup besar sedangkan pada klasifikasi, prediksi dianggap benar karena sama-sama dalam kategori 3 *dwelling time*. Sebagai tambahan, klasifikasi kurang sensitif terhadap kesalahan/galat dibandingkan dengan regresi.

Selanjutnya adalah penelitian kedua yang berjudul “Analisis Prediksi *Dwell Time of Import Containers Menggunakan Artificial Neural Network* pada Terminal Petikemas Semarang” oleh Ana Shohibul Manshur. Penelitian ini mendapatkan akurasi sebesar 87,64%. Sama halnya seperti penelitian pertama, penelitian ini memiliki variabel dengan satuan waktu yaitu hari bongkar, bulan bongkar dimana keduanya memiliki nilai *importance* yang lebih besar dibandingkan variabel lain yang dimiliki. Hasil prediksinya lebih baik daripada penelitian pertama, karena penelitian ini memiliki hari dan bulan “bongkar” yang berkaitan langsung dengan *dwelling time*. Waktu pembongkaran sangat berpengaruh terhadap *dwelling time* karena umumnya periode tersebut yang membuat *dwelling time* menjadi lebih lama maupun lebih cepat. Misalnya pelabuhan tersebut seringkali melakukan pembongkaran terjadwal pada hari rabu, kamis, dan jum’at, maka kontainer yang datang pada hari senin harus menunggu dua hari untuk dibongkar sehingga *dwelling time* nya jadi lebih lama dibandingkan kontainer yang datang pada hari rabu. Selain itu juga, hari bongkar tipenya adalah kategorik (7 kategori hari), sehingga pola *dwelling time* akan lebih terlihat, seperti hari apa yang paling sering membuat *dwelling time* jadi lebih lambat atau cepat.

Penelitian ini belum bisa memprediksi *dwelling time* seperti dua penelitian terkait. Diperlukan penelitian lebih lanjut seperti mengkaji ulang sistem an kontainer di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar patokan terhadap hasil analisis penelitian ini untuk dijadikan evaluasi, kedepannya dihasilkan sistem pengelolaan yang lebih terkontrol dengan



data yang lebih konsisten polanya. Diperlukan juga perekaman data sistem pengelolaan lain yang dapat membantu untuk evaluasi operator terminal dengan teknologi prediksi (*forecasting*) untuk dapat mengelola sumber daya secara efisien dan menerapkan kebijakan yang tepat.



## BAB 5. KESIMPULAN DAN SARAN

### 5.1 Kesimpulan

1. Penelitian menggunakan beberapa parameter yang diolah untuk dapat memprediksi *dwelling time*, antara lain Pelabuhan Asal, Status Kontainer, Komoditas, Jenis Kontainer, Ukuran Kontainer, dan *Gross*. Dataset dari parameter-parameter yang ada memiliki kualitas yang buruk dikarenakan terdapat banyak anomali, sehingga dilakukan beberapa tahap *preprocessing* untuk mengoptimalkan prediksi. Tahapan-tahapan awal yang dilakukan untuk memprediksi *dwelling time* antara lain input data, data *selection*, data *integration*, data *cleaning*, *outlier handling*, serta data *transformation*. Sedangkan tahapan saat menggunakan algoritma *Support Vector Regression* (SVR) maupun *Artificial Neural Network* (ANN) antara lain *feature scaling*, pembagian data *training* dan data testing, *tuning* parameter, data *training*, data *testing*, serta evaluasi model dengan data baru. Semua tahapan dilakukan untuk mendapatkan hasil prediksi yang maksimal.
2. Penerapan *forecasting* pada penelitian ini menggunakan algoritma SVR dan ANN dalam melakukan pelatihan dataset. Modelnya dipakai untuk memprediksi *dwelling time* kontainer di Terminal Petikemas Pelabuhan Soekano-Hatta pada evaluasi model. Setelah dilakukan pengolahan dataset untuk *dwelling time* yang ada, pada evaluasi model dengan data baru, didapatkan RMSE 1,85 dan MAE 1,3 pada SVR serta RMSE 1,76 dan MAE 1,27 pada ANN. Error yang didapatkan terbilang sangat besar karena kualitas data yang buruk untuk dijadikan fitur prediksi. Jika melihat hasil prediksi dengan data baru, ANN menghasilkan nilai yang lebih baik. Hal ini dikarenakan ANN lebih mampu beradaptasi dalam menangani kompleksitas dan ketidak-linearitas data yang baru dibandingkan SVR.
3. Terdapat banyak anomali yang sangat mempengaruhi pelatihan data untuk membaca pola data. Hal ini membuktikan data yang ada tidak mampu untuk memprediksi *dwelling time* dan parameter-parameter yang diambil tidak dapat dipakai untuk memprediksi *dwelling time*. Berdasarkan wawancara langsung dengan pihak perusahaan, ketidak-teraturan data pada



Terminal Petikemas Makassar dikarenakan faktor lain di luar data-data yang terekam (parameter-parameter yang diambil), seperti lamanya administrasi maupun perbedaan lama pengelolaan masing-masing kontainer. Sayangnya data mengenai hal-hal di luar sistem tidak direkam oleh pihak perusahaan karena dianggap tidak diperlukan untuk sementara waktu ini. Selain itu, berdasarkan hasil uji ANOVA, variabel Jenis Kontainer memiliki taraf tidak signifikan, sehingga Terminal Petikemas Makassar perlu mengkaji kembali variabel tersebut, apakah ada masalah pada perekaman data atau sistemnya.

## 5.2 Saran

1. Menggunakan dataset yang jauh lebih banyak dan diharapkan pola data semakin terbaca oleh algoritma sehingga dapat mengurangi error dan meningkatkan akurasi dari prediksi.
2. Pengembangan selanjutnya dapat menggunakan algoritma lainnya untuk prediksi agar dapat dijadikan perbandingan hasil antara algoritma sehingga dapat menambah pengetahuan dan memperoleh algoritma yang lebih baik untuk diterapkan.
3. Menambah parameter penelitian seperti lama administrasi dan pengelolaan kontainer.
4. Memperbaiki sistem pada pengelolaan kontainer di Terminal Petikemas Pelabuhan Soekarno Hatta Makassar, agar data yang dihasilkan lebih rapi dan stabil sehingga dapat membantu evaluasi sistem pengelolaan kedepannya.



## DAFTAR PUSTAKA

- Afiffudin, S. (2019). Penanganan Petikemas Pada Pt. Salam Pacific Indonesia Lines (Spil) Cabang Manokwari Papua Barat. Karya Tulis.
- Ahmad, A. (2017). Mengenal artificial intelligence, machine learning, neural network, dan deep learning. *J. Teknol. Indones.*, no. October, 3.
- Ali Ikhwan, S. D. (2015). Penerapan Data Mining dengan Algoritma Fp-Growth untuk Mendukung Strategi Promosi Pendidikan (Studi Kasus Kampus STMIK Triguna Dharma). *Jurnal Ilmiah Saindikom*, 211-226.
- Ali, P. J. M., Faraj, R. H., Koya, E., Ali, P. J. M., & Faraj, R. H. (2014). Data normalization and standardization: a technical report. *Mach Learn Tech Rep*, 1(1), 1-6.
- Amei, W., Huailin, D., Qingfeng, W., & Ling, L. (2011, November). A survey of application-level protocol identification based on machine learning. *In 2011 International Conference on Information Management, Innovation Management and Industrial Engineering* (Vol. 3, pp. 201-204). IEEE.
- Anggia, P. N. (2019). Kinerja Pt. Nilam Port Terminal Indonesia Dalam Memperlancar Kegiatan Bongkar Muat Di Pelabuhan Tanjung Perak Surabaya. Karya Tulis.
- Anggraeni, R. M. (2014). Perbandingan Algoritma Apriori dan Algoritma FP-Growth untuk Rekomendasi Pada Transaksi Peminjaman Buku di Perpustakaan Universitas Dian Nuswantoro. *Tek. Inform.*, 1-6.
- Assena, D. F. (2020). Model Algoritma K-Means Clustering Untuk Menentukan Strategi Promosi Universitas Islam Nahdlatul Ulama Jepara (Doctoral dissertation, UNISNU Jepara).
- Azizah, S. (2019). Analisis Prediksi Dwell Time of Import Containers Menggunakan Artificial Neural Network pada Terminal Petikemas Semarang.
- Brownlee, J. (2016). Machine learning algorithms from scratch with Python. *Machine Learning Mastery*.
- Cholissodin, I., Sutrisno, S., Soebroto, A. A., Hasanah, U., & Febiola, Y. I. (2020). AI, Machine Learning & Deep Learning. Fakultas Ilmu Komputer, Universitas Brawijaya, Malang.



ii, N., & Shawe-Taylor, J. (2000). An introduction to support vector machines and other kernel-based learning methods. Cambridge university press.

- Das, S., & Nene, M. J. (2017, Maret). A survey on types of machine learning techniques in intrusion prevention systems. In *2017 international conference on wireless communications, signal processing and networking (WiSPNET)* (pp. 2296-2299). IEEE. <https://doi.org/10.1109/WiSPNET.2017.8300169>
- Edy, N. H. (2009). Manajemen Kepelabuhanan, Seri 01, Edisi II. PT. Pelabuhan Indonesia (Persero).
- Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep learning. MIT press.
- Hsu, C.W., Chang, C.C, & Lin, C.J. (2004). A Particial Guide to Support Vector Classification. Departement of Computer Science and Information Engineering. 2(3): 1396-1400.
- Iswahyudi, C. (2016). Pengantar Forecasting (Teknik Peramalan). Stikom Bali.
- Javed, F., Chan, G. S., Savkin, A. V., Middleton, P. M., Malouf, P., Steel, Lovell, N. H. (2009). RBF kernel based support vector regression to estimate the blood volume and heart rate responses during hemodialysis. 2009 annual international conference of the IEEE engineering in medicine and biology society (pp. 4352-4355).
- Jiawei, H., Micheline, K., & Jian, P. (2011). Data Mining Concept and Techniques. Morgan Kaufmann, San Fransiso.
- Larose, D. T., & Larose, C. D. (2014). *Discovering Knowledge In Data: An Introduction To Data Mining* (Vol. 4). John Wiley & Sons.
- Lestari, Y. D. (2017). Penerapan Data Mining Menggunakan Algoritma Fp-Tree Dan Fp-Growth Pada Data Transaksi Penjualan Obat.
- Mahmud, M., Kaiser, M. S., Hussain, A., & Vassanelli, S. (2018). Applications of deep learning and reinforcement learning to biological data. *IEEE transactions on neural networks and learning systems*, 29(6), 2063-2079.
- Mardi, Y. (2017). Data Mining: Klasifikasi Menggunakan Algoritma C4. 5. Jurnal Edik Informatika Penelitian Bidang Komputer Sains dan Pendidikan Informatika, 2(2), 213-219.
- Mitchell, T. M. (1997). Machine learning. In McGraw Hill Series in Computer Science. Retrieved from <http://www.worldcat.org/oclc/61321007>.
- Nainel, Y. L., Buulolo, E., & Lubis, I. (2020). Penerapan Data Mining Untuk Estimasi Penjualan Obat Berdasarkan Pengaruh Brand Image Dengan Igoritma Expectation Maximization (Studi Kasus: PT. Pyridam Farma bk). *JURIKOM (Jurnal Riset Komputer)*, 7(2), 214-224.
- a, G., & Agustini, K. (2022). Konsep Dasar Data Mining.



- Qiang, W., & Zhongli, Z. (2011, Agustus). Reinforcement learning model, algorithms and its application. *In 2011 International Conference on Mechatronic Science, Electric Engineering and Computer (MEC)* (pp. 1143-1146). IEEE.
- Salauddin, M., Shaffrey, D., & Habib, M. A. (2023). Data-driven approaches in predicting scour depths at a vertical seawall on a permeable shingle foreshore. *Journal of Coastal Conservation*, 27(3), 18..
- Santosa, B. (2007). Data mining teknik pemanfaatan data untuk keperluan bisnis. Yogyakarta: Graha Ilmu, 978(979), 756.
- Santoso, B. (2007). Data Mining: Teknik Pemanfaatan Data Untuk Bisnis. *Teori dan Aplikasi, Graha Ilmu*.
- Somvanshi, M., Chavan, P., Tambade, S., & Shinde, S. V. (2016, Agustus). A review of machine learning techniques using decision tree and support vector machine. *In 2016 international conference on computing communication control and automation (ICCUBEA)* (pp. 1-7). IEEE.
- Spot Intelligence. (2024). Support Vector Regression (SVR). Spot Intelligence. Diakses pada 5 mei 2024 melalui <https://spotintelligence.com/2024/05/08/support-vector-regression-svr/>.
- Tahir, M., & Sitompul, N. (2021). Penerapan algoritma fp-growth dalam menentukan kecenderungan mahasiswa mengambil mata kuliah pilihan. *Network Engineering Research Operation*, 6(1), 56-63.
- Tarigan, P. M. S., Hardinata, J. T., Qurniawan, H., Safii, M., & Winanjaya, R. (2022). Implementasi Data Mining Menggunakan Algoritma Apriori Dalam Menentukan Persediaan Barang: Studi Kasus: Toko Sinar Harahap. *Jurnal Janitra Informatika dan Sistem Informasi*, 2(1), 9-19.
- Thupae, R., Isong, B., Gasela, N., & Abu-Mahfouz, A. M. (2018, Oktober). Machine learning techniques for traffic identification and classification in SDWSN: A survey. *In IECON 2018-44th annual conference of the IEEE Industrial Electronics Society* (pp. 4645-4650). IEEE. <https://doi.org/10.1109/IECON.2018.8591178>
- Triatmodjo, B. (2010). Perencanaan pelabuhan. Yogyakarta: Beta Offset.
- Utami, S. W. (2015) Percepatan Dwelling Time: Strategi Peningkatan Kinerja Perdagangan Internasional Pelabuhan Tanjung Priok. *Economics Development Analysis Journal* 4.1 (2015): 82-90.



(2023). Neural networks and activation functions. V7 Labs. Diakses pada ) Mei 2024 melalui <https://www.v7labs.com/blog/neural-networks-activation-functions>

- Veriane, S. P. Implementasi Sistem Rekomendasi pada Fitur Keranjang Menggunakan Metode Algoritma Apriori pada Aplikasi SayurMurah. (2023). Com Berbasis Website. JUSTIN (Jurnal Sistem dan Teknologi Informasi), 11(1), 113-118.
- Wahyudi, A. K., Azizah, N., & Saputro, H. (2022). Data Mining Klasifikasi Kepribadian Siswa Smp Negeri 5 Jepara Menggunakan Metode Decision Tree Algoritma C4. 5. *Journal of Information System and Computer*, 2(2), 8-13.
- Yasin, H., Prahutama, A., & Utami, T.W. (2014). Prediksi Harga Saham Menggunakan Support Vector Regression dengan Algoritma Grid Search. *Media Statistika*. 7(1): 29-35.
- Yunitasari, M., Maharani, T., Hikmahwan, B., Komunitas, A., Pacitan, N., & No, W. M. (2022). Implementasi Metode K-Mean untuk Pengelompokan Data Jamaah. *Kumpulan Jurnal Ilmu Komputer*, 9(1), 1-9.



**Lampiran 1** Dataset

<https://drive.google.com/drive/folders/1Tf-xPOxUNebFIJE6-KRuUGE7Qg58AzYn?usp=sharing>



Optimized using  
trial version  
[www.balesio.com](http://www.balesio.com)

## Lampiran 2 Source Code

### a. Preprocessing

```

import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error
import os

from google.colab import drive
drive.mount('/content/drive')

# List file Excel yang akan digabungkan
# Buat list untuk menyimpan data Excel yang akan digabungkan
file_names1 = ['/content/drive/MyDrive/DATASET/8. NOVEMBER
2022/Inbound List_00-11-2022-CTPH HONOUR.xlsx',
               '/content/drive/MyDrive/DATASET/8. NOVEMBER
2022/Inbound List_00-11-2022-MERATUS AMPANA.xlsx',
               '/content/drive/MyDrive/DATASET/8. NOVEMBER
2022/Inbound List_00-11-2022-TANTO.xlsx',
               '/content/drive/MyDrive/DATASET/7. DESEMBER
2022/Inbound List_03-12-2022-MERATUS MANADO.xlsx',
               '/content/drive/MyDrive/DATASET/7. DESEMBER
2022/Inbound List_13-12-2022-MV TANTO TERANG.xlsx',
               '/content/drive/MyDrive/DATASET/7. DESEMBER
2022/Inbound List_29-12-2023-CTP HONOUR.xlsx',
               '/content/drive/MyDrive/DATASET/1. JANUARI
2023/Inbound List_05-01-2023-MERATUS MEDAN 3.xlsx',
               '/content/drive/MyDrive/DATASET/1. JANUARI
2023/Inbound List_06-01-2023-CTP HONOUR.xlsx',
               '/content/drive/MyDrive/DATASET/1. JANUARI
2023/Inbound List_10-01-2023-TANTO TERANG.xlsx',
               '/content/drive/MyDrive/DATASET/2. FEBRUARI
2023/Inbound List_03-02-2023-MERATUS MEDAN 3.xlsx',
               '/content/drive/MyDrive/DATASET/2. FEBRUARI
2023/Inbound List_07-02-2023-TANTO TERANG.xlsx',
               '/content/drive/MyDrive/DATASET/2. FEBRUARI
2023/Inbound List_08-02-2023-CTP HONOUR.xlsx',
               '/content/drive/MyDrive/DATASET/3. MARET
nbound List_04-03-2023-MERATUS PROJECT 3.xlsx',
               '/content/drive/MyDrive/DATASET/3. MARET
nbound List_09-03-2023-CTP HONOUR.xlsx',
               '/content/drive/MyDrive/DATASET/3. MARET
nbound List_29-03-2023-TANTO TERANG.xlsx',

```



```

        '/content/drive/MyDrive/DATASET/4. APRIL
2023/Inbound List_03-04-2023-CTP HONOUR.xlsx',
        '/content/drive/MyDrive/DATASET/4. APRIL
2023/Inbound List_06-04-2023-MERATUS MEDAN 2.xlsx',
        '/content/drive/MyDrive/DATASET/4. APRIL
2023/Inbound List_09-04-2023-TANTO TERANG.xlsx',
        '/content/drive/MyDrive/DATASET/5. MEI
2023/Inbound List_10-05-2023-CTP HONOUR.xlsx',
        '/content/drive/MyDrive/DATASET/5. MEI
2023/Inbound List_20-05-2023-TANTO TERANG.xlsx',
        '/content/drive/MyDrive/DATASET/5. MEI
2023/Inbound List_24-05-2023-MERATUS MEDAN 3.xlsx'
    ]

# Membaca setiap file Excel dan mengubahnya menjadi DataFrame
data_frames = [pd.read_excel(file) for file in file_names1]

# Menggabungkan DataFrame secara vertical
combined_df_inbound = pd.concat(data_frames, axis=0)

# Buat list untuk menyimpan data Excel yang akan digabungkan
file_names2 = ['/content/drive/MyDrive/DATASET/8. NOVEMBER
2022/Terminal In & Out_00-11-2022-CTPH HONOUR.xlsx',
                '/content/drive/MyDrive/DATASET/8. NOVEMBER
2022/Terminal In & Out_00-11-2022-MERATUS AMPANA.xlsx',
                '/content/drive/MyDrive/DATASET/8. NOVEMBER
2022/Terminal In & Out_00-11-2022-TANTO.xlsx',
                '/content/drive/MyDrive/DATASET/7. DESEMBER
2022/Terminal In & Out_03-12-2022-MERATUS MANADO.xlsx',
                '/content/drive/MyDrive/DATASET/7. DESEMBER
2022/Terminal In & Out_13-12-2022-MV TANTO TERANG.xlsx',
                '/content/drive/MyDrive/DATASET/7. DESEMBER
2022/Terminal In & Out_29-12-2023-CTP HONOUR.xlsx',
                '/content/drive/MyDrive/DATASET/1. JANUARI
2023/Terminal In & Out_05-01-2023-MERATUS MEDAN 3.xlsx',
                '/content/drive/MyDrive/DATASET/1. JANUARI
2023/Terminal In & Out_06-01-2023-CTP HONOUR.xlsx',
                '/content/drive/MyDrive/DATASET/1. JANUARI
2023/Terminal In & Out_10-01-2023-TANTO TERANG.xlsx',
                '/content/drive/MyDrive/DATASET/2. FEBRUARI
2023/Terminal In & Out_03-02-2023-MERATUS MEDAN 3.xlsx',
                '/content/drive/MyDrive/DATASET/2. FEBRUARI
2023/Terminal In & Out_07-02-2023-TANTO TERANG.xlsx',
                '/content/drive/MyDrive/DATASET/2. FEBRUARI
erminal In & Out_08-02-2023-CTP HONOUR.xlsx',
                '/content/drive/MyDrive/DATASET/3. MARET
erminal In & Out_04-03-2023-MERATUS PROJECT 3.xlsx',
                '/content/drive/MyDrive/DATASET/3. MARET
erminal In & Out_09-03-2023-CTP HONOUR.xlsx',

```



```

        '/content/drive/MyDrive/DATASET/3. MARET
2023/Terminal In & Out_29-03-2023-TANTO TERANG.xlsx',
        '/content/drive/MyDrive/DATASET/4. APRIL
2023/Terminal In & Out_03-04-2023-CTP HONOUR.xlsx',
        '/content/drive/MyDrive/DATASET/4. APRIL
2023/Terminal In & Out_06-04-2023-MERATUS MEDAN 2.xlsx',
        '/content/drive/MyDrive/DATASET/4. APRIL
2023/Terminal In & Out_09-04-2023-TANTO TERANG.xlsx',
        '/content/drive/MyDrive/DATASET/5. MEI
2023/Terminal In & Out_10-05-2023-CTP HONOUR.xlsx',
        '/content/drive/MyDrive/DATASET/5. MEI
2023/Terminal In & Out_20-05-2023-TANTO TERANG.xlsx',
        '/content/drive/MyDrive/DATASET/5. MEI
2023/Terminal In & Out_24-05-2023-MERATUS MEDAN 3.xlsx'
    ]

# Membaca setiap file Excel dan mengubahnya menjadi DataFrame
data_frames = [pd.read_excel(file) for file in file_names2]

# Menggabungkan DataFrame secara vertical
combined_df_inOut = pd.concat(data_frames, axis=0)

combined_df1 = combined_df_inbound.drop(['No.', 'Stowage',
'Op.Type', 'DS Plan', 'POW',
                                     'Position', 'Class', 'Out
Crr.Type', 'Out Carrier',
                                     'Thru', 'Direct', 'OPR',
'POR', 'POD', 'NPOD', 'FPOD',
                                     'FDEST', 'B/L', 'LEN', 'WID',
'HGT', 'Type',
                                     'VGM', 'Restow Type', 'Restow
STWG', 'RF', 'DG', 'OOG',
                                     'Temp.', 'C/F', 'UNNO',
'IMDG', 'OD F', 'OD B', 'OD L',
                                     'OD R', 'OD H', 'Comb.', 'Sub
STWG', 'TML Unit ID',
                                     'Hatch IDX', 'Bay IDX', 'Row
IDX', 'Tier IDX', 'Block', 'Bill To', '2nd Bill To', 'ISO' ],
axis =1)

combined_df2 = combined_df_inOut[['Cargo', 'Length', 'Dwelling
Time']].copy()

```



```

mukan nilai yang tidak sama di kolom "Cargo"
ent_cargos = set(combined_df1['Cargo']) ^
mbined_df2['Cargo'])

```

```

mpilkan nilai yang tidak sama di kolom "Cargo"
"Nilai yang tidak sama di kolom 'Cargo: ")

```

```

for cargo in different_cargos:
    print(cargo)

# Memeriksa isi kolom "Cargo" dari kedua data frame
cargo_df1 = set(combined_df1['Cargo'])
cargo_df2 = set(combined_df2['Cargo'])

# Memeriksa perbedaan dalam isi kolom "Cargo"
diff_cargo = cargo_df1.symmetric_difference(cargo_df2)
if len(diff_cargo) == 0:
    print("Kolom 'Cargo' pada kedua data frame memiliki isi yang sama.")
else:
    print("Perbedaan dalam isi kolom 'Cargo':", diff_cargo)

# Menemukan posisi kolom "Cargo" dalam kedua data frame
position_df1 = combined_df1.columns.get_loc('Cargo')
position_df2 = combined_df2.columns.get_loc('Cargo')

print("Posisi kolom 'Cargo' dalam data frame pertama:", position_df1)
print("Posisi kolom 'Cargo' dalam data frame kedua:", position_df2)

# Memeriksa nilai NaN atau null pada kedua data frame
nan_values_df1 = combined_df1.isnull().any()
nan_values_df2 = combined_df2.isnull().any()

# Menampilkan kolom dengan nilai NaN atau null pada data frame pertama
nan_columns_df1 = nan_values_df1[nan_values_df1].index
print("Kolom dengan nilai NaN atau null pada data frame pertama:", nan_columns_df1.tolist())

# Menampilkan kolom dengan nilai NaN atau null pada data frame kedua
nan_columns_df2 = nan_values_df2[nan_values_df2].index
print("Kolom dengan nilai NaN atau null pada data frame kedua:", nan_columns_df2.tolist())

# Mencari posisi nilai NaN pada kedua data frame
nan_positions_df1 = np.where(pd.isnull(combined_df1))
nan_positions_df2 = np.where(pd.isnull(combined_df2))

```



```

# Menampilkan posisi nilai NaN pada data frame pertama
print("Jumlah kolom kosong di data frame pertama : ",
      nan_positions_df1[0].shape)
# Menampilkan posisi nilai NaN pada data frame kedua
print("Jumlah kolom kosong di data frame kedua : ",
      nan_positions_df2[0].shape)

```

```

print("Posisi nilai NaN pada data frame pertama:")
for row, col in zip(nan_positions_df1[0],
nan_positions_df1[1]):
    print("Baris:", row, "Kolom:", col)

# Menampilkan posisi nilai NaN pada data frame kedua
print("\nPosisi nilai NaN pada data frame kedua:")
for row, col in zip(nan_positions_df2[0],
nan_positions_df2[1]):
    print("Baris:", row, "Kolom:", col)

# Mengecek baris dengan keseluruhan isinya kosong
empty_rows1 = combined_df1[combined_df1.isnull().all(axis=1)]
empty_rows2 = combined_df2[combined_df2.isnull().all(axis=1)]

# Mengambil nilai dari masing-masing kolom 'Cargo'
df1_cargo_values = combined_df1['Cargo'].values
df2_cargo_values = combined_df2['Cargo'].values

# Membuat data frame terpisah dengan dua kolom 'Cargo' dari
masing-masing data frame
df_combined = pd.DataFrame({
    'df1_Cargo': df1_cargo_values,
    'df2_Cargo': df2_cargo_values
})

# Cek apakah terdapat baris dengan isi nilai 'Cargo' yang
berbeda
different_cargo_rows = df_combined[df_combined['df1_Cargo']
!= df_combined['df2_Cargo']]

# Menampilkan hasil
print("Data frame kombinasi dengan dua kolom 'Cargo':")
print(df_combined)

if different_cargo_rows.empty:
    print("Tidak ada baris dengan isi nilai 'Cargo' yang
berbeda.")
else:
    print("Baris dengan isi nilai 'Cargo' yang berbeda:")
    print(different_cargo_rows)

combined_df1_index =
    ed_df1.reset_index().rename(columns={'Cargo': 'id'})
ed_df2_index =
    ed_df2.reset_index().rename(columns={'Cargo': 'id'})

dataset = pd.concat([combined_df1_index,
ed_df2_index], axis=1)

```



```

merge_fix_data =
merge_dataset.drop(merge_dataset.columns[[0,1,7,8]], axis =1)
# Menyimpan hasil gabungan ke dalam file csv dan excel
merge_fix_data.to_excel('merge_fix_data.xlsx', index=False)
merge_fix_data.to_csv('merge_fix_data.csv', index=False)

# Mencari posisi nilai NaN pada kedua data frame
nan_positions_mdf1 = np.where(pd.isnull(merge_fix_data))

# Menampilkan posisi nilai NaN pada data frame pertama
print("Posisi nilai NaN pada data frame pertama:")
for row, col in zip(nan_positions_mdf1[0],
nan_positions_mdf1[1]):
    print("Baris:", row, "Kolom:", col)

# Menampilkan semua isi kolom 'Cargo' jika terdapat nilai
null atau NaN pada kolom 'Dwelling Time'
cargo_with_null_dwelling =
merge_fix_data.loc[merge_fix_data['Dwelling Time'].isnull()]
print("Posisi baris 'Dwelling Time' yang kosong:")
print(cargo_with_null_dwelling)

# Menampilkan semua isi kolom 'Cargo' jika terdapat nilai
null atau NaN pada kolom 'GW'
cargo_with_null_dwelling =
merge_fix_data.loc[merge_fix_data['GW'].isnull()]
print("Posisi baris 'GW' yang kosong:")
print(cargo_with_null_dwelling)

'''Setelah di cek pada data outliers pada GW, diketahui bahwa
maksimum daya angkut kontainer 20 adalah berkisar diangkan 20
ton sedangkkn berat 218800 setara dengan 241 ton. Maka dari
itu diasumsikan ada kelebihan nol pada saat input data'''
#Ubah nilai oulier pada GW
#Menyaring data frame berdasarkan kolom GW
df_filtered = merge_fix_data[merge_fix_data['GW'] ==
218800.0].copy()

# Mengubah nilai outlier
df_filtered.loc[:, 'GW'] = 21880.0

# Menggabungkan data frame hasil perubahan dengan data frame
...
_p01 = pd.concat([merge_fix_data[merge_fix_data['GW']
800.0], df_filtered])
_p01.head()

```



```

'''Karena data GW bergantung pada Length, maka pengisian
nilai NaN pada Gw akan bergantung pada Length masing-masing.
Data NaN pada GW akan diisi dengan nilai rata-rata pada tiap
kategori length, karena data gross terdistribusi normal
'''
# Menghitung rata-rata GW untuk setiap kategori Length
mean_gw = df_pre_po1.groupby('Length')['GW'].mean()
print(mean_gw)

# Mengecek baris dengan nilai GW dan DW kosong
missing_rows = df_pre_po1[df_pre_po1['GW'].isnull() &
df_pre_po1['Dwelling Time'].isnull()]
print(missing_rows)

# Menghapus baris dengan nilai GW dan Dwelling Time kosong
secara bersamaan
df_pre_po2 = df_pre_po1.dropna(subset=['GW', 'Dwelling
Time'], how='all')

# Mengisi nilai GW yang kosong dengan rata-rata tiap jenis
Length
df_pre_po2['GW'] =
df_pre_po2['GW'].fillna(df_pre_po2['Length'].map(mean_gw))

# df_pre_po2.dropna(subset = ['GW'], inplace=True)

# Menampilkan data setelah pengisian nilai GW kosong
print(df_pre_po2)

# Membuat DataFrame baru dengan kolom 'Dwelling Time'
df_pre_pro3 = df_pre_po2.copy() # Membuat salinan DataFrame
awal
# df_pre_pro3 = df_pre_pro11.copy() # Membuat salinan
DataFrame awal
# median_dwelling_time = df_pre_po2['Dwelling Time'].median()
# df_pre_pro3['Dwelling Time'] = df_pre_po2['Dwelling
Time'].fillna(median_dwelling_time)

# untuk prepro 6 dan 10, nan dwelling akan dihilangkan
df_pre_pro3.dropna(subset = ['Dwelling Time'], inplace=True)

df_pre_pro6 = df_pre_pro3.copy()

```



```

hilangkan data yang memuat dwelling time diatas 11,9
_pro10 = df_pre_pro6[df_pre_pro6['Dwelling Time'] <=
df_pre_pro10)

```

```

simpan hasil preprocessing ke dalam file csv dan excel

```

```
df_pre_pro10.to_excel('df_pre_pro10.xlsx', index=False)
df_pre_pro10.to_csv('df_pre_pro10.csv', index=False)
```

## b. Preprocessing Tambahan untuk Forecasting

```
import numpy as np
import pandas as pd
from google.colab import data_table
data_table.enable_dataframe_formatter()

df = pd.read_csv(r'/content/df_pre_pro10.csv')

# Mengecek keberadaan nilai null pada seluruh dataframe
null_df = df.isna()
print(null_df)

# Mengecek posisi nilai null pada setiap kolom
for col in null_df.columns:
    if null_df[col].any():
        print(f"Kolom {col} memiliki nilai null pada index
berikut:")
        print(null_df.index[null_df[col]])

df = df.dropna()

# Menghilangkan leading spaces dari nama kolom

df.columns = df.columns.str.strip()

# Rename nama kolom
df.columns = ['Pelabuhan Asal', 'Status Kontainer',
'Komoditas', 'Jenis Kontainer', 'Gross',
'Ukuran Kontainer', 'Dwelling Time', ]

#Cek outlier DWELLING TIME
import matplotlib.pyplot as plt
import seaborn as sns

kolom = 'Dwelling Time'

# Buat boxplot
sns.boxplot(x=df[kolom])
```



```
ng batas bawah dan atas data outliers
f[kolom].quantile(0.25)
f[kolom].quantile(0.75)
q3 - q1
bawah = q1 - 1.5 * iqr
```

```

batas_atas = q3 + 1.5 * iqr

# Cari indeks baris yang mengandung data outliers
outlier_idx = df[(df[kolom] < batas_bawah) | (df[kolom] >
batas_atas)].index

# Tampilkan indeks baris yang mengandung data outliers
print("Data outliers pada kolom", kolom, "ada pada baris
ke:")
print(outlier_idx)
print("Q1 :", q1)
print("Q3 :", q3)
print("iqr :", iqr)

# Tampilkan grafik
plt.show()

#Hapus semua baris data outlier pada Dwelling Time
import pandas as pd

# Muat data
df = df

# Hitung IQR
Q1 = df['Dwelling Time'].quantile(0.25)
Q3 = df['Dwelling Time'].quantile(0.75)
IQR = Q3 - Q1

# Menentukan batasan bawah (lower bound) dan atas (upper
bound) untuk deteksi outlier
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Menghapus baris yang memiliki nilai outlier di 'Dwelling
Time'
df_clean = df[(df['Dwelling Time'] >= lower_bound) &
(df['Dwelling Time'] <= upper_bound)]

# Reset index
df_clean = df_clean.reset_index(drop=True)

# Print DataFrame yang telah dibersihkan
df_clean.head(10)

```



nganan Kedua  
s semua baris data outlier pada Dwelling Time  
pandas as pd

data

```

df = df_clean

# Hitung IQR
Q1 = df_clean['Dwelling Time'].quantile(0.25)
Q3 = df_clean['Dwelling Time'].quantile(0.75)
IQR = Q3 - Q1

# Menentukan batasan bawah (lower bound) dan atas (upper
bound) untuk deteksi outlier
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

# Menghapus baris yang memiliki nilai outlier di 'Dwelling
Time' pada penanganan kedua
df_clean2 = df_clean[(df_clean['Dwelling Time'] >=
lower_bound) & (df_clean['Dwelling Time'] <= upper_bound)]

# Reset index
df_clean2 = df_clean2.reset_index(drop=True)

# Print DataFrame yang telah dibersihkan
df_clean2.head(10)

df = df_clean2

# Mentransformasikan data ke tipe numerik dengan one hot
encoding
one_hot_encoded_data = pd.get_dummies(df, columns =
['Pelabuhan Asal', 'Status Kontainer', 'Komoditas', 'Jenis
Kontainer'])
print(one_hot_encoded_data)

# Import libraries
from sklearn.preprocessing import OneHotEncoder

# Mengkonversu tipe kolol ke kategori
df['Pelabuhan Asal']=df['Pelabuhan Asal'].astype('category')
df['Status Kontainer']=df['Status
Kontainer'].astype('category')
df['Komoditas']=df['Komoditas'].astype('category')
df['Jenis Kontainer']=df['Jenis
Kontainer'].astype('category')

```



```

tapkan nilai numerik dan menyimpannya di kolom lain
labuhan Asal new']=df['Pelabuhan Asal'].cat.codes
atus Kontainer new']=df['Status Kontainer'].cat.codes
noditas new']=df['Komoditas'].cat.codes
nis Kontainer new']=df['Jenis Kontainer'].cat.codes

```

```

# Membuat sebuah instance dari One-hot-encoder
enc=OneHotEncoder()

# Passing kolom yang ter-encode
enc_data=pd.DataFrame(enc.fit_transform(df[['Pelabuhan Asal
new', 'Status Kontainer new', 'Komoditas new', 'Jenis
Kontainer new']]).toarray())

# Merge dengan data utama
new_df=df.join(enc_data)
#new_df = data.join([str(i) for i in enc_data])
#values = ','.join([unicode(i) for i in value_list])

# Memilih variabel yang akan diakan dipakai
df = new_df[['Pelabuhan Asal new', 'Status Kontainer new',
'Komoditas new', 'Jenis Kontainer new', 'Ukuran Kontainer',
'Gross', 'Dwelling Time']]

# Rename kolom
df.columns = ['Pelabuhan Asal', 'Status Kontainer',
'Komoditas', 'Jenis Kontainer',
'Ukuran Kontainer', 'Gross', 'Dwelling Time']

df = df.dropna()

# Menyimpan hasil preprocessing ke dalam file csv dan excel
df.to_excel('dataset_prediction_prepo101010.xlsx',
index=False)
df.to_csv('dataset_prediction_prepo101010.csv', index=False)

```

### c. Prediksi dengan SVR

```

import pandas as pd

data =
pd.read_csv('/content/dataset_prediction_prepo101010.csv')

import pandas as pd
from sklearn.model_selection import train_test_split,
GridSearchCV
from sklearn.svm import SVR
from sklearn.preprocessing import StandardScaler
sklearn.metrics import mean_squared_error
sklearn.pipeline import make_pipeline

data
ata # Ganti dengan nama file dan lokasi yang sesuai

```



```

# Membagi dataset menjadi data prediktor (X) dan data target
(y)
X_original = df.drop('Dwelling Time', axis=1)
y_original = df['Dwelling Time']

# Standarisasi/scaling feature
scaler = StandardScaler()
X_original_scaled = scaler.fit_transform(X_original)

# Membuat DataFrame dengan scaled features dan nama kolom
original
X_original_scaled_df = pd.DataFrame(X_original_scaled,
columns=X_original.columns)

# Membagi data menjadi data training dan data testing
X_train, X_test, y_train, y_test =
train_test_split(X_original_scaled_df, y_original,
test_size=0.2, random_state=42)

# Membuat model SVR
svr_model = SVR()

# Menentukan grid parameter untuk grid search
param_grid = {
    'kernel': ['linear', 'rbf', 'poly'],
    'C': [0.1, 1, 10],
    'epsilon': [0.1, 0.2, 0.5]
}

# Mengeksekusi grid search dengan cross-validation
grid_search = GridSearchCV(svr_model, param_grid, cv=5,
scoring='neg_mean_squared_error', n_jobs=-1)
grid_search.fit(X_train, y_train)

# Mencetak parameter terbaik yang didapatkan oleh grid search
print("Best Parameters: ", grid_search.best_params_)

# Membuat prediksi di data testing menggunakan model terbaik
dari grid search
y_pred = grid_search.predict(X_test)

# Mengevaluasi model
mse = mean_squared_error(y_test, y_pred)
print(f'Mean Squared Error: {mse}')

# Membuat prediksi dan membandingkannya dengan data actual
y_test = y_test.to_numpy()
y_test_flat = y_test.flatten()

```



```

y_pred_flat = y_pred.flatten()

data = {'Actual (y_test)': y_test_flat, 'Predicted (y_pred)':
y_pred_flat}
prediciton_result_df = pd.DataFrame(data)

# Menyimpan prediksi hasil training ke dalam file csv dan
excel
prediciton_result_df.to_excel('svr_prediciton_result_df.xlsx'
, index=False)
prediciton_result_df.to_csv('svr_prediciton_result_df.csv',
index=False)

# Fit model menggunakan parameter terbaik (agar tidak harus
fit lagi nanti di saat testing model dengan data baru)
svr_model = make_pipeline(StandardScaler(), SVR(C=1,
epsilon=0.5, kernel = 'rbf'))
svr_model.fit(X_train, y_train)
y_pred = svr_model.predict(X_test)

# Menghitung RMSE
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f"RMSE: {rmse}")

# Menghitung MAE
mae = mean_absolute_error(y_test, y_pred)
print('MAE:', mae)

# Menghitung accuracy
accuracy = 100 - (mae / np.mean(y_test)) * 100
print('Accuracy:', accuracy)

# Menyimpan fitted model ke file
import joblib
joblib.dump(svr_model,
'prepro101010_dwelling_time_SVR_model.pkl')

# MODEL TESTING
# Memuat model
loaded_model =
joblib.load('/content/prepro101010_dwelling_time_SVR_model.pk
l')

```



```

at dataset baru
taset =
d_csv('/content/dataset_prediction_prepro101010.csv')

agi dataset barumenjadi data prediktor (X) dan data
(y)

```

```

X_new = new_dataset.drop('Dwelling Time', axis=1)
y_new = new_dataset['Dwelling Time']

# Standarisasi data predictor/features pada dataset yang baru
X_new_scaled = scaler.transform(X_new)

# Membuat DataFrame dengan scaled feature and nama kolom
original
X_new_scaled_df = pd.DataFrame(X_new_scaled,
columns=X_original.columns)

# Test model pada dataset baru
predictions = loaded_model.predict(X_new_scaled_df)

# Mengevaluasi performa model dataset baru
mse_new = mean_squared_error(y_new, predictions)
print("Mean Squared Error on New Dataset:", mse_new)

# Menghitung RMSE
rmse = np.sqrt(mean_squared_error(y_new, predictions))
print(f"RMSE: {rmse}")

# Menghitung MAE
mae = mean_absolute_error(y_new, predictions)
print('MAE:', mae)

# Menghitung accuracy
accuracy = 100 - (mae / np.mean(y_new)) * 100
print('Accuracy:', accuracy)

# Membandingkan hasil prediksi dengan data actual
y_new = y_new.to_numpy()
y_new_flat = y_new.flatten()
predictions_flat = predictions.flatten()
data = {'Actual (y_new)': y_new_flat, 'Predicted
(predictions)': predictions_flat}
predicition_testing_result_df = pd.DataFrame(data)

# Menyimpan hasil prediksi ke dalam file csv dan excel
predicition_testing_result_df.to_excel('svr_predicition_testing
_result_df.xlsx', index=False)
predicition_testing_result_df.to_csv('svr_predicition_testing_r
esult_df.csv', index=False)

```



prediksi dengan ANN

```

pandas as pd
numpy as np

```

```

from keras.models import Sequential
from keras.layers import Dense
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error

data =
pd.read_csv('/content/dataset_prediction_prep01010.csv')

# Membagi dataset menjadi data prediktor (X) dan data target
(y)X = data.drop('Dwelling Time', axis=1)
y = data['Dwelling Time']

# Standarisasi data training
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
scaled_df = pd.DataFrame(X_scaled, columns=X.columns)

from google.colab import data_table
data_table.DataTable(scaled_df, include_index=False,
num_rows_per_page=10)
standarized_df = pd.DataFrame(scaled_df)

# Membagi dataset menjadi data training dan data testing
X_train, X_test, y_train, y_test = train_test_split(X_scaled,
y, test_size=0.2, random_state=42)

# Menentukan arsitektur model
model = Sequential()
model.add(Dense(units=16, activation='relu',
input_dim=X_train.shape[1]))
model.add(Dense(units=8, activation='relu'))
model.add(Dense(units=1, activation='linear'))

# Compile model
model.compile(optimizer='adam', loss='mean_squared_error')

# Train model
model.fit(X_train, y_train, epochs=100, batch_size=32)

# Membuat prediksi pada data testing
` = model.predict(X_test)

andingkan hasil prediksi dengan data aktual
= y_test.to_numpy()
_flat = y_test.flatten()
flat = y_pred.flatten()

```



```

data = {'Actual (y_test)': y_test_flat, 'Predicted (y_pred)':
y_pred_flat}
predicition_result_df = pd.DataFrame(data)

predicition_result_df.to_excel('ann_predicition_result_df.xlsx'
, index=False)
predicition_result_df.to_csv('ann_predicition_result_df.csv',
index=False)

# Menghitung RMSE
rmse = np.sqrt(mean_squared_error(y_test, y_pred))
print(f"RMSE: {rmse}")

# Menghitung MAE
mae = mean_absolute_error(y_test, y_pred)
print('MAE:', mae)

# Menghitung accuracy
accuracy = 100 - (mae / np.mean(y_test)) * 100
print('Accuracy:', accuracy)

# Menyimpan model ke file h5
model.save('prepro101010_dwelling_time_ANN_model.h5')

# MODEL TESTING
# Memuat dataset baru
new_data =
pd.read_csv("/content/testing_dataset_prediction_prepro1.csv"
)

# Memuat model
loaded_model =
keras.models.load_model("/content/prepro101010_dwelling_time_
ANN_model.h5")

# Membagi dataset menjadi data prediktor (X) dan data target
(y) pada dataset baru
X_new = new_data.drop('Dwelling Time', axis=1)
y_new = new_data['Dwelling Time']

# Standarisasi dataset
scaler = StandardScaler()

scaled = scaler.fit_transform(X_new)
predictions = loaded_model.predict(X_new_scaled)

# Menghitung RMSE (Root Mean Squared Error) untuk data baru
rmse_new = np.sqrt(mean_squared_error(y_new, predictions))

```



```

# Menghitung MAE (Mean Absolute Error) untuk data baru
mae_new = mean_absolute_error(y_new, predictions)

# Menghitung R-squared (R^2) untuk data baru
r2_new = r2_score(y_new, predictions)

# Menghitung accuracy
accuracy = 100 - (mae_new / np.mean(y_new)) * 100

# Mencetak the evaluation metrics untuk data baru
print(f"Root Mean Squared Error (RMSE): {rmse_new}")
print(f"Mean Absolute Error (MAE): {mae_new}")
print(f"R-squared (R^2): {r2_new}")
print(f"Accuracy: {accuracy}")

# Membandingkan hasil prediksi dengan data aktual
y_new = y_new.to_numpy()
y_new_flat = y_new.flatten()
predictions_flat = predictions.flatten()

data = {'Actual (y_new)': y_new_flat, 'Predicted
(predictions)': predictions_flat}
prediciton_testing_result_df = pd.DataFrame(data)

# Menyimpan hasil prediksi ke dalam file csv dan excel
prediciton_testing_result_df.to_excel('ann_prediciton_testing
_result_df.xlsx', index=False)
prediciton_testing_result_df.to_csv('ann_prediciton_testing_r
esult_df.csv', index=False)

```



**Lampiran 3** Dataset Hasil Preprocessing

<https://docs.google.com/spreadsheets/d/11t-vBcv8k9nzH1I9PYrgvVvRxGcuXve9/edit?usp=sharing&oid=107382073160629724840&rtpof=true&sd=true>



Optimized using  
trial version  
[www.balesio.com](http://www.balesio.com)

#### Lampiran 4 Hasil Prediksi

a. SVR

[https://docs.google.com/spreadsheets/d/1Azt-i\\_T8TyZoLGYX5A\\_FF21WYKozpKPR/edit?usp=sharing&oid=107382073160629724840&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1Azt-i_T8TyZoLGYX5A_FF21WYKozpKPR/edit?usp=sharing&oid=107382073160629724840&rtpof=true&sd=true)

b. ANN

[https://docs.google.com/spreadsheets/d/1ut3DEG4CeXnC\\_G4ruBVRSI48qQwe7Wj8/edit?usp=sharing&oid=107382073160629724840&rtpof=true&sd=true](https://docs.google.com/spreadsheets/d/1ut3DEG4CeXnC_G4ruBVRSI48qQwe7Wj8/edit?usp=sharing&oid=107382073160629724840&rtpof=true&sd=true)

