

# **Perbandingan Algoritma Support Vector Regression dan Artificial Neural Network dalam Prediksi Jumlah Pendaftar Calon Mahasiswa Baru Institut Teknologi Del**

Dokumen Tugas Akhir

Disampaikan Sebagai Bagian dari Persyaratan Kelulusan Sarjana

Program Studi Sarjana Informatika

Oleh :

11S19034      Hotmangasi Manurung

11S19061      Agnes Bertua Nababan



Institut Teknologi Del

2022/2023

## DAFTAR ISI

<b>DAFTAR ISI .....</b>	<b>ii</b>
<b>DAFTAR GAMBAR .....</b>	<b>vi</b>
<b>DAFTAR TABEL .....</b>	<b>vii</b>
<b>ABSTRAK .....</b>	<b>1</b>
<b>BAB 1 PENDAHULUAN .....</b>	<b>3</b>
1.1 Latar Belakang .....	3
1.2 Rumusan Masalah .....	6
1.3 Tujuan Penelitian .....	6
1.4 Ruang Lingkup Penelitiann.....	6
1.5 Hasil yang Diharap kann.....	7
1.6 Tahapan Penelitiann.....	7
1.7 Sistematika Penyajian .....	9
<b>BAB 2 TINJAUAN PUSTAKA .....</b>	<b>11</b>
2.1 Penelitian Terkait .....	11
2.2 Pendaftaran Calon Mahasiswa Baru .....	13
2.3 Machine Learning .....	14
2.4 Data Preprocessing.....	16
2.5 Label Encoding .....	17
2.6 K-Fold Cross Validation .....	17
2.7 <i>Grid Search</i> .....	18
2.8 Support Vector Regression (SVR) .....	19
2.9 Artificial Neural Network .....	23
2.10 Backpropagation .....	24
2.11 Early Stopping.....	30
2.12 Performansi Model.....	31
2.12.1 Mean Square Error.....	31
2.12.2 Root Mean Square Error.....	31
2.13 Flask.....	32
<b>BAB 3 ANALISIS .....</b>	<b>33</b>

3.1 Analisis Domain.....	33
3.2 Analisis Data .....	33
3.2.1 Sumber Data .....	34
3.2.2 Seleksi Data .....	34
3.3 Preprocessing Data.....	35
3.3.1 Data Integration .....	35
3.3.2 Data Cleaning .....	36
3.3.3 Data Encoding .....	39
3.4 Analisis Algoritma SVR .....	39
3.5 Analisis Algoritma ANN Backpropagation .....	45
3.6 Analisis Evaluasi Model .....	49
3.7 Hipotesis Penelitian.....	50
3.7.1 Perumusan H0 dan H1 .....	51
<b>BAB 4 PERANCANGAN.....</b>	<b>52</b>
4.1 Rancangan Desain Penelitian.....	52
4.2 Rancangan Pre-Processing Data .....	54
4.3 Rancangan Implementasi Algoritma.....	54
4.3.1 Rancangan Implementasi Algoritma SVR .....	55
4.3.2 Rancangan Implementasi Algoritma ANN Backpropagation .....	57
4.4 Rancangan Aplikasi Del Predict .....	60
4.4.1 Use Case Diagram .....	61
4.4.2 Use Case Scenario .....	62
4.4.3 Design Aplikasi .....	63
<b>BAB 5 IMPLEMENTASI .....</b>	<b>67</b>
5.1 Lingkungan Implementasi.....	67
5.2 Batasan Implementasi .....	68
5.3 Implementasi Preprocessing .....	68
5.3.1 Pengecekan Outlier.....	68
5.3.2 Encoding.....	69

5.4	Implementasi Algoritma Support Vector Regression.....	70
5.4.1	Pencarian nilai parameter terbaik .....	70
5.4.2	Percobaan variasi nilai k-fold Cross Validation .....	71
5.5	Implementasi Algoritma Artificial Neural Network .....	72
5.5.1	Pencarian Nilai Epoch .....	76
5.5.2	Percobaan Variasi Nilai Epoch .....	77
5.5	Uji Validasi Algoritma .....	77
<b>BAB 6</b>	<b>HASIL DAN PEMBAHASAN .....</b>	<b>78</b>
6.1.	Hasil dan Pembahasan Preprocessing Dataset .....	78
6.2.	Hasil dan Pembahasan Algoritma Support Vector Regression.....	79
6.3.	Hasil dan Pembahasan Algoritma Artificial Neural Network .....	80
6.4.	Pembahasan Evaluasi Model .....	82
6.5.	Hasil dan Pembahasan Uji Validasi .....	82
6.6.	Hasil Implementasi Simulator.....	83
<b>BAB 7</b>	<b>KESIMPULAN DAN SARAN .....</b>	<b>84</b>
7.1.	Kesimpulan .....	84
7.2.	Saran.....	85
<b>LAMPIRAN</b>	<b>.....</b>	<b>86</b>
Lampiran A	Source Code.....	86
Source Code 1.	Encoding Data.....	86
Source Code 2.	<i>Grid Search</i> ANN .....	86
Source Code 3.	<i>Grid Search</i> SVR .....	87
Source Code 4.	Pemodelan Algoritma ANN.....	88
Source Code 5.	Pemodelan Algoritma SVR.....	90
Lampiran B	Hasil Implementasi .....	93
B.1.	Hasil Implementasi Early Stopping .....	93
B.2.	Hasil Implementasi Artificial Neural Network.....	108
B.3.	Hasil Implementasi Support Vector Regression .....	111
Lampiran C	Hasil Implementasi .....	112

<b>DAFTAR REFERENSI.....</b>	<b>113</b>
------------------------------	------------

## DAFTAR GAMBAR

Gambar 2. 1 Jumlah Pendaftar Calon Mahasiswa baru IT Del 2016-2022 .....	13
Gambar 2. 2 Skema 10 Fold Cross Validation .....	18
Gambar 2. 3 Ilustrasi SVR.....	21
Gambar 2. 4 Arsitektur Algoritma Artificial Neural Network .....	23
Gambar 2. 5 Model Overfitting .....	30
Gambar 4. 1 Rancangan desain penelitian yang dilakukan .....	52
Gambar 4. 2 Rancangan pelaksanaan preprocessing data .....	54
Gambar 4. 3 Flow Algoritma Support Vector Regression .....	55
Gambar 4. 4 Flow Algoritma ANN Backpropagation.....	57
Gambar 4. 5 Use Case Diagram Aplikasi.....	61
Gambar 4. 6 Halaman Home .....	64
Gambar 4. 7 Halaman Prediksi SVR.....	64
Gambar 4. 8 Halaman Hasil Prediksi SVR.....	65
Gambar 4. 9 Halaman Prediksi ANN .....	65
Gambar 4. 10 Halaman Hasil Prediksi ANN.....	66

## DAFTAR TABEL

Tabel 3. 1 Informasi yang diperoleh dari BAAK IT Del.....	34
Tabel 3. 2 Atribut pada dataset.....	35
Tabel 3. 3 Dataset hasil integration .....	36
Tabel 3. 4 Dataset sebelum dilakukan missing value.....	37
Tabel 3. 5 Dataset sesudah dilakukan missing value.....	38
Tabel 3. 6 Inisialisasi nilai parameter.....	39
Tabel 3. 7 Data Uji .....	40
Tabel 3. 8 Data Uji .....	40
Tabel 3. 9 Pasangan Data .....	41
Tabel 3. 10 Hasil Perhitungan RBF.....	41
Tabel 3. 11 Hasil Perhitungan Matriks Hessian .....	42
Tabel 3. 12 Hasil Perhitungan Nilai Error .....	42
Tabel 3. 13 Hasil Perhitungan Delta Alpha.....	43
Tabel 3. 14 Hasil Perhitungan Nilai Alpha Baru.....	44
Tabel 3. 15 Hasil Perhitungan Nilai Alpha Star .....	44
Tabel 3. 16 Data Uji yang Ditesting .....	45
Tabel 3. 17 Hasil Alpha Star yang didapat sebelumnya.....	45
Tabel 3. 18 Nilai bias dari input layer ke hidden layer.....	46
Tabel 3. 19 Nilai bobot dari inputan layer ke hidden layer .....	46
Tabel 3. 20 Nilai bias dari hidden layer ke output layer.....	46
Tabel 3. 21 Nilai bobot dari hidden layer ke output layer .....	47
Tabel 4. 1 Inisialisasi Hyperparameter ANN .....	58
Tabel 4. 2 Deskripsi Use Case Diagram.....	61
Tabel 4. 3 Use Case Scenario Prediksi ANN .....	62
Tabel 4. 4 Use Case Scenario Prediksi SVR .....	63
Tabel 5. 1 Spesifikasi Hardware.....	67
Tabel 5. 2 Spesifikasi Software .....	67
Tabel 5. 3 Nilai parameter algoritma support vector regression .....	70
Tabel 6. 1 Dataset yang sudah dilakukan pre-processing .....	78
Tabel 6. 2 Kernel dan Parameter Terbaik.....	79

Tabel 6. 3 Evaluasi RMSE menggunakan nilai k .....	80
Tabel 6. 4 Hasil Uji Validasi dengan model SVR .....	82
Tabel 6. 5 Hasil Uji Validasi dengan Model ANN.....	83



## ABSTRAK

Kesulitan yang dihadapi oleh calon pendaftar atau siswa SMA dalam melanjutkan pendidikannya adalah kesulitan untuk menentukan perguruan tinggi yang akan dipilih. Kesulitan ini sering terjadi dikarenakan kurangnya informasi yang dimiliki untuk mengenal perguruan tinggi yang ada. Pelaksanaan promosi sangat membantu para calon pendaftar atau siswa SMA untuk mengenal perguruan tinggi dan membantu untuk memutuskan perguruan tinggi yang akan dipilih. Bentuk promosi ini dapat dimulai dengan melakukan promosi pada sekolah yang kemungkinan jumlah pendaftar ke kampus Institut Teknologi Del yang masih sedikit. Untuk mengetahui kemungkinan jumlah pendaftar dari suatu sekolah, maka perlu dibuat sebuah sistem yang dapat memprediksi jumlah pendaftar berdasarkan sekolah asal untuk tahun selanjutnya. Pada penelitian ini, penulis mencoba membandingkan dua algoritma yaitu algoritma Support Vector Regression dan Artificial Neural Network untuk menghasilkan model yang terbaik dalam melakukan prediksi jumlah pendaftar calon mahasiswa baru Institut Teknologi Del. Pada proses percobaan, peneliti melakukan pencarian *hyperparameter* untuk digunakan pada model dengan menggunakan *gridsearch*. Dari hasil percobaan yang dilakukan, model terbaik yang dapat memprediksi jumlah pendaftar adalah model algoritma Artificial Neural Network dengan nilai RMSE terendah yang dihasilkan adalah 0.04929599755078. Meskipun model Artificial Neural Network merupakan model terbaik, namun model ini belum dapat melakukan prediksi dengan baik dikarenakan jumlah pola data yang dipelajari masih dalam kategori sedikit, sehingga model belum dapat melakukan prediksi dengan baik.

Kata Kunci : *Artificial Neural Network, Support Vector Regression, Gridsearch, RMSE*, jumlah pendaftar calon mahasiswa.

## ABSTRACT

The difficulty faced by prospective applicants or high school students in continuing their education is the difficulty in determining which college to choose. This difficulty often occurs due to the lack of information owned to recognize existing universities. The implementation of promotion is very helpful for prospective applicants or high school students to get to know the college and help to decide which college to choose. This form of promotion can be started by conducting promotions at schools that may have a small number of applicants to the Del Institute of Technology campus. To find out the possible number of applicants from a school, it is necessary to create a system that can predict the number of applicants based on the school of origin for the following year. In this study, the authors tried to compare two algorithms, namely the Support Vector Regression algorithm and Artificial Neural Network to produce the best model in predicting the number of applicants for new students of Del Institute of Technology. In the experiment process, researchers conducted a hyperparameter search to be used in the model using gridsearch. From the results of the experiments conducted, the best model that can predict the number of applicants is the Artificial Neural Network algorithm model with the lowest RMSE value produced is 0.04929599755078. Although the Artificial Neural Network model is the best model, this model has not been able to predict well because the number of data patterns learned is still in the small category, so the model has not been able to predict well.

**Keywords:** Artificial Neural Network, Support Vector Regression, Gridsearch, RMSE, number of student applicants.

## **BAB 1**

### **PENDAHULUAN**

Pada bab ini dijelaskan mengenai latar belakang topik penelitian, rumusan permasalahan penelitian, tujuan penelitian, ruang lingkup penelitian, metode penelitian dan sistematika penulisan dalam menyusun Tugas Akhir ini.

#### **1.1 Latar Belakang**

Pembukaan pendaftaran mahasiswa baru adalah rutinitas setiap perguruan tinggi. Rutinitas ini juga dilakukan oleh Institut Teknologi Del setiap tahunnya dengan membuka beberapa jalur yaitu PMDK, USM 1, USM 2, USM 3, dan jalur nilai UTBK (ItDel, 2022). Jumlah penerimaan mahasiswa baru mencerminkan bagaimana pandangan dan minat masyarakat pada perguruan tinggi tersebut. Perguruan tinggi swasta saling bersaing untuk meningkatkan jumlah pendaftar mahasiswa baru sehingga jumlah pendaftar mahasiswa baru dapat meningkat maupun menurun setiap tahunnya (Khoirudin et al., 2019). Beberapa faktor yang mempengaruhi daya tarik untuk mendaftar ke suatu perguruan tinggi yaitu besarnya biaya pendidikan, beasiswa yang disediakan, dan bagaimana perguruan tinggi mempromosikan kampusnya (Barus & Simamora, 2020).

Salah satu kesulitan yang dihadapi oleh calon pendaftar atau siswa SMA dalam melanjutkan studinya adalah kesulitan untuk menentukan perguruan tinggi yang akan dipilih. Kesulitan ini sering terjadi dikarenakan kurangnya informasi yang dimiliki untuk mengenal perguruan tinggi yang ada. Pada bentuk kesulitan seperti ini peran dari pihak promosi perguruan tinggi bekerja dalam menarik perhatian pendaftar sangat penting (Barus & Simamora, 2020). Pelaksanaan promosi ini sangat membantu para calon pendaftar atau siswa SMA untuk mengenal perguruan tinggi dan membantu untuk memutuskan perguruan tinggi yang akan dipilih.

Kegiatan promosi juga dilakukan oleh Institut Teknologi Del (IT Del) untuk meningkatkan daya tarik calon pendaftar untuk berkuliah di institusinya. Ada banyak bentuk promosi yang dilakukan pihak IT Del, salah satu bentuk promosi

yang dilakukan adalah pengenalan institut ke sekolah-sekolah. Bentuk promosi ini dapat dimulai dengan melakukan promosi pada sekolah yang kemungkinan jumlah pendaftar ke kampus Institut Teknologi Del yang masih sedikit. Untuk mengetahui kemungkinan jumlah pendaftar dari suatu sekolah, maka perlu dibuat sebuah sistem yang dapat memprediksi jumlah pendaftar berdasarkan sekolah asal untuk tahun selanjutnya.

Prediksi jumlah pendaftar telah dilakukan oleh beberapa peneliti sebelumnya diantaranya, Sistem Prediksi untuk Menentukan Jumlah Pendaftaran Mahasiswa Baru pada Universitas Catur Insan Cendekia Menggunakan Metode *Least Square* (Muhadzdzab et al., 2020) yang membuat sistem prediksi untuk mengetahui jumlah calon mahasiswa baru sebagai upaya untuk mengetahui bagaimana prioritas atau banyak jumlah calon mahasiswa baru yang akan diterima dan sebagai sarana peminat trend prodi terbanyak pada setiap tahunnya. Pada penelitian tersebut, peneliti melakukan prediksi menggunakan algoritma *Least Square*. Berdasarkan hasil prediksi yang dilakukan diperoleh bahwa prediksi jumlah pendaftar pada tahun 2019 adalah 144 orang dengan perolehan MAPE 8,87%.

Penelitian lain yang membahas mengenai prediksi jumlah pendaftar antara lain adalah Implementasi Algoritma Regresi Linear dalam Sistem Prediksi Pendaftar Mahasiswa Baru Sekolah Tinggi Teknologi Indonesia Tanjungpinang (Setyoningrum et al., 2022). Pada penelitian ini peneliti membuat sebuah sistem prediksi jumlah pendaftar dengan menggunakan algoritma regresi linear, dimana dari hasil penelitian ini peneliti berhasil membangun sebuah sistem yang digunakan dalam memprediksi jumlah pendaftar mahasiswa Baru sekolah Tinggi Teknologi Indonesia.

Pada penelitian ini, penulis akan melakukan sistem prediksi jumlah pendaftar calon mahasiswa baru dengan menggunakan hasil perbandingan algoritma Support Vector Regression (SVR) dan Artificial Neural Network (ANN). Sebelum melakukan penelitian, peneliti juga melakukan eksplorasi mengenai algoritma ini dan memperoleh beberapa hasil penelitian dengan kedua algoritma

tersebut. Beberapa penelitian yang membahas mengenai kedua algoritma ini antara lain, Perbandingan Algoritma Backpropagation Neural Network dan Long Short Term Memory dalam Memprediksi Harga Bitcoin (Andreas et al., 2022). Pada penelitian ini, peneliti membandingkan dua algoritma regresi yaitu algoritma Backpropagation dengan algoritma LSTM (*Long Short-Term Memory*) dalam melakukan prediksi harga Bitcoin. Pada penelitian ini algoritma dibandingkan dengan melakukan perhitungan RMSE dari kedua algoritma, kemudian algoritma yang memiliki RMSE terkecil dipilih sebagai algoritma yang lebih baik, dimana dari hasil penelitian ini algoritma dengan RMSE terkecil adalah algoritma Backpropagation.

Penelitian lainnya adalah Perbandingan Regresi Linear, Backpropagation, dan Fuzzy Mamdani dalam Prediksi Harga Emas (Nafi'iyah, 2016). Pada penelitian ini dilakukan pemodelan dengan menggunakan algoritma regresi linear, backpropagation, dan *fuzzy mandani* dalam melakukan prediksi harga emas. Ketiga algoritma ini dibandingkan berdasarkan nilai persentasi akurasi yang didapatkan. Berdasarkan hasil penelitian didapatkan bahwa dari hasil prediksi ketiga algoritma bahwa algoritma *backpropagation* merupakan algoritma yang paling baik dalam memprediksi emas, kemudian dilanjutkan dengan algoritma regresi linier. Sedangkan untuk algoritma *fuzzy mamdani* tidak dapat melakukan prediksi harga emas secara baik, yang dibuktikan dengan nilai akurasi yang tidak sampai 1%.

Penelitian lainnya yang telah dilakukan yaitu Analisis Perbandingan Prediksi Obat dengan Menggunakan Metode ABC Analysis dan SVR pada Aplikasi "MORBIS" (Maryana et al., 2019). Pada penelitian ini didapatkan hasil bahwa optimasi peramalan penjualan obat dengan menggunakan SVR adalah dengan menggunakan proses *preprocessing* data dengan scaling linier mendapatkan hasil yang lebih optimal dengan nilai MAPE 17,51%.

Dari hasil kajian beberapa penelitian di atas diperoleh bahwa algoritma SVR dan ANN merupakan algoritma terbaik untuk melakukan prediksi dibandingkan algoritma lainnya. Oleh karena itu, penulis mencoba untuk menganalisis

algoritma SVR dan ANN lalu menentukan algoritma yang terbaik dari kedua algoritma tersebut. Setelah ditentukan algoritma terbaik, maka penulis akan membuat sebuah aplikasi website untuk memprediksi jumlah pendaftar calon mahasiswa dengan menggunakan algoritma tersebut.

## 1.2 Rumusan Masalah

Rumusan masalah dari penelitian ini antara lain :

1. Bagaimana cara memprediksi jumlah pendaftar calon mahasiswa baru Institut Teknologi Del di tahun selanjutnya dengan menggunakan algoritma *Support Vector Regression* (SVR) dan *Artificial Neural Network* (ANN) ?
2. Bagaimana nilai RMSE yang diperoleh dari hasil pembentukan model algoritma yang telah dibuat?

## 1.3 Tujuan Penelitian

Tujuan dari penelitian ini antara lain :

1. Melakukan prediksi jumlah pendaftar calon mahasiswa baru Institut Teknologi Del dengan menggunakan algoritma *Support Vector Regression* (SVR) dan *Artificial Neural Network* (ANN).
2. Membandingkan nilai performansi model yang dilakukan setelah pembentukan model algoritma *Support Vector Regression* (SVR) dan *Artificial Neural Network* (ANN).

## 1.4 Ruang Lingkup Penelitian

Ruang lingkup penelitian ini adalah sebagai berikut :

1. Penelitian ini berfokus pada penggunaan algoritma *Support Vector Regression* (SVR) dan algoritma *Artificial Neural Network* (ANN) *Backpropagation* dalam prediksi jumlah pendaftar calon mahasiswa baru di Institut Teknologi Del.
2. Data yang digunakan adalah data yang diterima dari BAAK IT Del.
3. Variabel yang digunakan yaitu nama sekolah, jumlah pendaftar tahun 2016, jumlah pendaftar tahun 2017, jumlah pendaftar tahun 2018, jumlah

pendaftar tahun 2019, jumlah pendaftar tahun 2020, jumlah pendaftar tahun 2021, jumlah pendaftar tahun 2022.

4. Sistem prediksi yang dilakukan hanya akan memprediksi jumlah pendaftar untuk 1 tahun selanjutnya.
5. Simulator algoritma akan dibuat dengan membuat aplikasi website.
6. Aplikasi yang dibangun hanya untuk melakukan prediksi jumlah pendaftar calon mahasiswa baru di Institut Teknologi Del dengan menggunakan algoritma yang menghasilkan evaluasi model terbaik.

### **1.5 Hasil yang Diharapkan**

Hasil yang diharapkan dalam pelaksanaan penelitian Tugas Akhir ini adalah mampu melakukan prediksi jumlah pendaftar calon mahasiswa baru di Institut Teknologi Del dan menemukan algoritma terbaik dari antara algoritma *Support Vector Regression* dan *Artificial Neural Network* (ANN) berdasarkan hasil performansi model yang diperoleh.

### **1.6 Tahapan Penelitian**

Tahapan penelitian dalam Tugas Akhir ini dideskripsikan pada pernyataan dibawah ini:

#### **1. Studi Literatur**

Pada tahapan ini dilakukan pengumpulan informasi terkait topik penelitian mengenai algoritma *Support Vector Regression* dan *Artificial Neural Network* dalam melakukan prediksi melalui sumber referensi jurnal, *paper*, buku, situs perguruan tinggi, dan *website* yang terpercaya (seperti *website* milik pemerintah dan situs berita yang bereputasi).

#### **2. Perumusan Masalah**

Pada tahapan ini yang dilaksanakan adalah merumuskan masalah dan merumuskan tujuan penelitian tersebut. Hasil analisis dari sumber referensi-referensi dan data yang sudah disiapkan akan membantu membuat rumusan masalah penelitian dan hasil rumusan masalah tersebut yang nantinya dijadikan tujuan penelitian.

#### **3. Pengumpulan Data**

Pada tahapan ini yang dilaksanakan adalah pengumpulan data-data sebagai bahan pemecahan masalah yang telah dirumuskan sebelumnya pada tahap kedua. Untuk pengumpulan data dilakukan peneliti dengan mengambil data dari kantor BAAK kampus Institut Teknologi Del.

#### 4. *Preprocessing* Data

Tahapan *preprocessing* ini dilakukan untuk menyiapkan data yang dapat diproses oleh algoritma *machine learning*. Memproses data terlebih dahulu penting dilakukan sehingga data yang akan digunakan tidak mengandung kata-kata ataupun karakter yang tidak diperlukan. Sehingga data akan lebih mudah untuk diproses dan meningkatkan akurasi prediksi.

#### 5. Analisis Data dan Algoritma

Pada tahapan ini peneliti melakukan analisis terhadap dataset yang akan digunakan dan akan menganalisis algoritma *Support Vector Regression* dan *Artificial Neural Network* (ANN). Pada analisis algoritma, peneliti memperdalam bagaimana proses kerja dari kedua algoritma. Peneliti juga membuat perhitungan matematis dari kedua algoritma tersebut.

#### 6. Implementasi Algoritma

Pada tahapan ini, peneliti melakukan implementasi kedua algoritma berdasarkan hasil analisis yang telah dilakukan pada tahapan sebelumnya. Implementasi algoritma dilakukan dengan menggunakan bahasa python yang menjadi bahasa pembelajaran *machine learning*.

#### 7. Kesimpulan dan Saran Penelitian

Pada tahapan ini peneliti melakukan pembahasan dan penarikan kesimpulan terhadap proses penelitian yang telah dilaksanakan yaitu dengan melakukan penerapan algoritma *Support Vector Regression* (SVR) dan *Artificial Neural Network* (ANN) dalam melakukan prediksi jumlah pendaftar calon mahasiswa baru Institut Teknologi Del.



## 1.7 Sistematika Penyajian

Penelitian Tugas Akhir ini terdiri dari beberapa bab yang disusun berdasarkan sistematika sebagai berikut :

Bab I Pendahuluan	Pada bab ini dijelaskan mengenai latar belakang topik penelitian, rumusan permasalahan penelitian, tujuan penelitian, ruang lingkup penelitian, metode penelitian dan sistematika penyajian dalam menyusun Tugas Akhir ini.
Bab II Tinjauan Pustaka	Pada bab ini menjelaskan rangkuman informasi dari beberapa dasar teori berupa daftar pustaka yang mendasari dan mendukung penelitian Tugas Akhir ini. Penjelasan rinci yakni penjelasan penelitian sebelumnya yang mirip dengan penelitian Tugas Akhir dilakukan dalam dokumen ini, penjelasan teoritis dari dasar acuan jurnal, artikel, atau buku yang diangkat peneliti dalam menelaah topik dalam Tugas Akhir ini.
Bab III Analisis	Pada bab ini, peneliti memberikan penjelasan mengenai analisis terhadap proses yang telah ditentukan sebelumnya dengan metode yang telah dipilih untuk melakukan proses prediksi jumlah pendaftar calon mahasiswa baru.
Bab IV Perancangan	Pada Bab ini, peneliti memberikan penjelasan mengenai rancangan penelitian yang akan dilakukan sehingga dapat menjelaskan secara rinci mengenai proses penelitian yang akan dilakukan dengan menggunakan <i>flowchart</i> .
Bab V Implementasi	Pada bab ini dijelaskan bagaimana proses mengenai implementasi yang dilakukan berdasarkan analisis bab III dan perancangan yang telah dibahas pada Bab IV.

Bab VI Hasil dan Pembahasan	Pada bab ini akan diuraikan hasil dan pembahasan dari implementasi dan percobaan yang dilakukan untuk mendapatkan hasil yang diharapkan dari penelitian Tugas Akhir ini.
Bab VII Kesimpulan dan Saran	Pada bab ini akan disampaikan kesimpulan dari hasil implementasi dan percobaan dalam penelitian yang dilakukan. Dan juga akan disampaikan beberapa saran yang diperlukan untuk penelitian selanjutnya.

## **BAB 2**

### **TINJAUAN PUSTAKA**

Pada bab 2 Tinjauan Pustaka ini berisi landasan teori yang memuat data dan informasi terkait yang dikumpulkan dari berbagai jurnal dan pustaka yang berhubungan dengan penelitian dalam pengerjaan Tugas Akhir untuk memperluas informasi dalam melakukan kajian.

#### **2.1 Penelitian Terkait**

Sebelum penulis melakukan penelitian Tugas Akhir ini, penulis terlebih dahulu melakukan eksplorasi masalah perbandingan algoritma dalam pembuatan sebuah model untuk mendapatkan hasil yang terbaik terhadap serangkaian kasus. Kemudian berdasarkan permasalahan tersebut, penulis melakukan analisis terhadap beberapa paper/jurnal untuk mendapatkan metode, objek penelitian, hasil penelitian serta relevansi dan juga perbedaan dengan penelitian sebelumnya. Berdasarkan hasil analisis yang didapatkan, penulis melanjutkan penelitian dengan melakukan pengembangan menggunakan metode dan objek penelitian yang berbeda, sehingga dilakukan perbandingan untuk membandingkan satu metode dengan metode lainnya untuk mendapatkan model terbaik.

1. Perbandingan Algoritma Backpropagation Neural Network dan Long Short Term Memory dalam Memprediksi Harga Bitcoin

Pada penelitian ini, peneliti membandingkan dua algoritma regresi yaitu algoritma Backpropagation dengan algoritma LSTM (*Long Short-Term Memory*) dalam melakukan prediksi harga Bitcoin. Pada penelitian ini, algoritma dibandingkan dengan melakukan perhitungan RMSE dari kedua algoritma, kemudian algoritma yang memiliki RMSE terkecil dipilih sebagai algoritma yang lebih baik, dimana dari hasil penelitian ini algoritma dengan RMSE terkecil adalah algoritma Backpropagation (Andreas et al., 2022).

2. Machine Learning menggunakan Perbandingan Dua Algoritma Antara Support Vector Regression (SVR) dan Decision Tree untuk Memprediksi Harga Saham Agro. Penelitian ini dilakukan dengan menggunakan metode Algoritma SVR

dan Decision Tree. Dari hasil penelitian ini didapatkan kesimpulan bahwa tingkat akurasi pada SVR sebesar 96,30% dan akurasi pada decision tree sebesar 90,66% sehingga kedua algoritma ini baik digunakan untuk prediksi saham BRI Agroniaga (AGRO) (Lembong, 2022).

3. Analisis Support Vector Regression (SVR) dengan Kernel Radial Basis Function (RBF) untuk Memprediksi Laju Inflasi di Indonesia.

Penelitian ini dilakukan dengan menggunakan metode algoritma *Support Vector Regression* dengan menggunakan kernel RBF. Dari hasil penelitian ini didapat kesimpulan bahwa Akurasi hasil prediksi yang dilakukan dengan melihat nilai dari Root Mean Square Error. Dengan nilai RMSE yang dihasilkan sebesar 0,0020 yang memiliki pengertian bahwa kemampuan model dapat mengikuti pola data dengan baik (Rais, 2022).

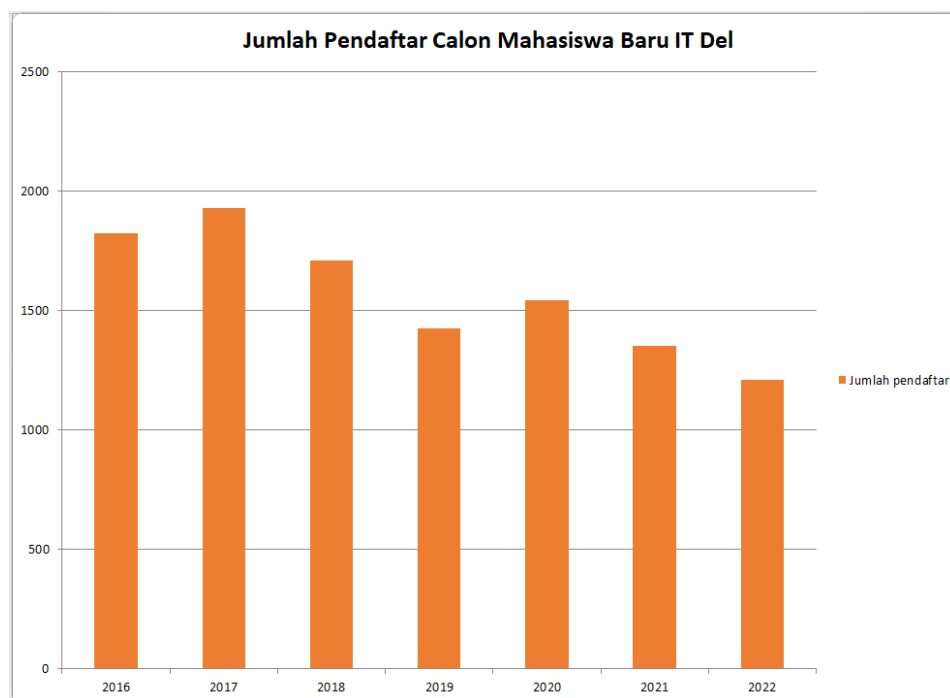
4. Application of artificial neural network and support vector regresion in predicting mass of ber fruits (*Ziziphus mauritiana* Lamk) based on fruit axial dimensions. Penelitian ini dilakukan dengan menggunakan algoritma Artificial Neural Network dan Support Vector Regression. Dari hasil penelitian ini didapatkan bahwa nilai RMSE dari ANN yaitu 1.8479, dengan  $R^2$  yaitu 0.9771 dan untuk algoritma SVR dengan nilai RMSE 1.8814, dengan nilai  $R^2$  yaitu 0.9947. Berdasarkan hasil yang didapatkan kemudian disimpulkan bahwa algoritma ANN mampu meningkatkan kemampuan untuk merancang sistem penilaian kultivar buah secara akurat (Abdel-Sattar et al., 2021)

5. Perbandingan Regresi Linear, Backpropagation, dan Fuzzy Mamdani dalam Prediksi Harga Emas (Nafi'iyah, 2016). Pada penelitian ini dilakukan pemodelan dengan menggunakan algoritma regresi linear, backpropagation, dan fuzzy mandani dalam melakukan prediksi harga emas. Ketiga algoritma ini dibandingkan berdasarkan nilai persentasi akurasi yang didapatkan. Berdasarkan hasil penelitian didapatkan bahwa dari hasil prediksi ketiga algoritma bahwa algoritma *backpropagation* merupakan algoritma yang paling baik dalam memprediksi emas, kemudian dilanjutkan dengan algoritma regresi linier. Sedangkan untuk algoritma *fuzzy mamdani* tidak dapat melakukan

prediksi harga emas secara baik, yang dibuktikan dengan nilai akurasi yang tidak sampai 1%.

## 2.2 Pendaftaran Calon Mahasiswa Baru

Setiap perguruan tinggi melakukan rutinitas pembukaan pendaftaran calon mahasiswa baru di setiap tahunnya. Rutinitas ini juga dilakukan oleh Institut Teknologi Del disetiap tahunnya dengan membuka beberapa jalur yaitu PMDK, USM 1, USM 2, USM 3, dan jalur nilai UTBK (Del, 2022). Jumlah pendaftar calon mahasiswa baru pun setiap tahunnya berubah dan perubahan yang terjadi pun tidak konsisten.



**Gambar 2. 1 Jumlah Pendaftar Calon Mahasiswa baru IT Del 2016-2022**

Sumber : Data BAAK IT Del

Berdasarkan hasil diagram di atas bahwa jumlah pendaftar calon mahasiswa IT Del selalu mengalami perubahan untuk setiap tahunnya. Perubahan jumlah yang selalu naik turun ini juga berlaku ketika kita melakukan *breakdown* data jumlah pendaftar berdasarkan asal sekolah pendaftar.

## 2.3 Machine Learning

*Machine learning* ditemukan pertama kali pada tahun 1959 oleh Arthur Samuel melalui jurnalnya yang berjudul “*Some Studies in Machine Learning Using the Game of Checkers*” (IBM Journal of Research and Development). *Machine learning* adalah salah satu cabang ilmu komputer yang memberikan kemampuan untuk belajar tanpa diprogram secara langsung (eksplisit).

*Machine Learning* membutuhkan model untuk didefinisikan sesuai dengan parameter tertentu. Proses *learning* adalah eksekusi program komputer untuk mengoptimalkan parameter model dengan menggunakan data *training* atau *past experience*. Sederhananya, machine learning merupakan sebuah pemrograman komputer untuk mencapai kriteria atau kinerja tertentu dengan menggunakan sekumpulan data *training* atau pengalaman di masa lalu (*past experience*).

*Machine learning* mempelajari agar komputer mampu “belajar” dari data. *Machine learning* melibatkan berbagai disiplin ilmu seperti statistika, ilmu komputer, matematika, dan bahkan neurologi. *Machine learning* menggunakan teori-teori statistika untuk membentuk model matematika. Model dapat bersifat *predictive* atau *descriptive*, atau bisa juga gabungan dari *predictive* dan *descriptive*. Secara umum, algoritma *machine learning* dikelompokkan menjadi empat kategori, yaitu:

### 1. *Supervised learning*

*Supervised learning* merupakan algoritma *machine learning* yang mengidentifikasi fitur secara eksplisit dan melakukan prediksi atau klasifikasi yang sesuai. *Supervised learning* menggunakan *data training* yang sudah diberi label untuk mempelajari *mapping function*, dari *input* variabel ( $x$ ) ke *output* variabel ( $y$ ). Permasalahan-permasalahan yang terkait dengan *supervised learning* dapat dikategorikan menjadi dua jenis, yaitu *classification* dan *regression*. *Classification* bertujuan untuk memprediksi *outcome* dari *input*, dimana variabel berbentuk kategori-kategori. Contoh : pria/wanita, sakit/sehat, tinggi/rendah, dan sebagainya. Sedangkan *regression* bertujuan untuk memprediksi *outcome* dari *input*, dimana *outcome* variabel

berbentuk nilai aktual (*real values*). Contoh : tinggi badan seseorang, curah hujan, dan sebagainya. Beberapa algoritma yang termasuk *supervised learning* antara lain adalah Decision tree, Naive Bayes Classifier, Artificial Neural Network, Support Vector Machine, Linear Regression, Logistic Regression, CART, KNN (K-Nearest Neighbor), dsb.

## 2. *Unsupervised learning*

*Unsupervised learning* merupakan algoritma *machine learning* yang mengidentifikasi data berdasarkan kepadatan, struktur, segmen serupa, dan fitur serupa lainnya. *Unsupervised learning* menggunakan *unlabeled training dataset* untuk memodelkan struktur dari data, sehingga *unsupervised learning* bersifat lebih subjektif dibandingkan dengan *supervised learning*. Permasalahan seputar *unsupervised learning* dapat dikelompokkan menjadi tiga kategori, yaitu *association*, *clustering*, *dimensionality reduction*. Beberapa algoritma yang termasuk dalam *unsupervised learning* antara lain K-Means, *Hierarchical Clustering*, *Fuzzy C-Means*, *Self-Organizing Map*, dan sebagainya.

## 3. *Reinforcement learning*

*Reinforcement learning* merupakan algoritma yang mengumpulkan informasi atau pengalaman untuk menentukan tindakan yang baik untuk mencapai target atau tujuan. Dengan algoritma ini, agen dapat mengevaluasi perilaku ideal dalam suatu kasus. Berbeda dengan algoritma *supervised learning* dan *unsupervised learning* yang sudah memiliki tujuan (*goal*), algoritma *reinforcement learning* tidak memiliki tujuan yang jelas. Oleh sebab itu, algoritma *reinforcement learning* ini dipaksa untuk belajar menemukan nilai optimal dengan mencari *trial* dan *error*. Beberapa algoritma yang dikelompokkan ke dalam *reinforcement learning* antara lain Algoritma Genetika (*GA*), Pemrograman Dinamis (*DP*), Iterasi Kebijakan Umum (*GPI*), *Monte Carlo*.

#### 4. *Deep Learning*

*Deep learning* merupakan metode learning yang memanfaatkan *artificial neural network* yang berlapis-lapis (*multi layer*). *Artificial neural network* ini dibuat mirip dengan otak manusia, dimana neuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit.

(Primartha, 2018)

### 2.4 Data Preprocessing

Data preprocessing adalah langkah awal data mining yang digunakan untuk mengubah data mentah menjadi format data dan informasi yang lebih efisien dan bermanfaat. Format pada data mentah yang diambil dari berbagai macam sumber seringkali mengalami error, missing value, dan tidak konsisten. Oleh karena itu, perlu dilakukan pengolahan data terlebih dahulu sebelum dilakukan pemodelan agar dapat menghasilkan output yang tepat dan akurat.

Berikut ini adalah tahapan yang dilakukan dalam data preprocessing antara lain.

#### 1. Data Cleaning

Data yang dikumpulkan kemungkinan tidak relevan, terdapat kesalahan, dan duplikasi, sehingga dibutuhkan proses pembersihan data. Dalam tahapan ini, data dibersihkan dengan beberapa proses seperti missing value dan noise.

#### 2. Data Integration

Tahapan ini merupakan lanjutan dari tahap data cleaning yang bertujuan untuk mengintegrasikan data dari sumber yang berbeda. Pada tahap data integration ini, perlu dilihat kembali sumber data yang diperoleh. Hal ini penting dilakukan agar data yang digunakan memiliki format yang sama.

#### 3. Data Reduction

Pada tahapan ini dilakukan pemilihan data dengan volume besar yang akan memakan waktu yang cukup lama. Maka dari itu, perlu dilakukan proses data reduction untuk membatasi kumpulan data, dengan tujuan untuk meningkatkan efisiensi penyimpanan, sekaligus mengurangi biaya dan menghemat waktu. Proses pengurangan atau data reduction dilakukan tanpa mengubah hasil dari analisis data. Pengurangan data ini dapat dilakukan



dengan tiga cara, yaitu pengurangan dimensi, pengurangan jumlah, dan kompresi data.

#### 4. Data Transformation

Pada tahapan ini data akan diubah menjadi bentuk yang sesuai dalam proses pemodelan data.

Adapun manfaat yang diperoleh dengan melakukan data preprocessing adalah :

1. Mempercepat proses data mining
2. Data yang dikumpulkan menjadi lebih mudah dimengerti
3. Mengurangi representasi dalam data
4. Mempermudah proses machine learning dan analisis data

### 2.5 Label Encoding

Dalam pemrosesan data sendiri, sebuah dataset seringkali memiliki data dengan nilai-nilai kategori pada sebuah kolom atau atribut. Dalam melakukan pemrosesan data dengan menggunakan algoritma pada pembelajaran mesin, perlu dilakukan pengubahan nilai-nilai kategori tadi menjadi nilai numerik. Salah satu teknik yang digunakan yaitu Label Encoder. Label Encoder merupakan teknik pada pembelajaran mesin yang digunakan untuk mengubah data kategori menjadi data numerik.

Pada label encoding, nilai-nilai kategori akan diubah menjadi bilangan bulat dengan menggunakan suatu aturan tertentu. Misalnya nama sekolah tertentu akan diubah menjadi sebuah bilangan tertentu. Dengan demikian seluruh kolom Nama Sekolah pada dataset akan dirubah menjadi kolom bertipe integer. Dan setelah itu dapat digunakan untuk melakukan pemodelan pada pembelajaran mesin (Santoso & Putri, 2023)

### 2.6 K-Fold Cross Validation

*K-Fold cross validation* adalah salah satu metode yang digunakan untuk mengevaluasi algoritma dengan membagi jumlah sebanyak  $k$  ke dalam data train dan data test. Cara kerja dari *k-fold cross validation* adalah pertama-tama data dipartisi menjadi  $k$  bagian yang berukuran sama (atau hampir sama). Kemudian setiap data ke- $k$  setiap bagian dijadikan sebagai data testing.

[illegible]

### Gambar 2. 2 Skema 10 Fold Cross Validation

Dari **Gambar 2.2** di atas kita dapat melihat langkah-langkah proses pembagian data menjadi 10 bagian. Pada bagian pertama atau *fold-1* data ke-1 dijadikan sebagai data uji dan sisanya menjadi data latih. Kemudian pada *fold-2* data ke-2 dijadikan sebagai data uji dan sisanya menjadi data latih, hal ini terjadi sampai mencapai nilai *fold-10* .

## 2.7 Grid Search

Metode *Grid Search* merupakan alternatif untuk menemukan parameter terbaik dari sebuah model. Metode ini dikategorikan sebagai metode lengkap untuk nilai parameter terbaik yang harus dieksplorasi masing-masing dengan menetapkan jenis prediksi terlebih dahulu. Kemudian metode akan menunjukkan skor untuk setiap nilai parameter untuk mempertimbangkan mana yang akan dipilih (Gunawan et al., 2020).

Metode *Grid Search* memiliki cara kerja kombinasi parameter satu persatu dan membandingkan nilai galat terkecil pada parameter tersebut. Pasangan kombinasi dari parameter terlebih dahulu disimpan dalam grid-grid. Baris ke-i dan kolom ke-j yang nilai galatnya terkecil merupakan kombinasi parameter ke-i dan parameter ke-j terpilih (Saputra et al., 2019).

## 2.8 Support Vector Regression (SVR)

Support Vector Regression (SVR) merupakan penerapan dari algoritma Support Vector Machine (SVM) untuk kasus regresi. Algoritma yang diperkenalkan oleh Vapnik pada tahun 1995 ini didasarkan pada *risk minimization*, yang dipergunakan untuk mengestimasi sebuah fungsi dengan cara meminimalkan batas atas generalization error.

Adapun keunggulan dari algoritma SVR yaitu:

1. SVR mampu memberikan model pelatihan lebih cepat. Dengan karakteristiknya untuk memecahkan permasalahan linear, maka algoritma ini lebih cocok untuk data set dengan dimensi lebih tinggi
2. SVR mampu menyelesaikan permasalahan minimal lokal dari optimasi nonlinier
3. SVR mampu melakukan penyelesaian norm error pada saat penalti outlier selama fase pelatihan. Hal ini yang diketahui dengan menggunakan trik kernel
4. SVR efektif untuk melakukan generalisasi sampel data yang sedikit
5. SVR mampu menghindari overfitting

(Saadah et al., 2021)

Tujuan dari Algoritma SVR adalah membuat data lebih acak agar dapat menerima regresi yaitu dengan memetakan pada dimensi yang lebih tinggi (Ridwan et al., 2021). Pada regresi linear persamaan yang digunakan dijabarkan sebagai berikut:

$$f(x) = w \cdot x + b \quad (\text{Pers. 2.9.1})$$

dengan :

w = vektor pembobot

x = vektor input berupa data

b = nilai bias

Pada SVR nonlinear pertama-tama data input  $x$  dipetakan oleh fungsi  $\varphi$  ke ruang feature. Dengan demikian fungsi regresi memiliki bentuk umum sebagai berikut:

$$f(x) = w\varphi(x) + b \quad (\text{Pers 2.9.2})$$

dengan :

$\varphi(x)$  = fungsi yang memetakan  $x$  pada sebuah dimensi

$w$  = vektor bobot

$b$  = bias

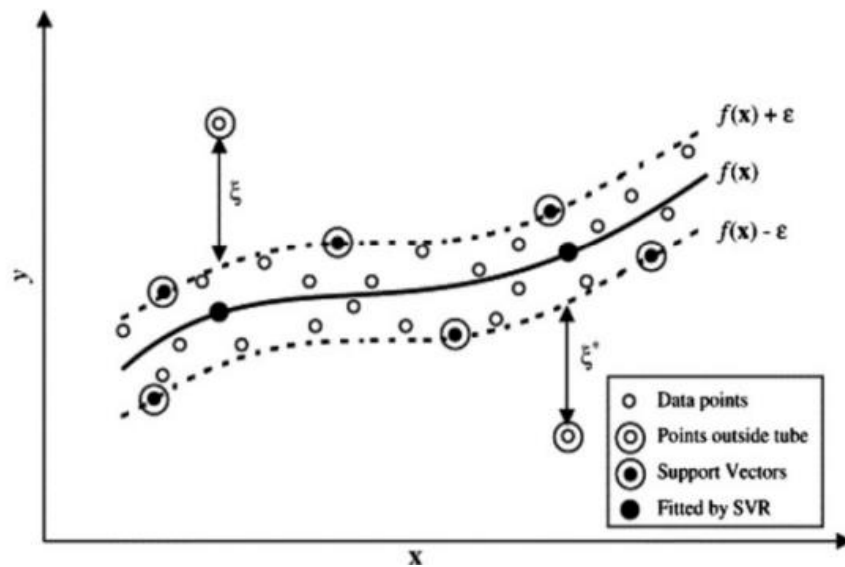
$$\min = \frac{1}{2} ||w||^2 \quad (\text{Pers. 2.9.3})$$

Adapun tujuan untuk meminimalkan fungsi tersebut adalah untuk membuat fungsi menjadi setipis mungkin dan mengurangi kompleksitas perhitungan waktu sehingga waktu yang digunakan menjadi lebih cepat. Berikut merupakan fungsi  $\varepsilon - insensitive loss$  agar dapat melakukan generalisasi dengan baik dengan batasan yang harus dipenuhi adalah titik harus berada pada rentang  $\varepsilon$  seperti pada persamaan berikut:

$$y_i - wx_i - b \leq \varepsilon \quad (\text{Pers. 2.9.4})$$

$$wx_i + b - y_i \leq \varepsilon, \quad \text{dimana } i=1,2,\dots,\lambda \quad (\text{Pers. 2.9.5})$$

Pada persamaan yang digunakan untuk meminimalkan fungsi regresi diasumsikan bahwa semua titik yang berada pada rentang  $f \pm \varepsilon$  disebut dengan istilah *feasible*, sedangkan apabila terdapat titik yang berada diluar rentang  $f \pm \varepsilon$  maka disebut dengan istilah *infeasible*, dimana dapat ditambahkan variabel slack  $\xi_i, \xi_i^*$  untuk mengatasi masalah titik yang berada di rentang  $f(x)$ .



**Gambar 2. 3 Ilustrasi SVR**

Semua titik yang berada di luar margin akan dikenai penalti sebesar  $C$  dengan syarat  $C > 0$ . Masalah optimasi untuk meminimalkan fungsi regresi dapat dijabarkan sebagai berikut:

$$\min \frac{1}{2} ||w||^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*), C > 0 \quad (\text{Pers. 2.9.6})$$

dengan :

$C$  = nilai penalti

$N$  = jumlah seluruh data

$\xi_i$  = titik di luar margin yang berada di atas hyperplane

$\xi_i^*$  = titik di luar margin yang berada berikut hyperplane

dengan ketentuan:

$$y_i - wx_i - b \leq \varepsilon + \xi_i \quad (\text{Pers. 2.9.7})$$

$$wx_i + b - y_i \leq \varepsilon + \xi_i^* \quad (\text{Pers. 2.9.8})$$

$$\xi_i, \xi_i^* \geq 0, \text{ dimana } i=1,2,\dots,\lambda \quad (\text{Pers. 2.9.9})$$

Sehingga diperoleh fungsi linear SVR seperti berikut:

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot x_i + b \quad (\text{Pers. 2.9.10})$$

Untuk kasus yang non-linear SVR digunakan persamaan seperti berikut:

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot k(x_i, x) + b \quad (\text{Pers. 2.9.11})$$

dengan :

$k(x_i, x)$  = fungsi kernel yang digunakan

$\alpha_i \alpha_i^*$  = dual variabel lagrange multiplier

b = bias

Ada tiga bentuk fungsi kernel pada model SVR, antara lain :

1. Kernel Linear

Kernel ini berfungsi untuk pemetaan ke ruang dimensi yang lebih tinggi dan yang sudah dianalisis sudah terpisah secara linier.

$$k(x, x') = x^T y + c \quad (\text{Pers. 2.9.12})$$

2. Kernel Polynomial

Kernel ini berfungsi untuk melakukan pemetaan ketika data tidak terpisah secara linier.

$$k(x, x') = (\alpha x^T y + c)^d \quad (\text{Pers. 2.9.13})$$

3. Kernel Sigmoid

Kernel ini berfungsi untuk melakukan pemetaan data ke dalam ruang fitur yang lebih tinggi.

$$k(x, y) = \tanh(\alpha x \cdot y + c) \quad (\text{Pers. 2.9.14})$$

4. Kernel Radial Basis Function (RBF)

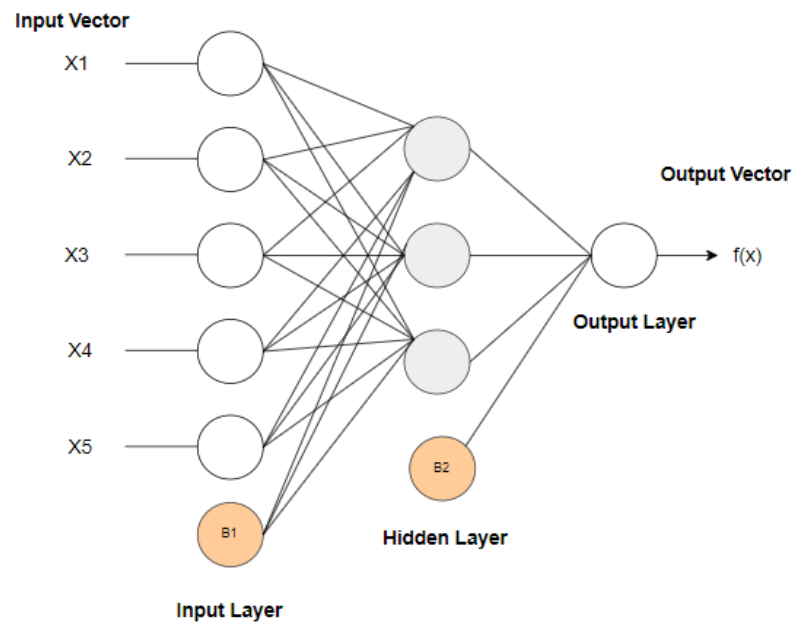
Kernel ini dapat mengatasi permasalahan yang bersifat non-linear.

$$k(x, x') = \exp(-\gamma \|x - y\|^2) \quad (\text{Pers. 2.9.15})$$

Penelitian ini akan dilakukan pencarian kernel terbaik untuk melakukan pemodelan. Algoritma SVR sendiri juga memiliki beberapa parameter yang dapat disesuaikan untuk mendapatkan model terbaik dengan nilai error yang rendah. Pada penelitian ini, beberapa parameter yang digunakan yaitu parameter cost, gamma, dan epsilon. Parameter cost merupakan parameter yang mengendalikan seberapa keras SVR untuk melakukan prediksi dengan benar setiap titik data pelatihan, parameter gamma berfungsi untuk mempengaruhi fleksibilitas SVR dalam menentukan batas pemisah antar kelas yang berbeda. Dan untuk parameter epsilon berfungsi untuk memutuskan seberapa sering algoritma harus mencoba tindakan acak dan menjelajahi lingkungan (Handayani et al., 2017).

## 2.9 Artificial Neural Network

*Artificial Neural Network* (ANN) atau jaringan saraf tiruan adalah pemodelan kompleks yang dapat memprediksi bagaimana ekosistem akan merespon perubahan variabel lingkungan, dan berguna dalam memproses informasi, terutama dalam sistem saraf biologis dalam sel otak manusia. Arsitektur jaringan syaraf tiruan memiliki beberapa lapisan, yang mana setiap lapisan memiliki jumlah node dan neuron yang berbeda - beda.



**Gambar 2. 4** Arsitektur Algoritma Artificial Neural Network

Berdasarkan gambar di atas terlihat bahwa jaringan syaraf tiruan terdiri dari tiga lapisan layer, diantaranya adalah :

### 1. Input Layer

Input layer merupakan lapisan yang terdiri dari neuron - neuron yang berfungsi menerima sinyal dari luar dan kemudian meneruskan ke neuron lain dalam jaringan (Yuniarti & Gunawan, 2021).

### 2. Hidden layer

*Hidden layer* merupakan tiruan dari sel - sel syaraf konektor pada jaringan syaraf biologis. yang mana lapisan ini berfungsi untuk meningkatkan

kemampuan jaringan dalam memecahkan permasalahan (Yuniarti & Gunawan, 2021).

### 3. Output Layer

Output layer merupakan tiruan sel-sel syaraf motor pada jaringan syaraf biologis. lapisan ini berfungsi untuk menyalurkan sinyal-sinyal keluaran hasil proses jaringan (Yuniarti & Gunawan, 2021).

## 2.10 Backpropagation

*Backpropagation* adalah salah satu bentuk metode *artificial neural network*. Metode ini menggunakan arsitektur *multilayer* dengan menggunakan metode pelatihan *supervised training* (Situmorang, 2022). Istilah *Backpropagation* mengacu pada bagaimana sebuah *error* dihitung mulai dari *output layer*, kemudian disebarkan ke *hidden layer* dan berakhir pada *input layer*. Pada input layer tidak terjadi proses komputasi, namun terjadi pengiriman sinyal *input X* ke *hidden layer*. Pada *hidden layer* dan *output layer* terjadi proses komputasi terhadap bobot dan bias, selain itu dilakukan perhitungan terhadap hasil dari *hidden layer* ke *output layer* tersebut dengan algoritma *Backpropagation*.

Langkah-langkah yang perlu dilakukan untuk pelatihan Backpropagation Neural Network adalah sebagai berikut :

### Fase 0 :

Langkah 1 : Melakukan inisialisasi bobot, bias, *learning rate*, dan *epoch*.

Langkah 2 : Jika kondisi berhenti tidak dipenuhi, maka kerjakan langkah 3 dan seterusnya, yang mana dikatakan berhenti jika nilai *error* “<=” dari nilai target *error (threshold)* atau jika jumlah *epoch* “>=” maksimum *epoch* (iterasi).

### Fase 1 : Feed Forward Propagation

Langkah 3 : Setiap unit pada *input layer* ( $p = 1, 2, 3, \dots, n$ ) akan menerima sinyal



*inputan* ( $i_p$ ) untuk diteruskan ke masing-masing *unit* yang berada di *hidden layer*.

Langkah 4 : Menghitung keluaran dari setiap *unit* tersembunyi ( $q = 1, 2, 3, \dots, n$ )

dengan menggunakan *dot product*, penambahan matrix, dan fungsi aktivasi.

Rumus perhitungan menggunakan *dot product* dan penambahan matrix :

$$[Z_{in1} \dots Z_{inp}] = [X_i \dots X_p] \cdot \begin{bmatrix} W_{11} & \dots & W_{1q} \\ \dots & \dots & \dots \\ W_{p1} & \dots & W_{pq} \end{bmatrix} + [b_{xz_1} \dots b_{xz_q}]$$

(Pers. 2.11.1)

dengan :

$Z_{inq}$  = keluaran untuk unit  $Z_q$

$Z_q$  = unit ke -q pada *hidden layer*

$X_p$  = neuron ke-p pada *input layer*

$W_{pq}$  = bobot *neuron input* ke *neuron hidden*

$b_{xz_q}$  = bias *neuron hidden* ke-q

Rumus perhitungan untuk mengaktifkan fungsi aktivasi dari nilai keluaran di masing-masing *neuron hidden* dengan menggunakan persamaan sigmoid :

$$\partial[Z_{out1} \dots Z_{outq}] = f([Z_{in1} \dots Z_{inq}]) = \left[ \frac{1}{1 + e^{-Z_{in1}}} \dots \frac{1}{1 + e^{-Z_{inq}}} \right]$$

(Pers. 2.11.2)

dengan :

$Z_{outq}$  = sinyal yang dikirim untuk *layer* berikutnya

e = nilai eksponensial dengan 2,718281....

Langkah 5 : Lakukan hal yang sama untuk menghitung keluaran setiap *unit* pada layer berikutnya hingga mencapai keluaran pada *output layer* dengan menggunakan rumus berikut :

$$[Y_{in1} \dots Y_{inr}] = [Z_{out1} \dots Z_{outq}] \cdot \begin{bmatrix} W_{11} & \dots & W_{1r} \\ \dots & \dots & \dots \\ W_{q1} & \dots & W_{qr} \end{bmatrix} + [b_{zy_1} \dots b_{zy_r}]$$

(Pers. 2.11.3)

Gunakan fungsi aktivasi untuk menghitung sinyal output, seperti pada persamaan :

$$\partial[Y_{out1} \dots Y_{outr}] = f([Y_{in1} \dots Y_{inr}]) = \left[ \frac{1}{1 + e^{-Y_{in1}}} \dots \frac{1}{1 + e^{-Y_{inr}}} \right]$$

(Pers. 2.11.4)

dengan :

$Y_r$  = unit ke-r pada *output layer*

$Y_{inr}$  = keluaran untuk unit  $Y_r$

$Z_{outq}$  = sinyal yang dikirim dari *hidden layer* ke *output layer*

$W_{qr}$  = bobot *neuron hidden* ke *neuron output*

$b_{ZY_r}$  = bias *neuron output* ke-r

$Y_{outr}$  = keluaran untuk *output layer* yang akan menjadi nilai prediksi

e = nilai eksponensial dengan 2,718281....

Langkah 6 : Menghitung nilai *error/loss* dengan menggunakan persamaan *Mean Square Error* (MSE). Bentuk persamaan dari MSE adalah sebagai berikut.

$$loss = \frac{1}{2} (aktual - Y_{outr})^2$$

(Pers. 2.11.5)

dengan :

aktual = nilai aktual

$Y_{outr}$  = nilai prediksi

Bila nilai *error* yang dihasilkan > target *error (threshold)*, lakukan *backpropagation* untuk memperbaharui parameter. Sebaliknya jika nilai *error* yang dihasilkan < target *error* maka nilai bobot dan bias disimpan untuk digunakan dalam pemodelan.

(Situmorang, 2022)

## **Fase 2 : Backporpagation**

Langkah 7: Pencarian nilai dari *local gradient* dari setiap unit-unit di *output layer* dan *hidden layer*

Pada tahap *backpropagation* ini hal yang paling utama adalah menyesuaikan bobot yang akan menghasilkan sebuah output atau keluaran dengann *error* yang

paling minimal. Dalam menyesuaikan bobot perlu dilakukan pencarian nilai dari *local gradient* dari setiap unit-unit di *output layer* dan *hidden layer*.

Pada pencarian *local gradient* di *hidden layer* dengan *output layer* berbeda, dilakukan pencarian dengan persamaan berikut :

$$\delta_r = (\text{aktual} - Y_{outr}).f'([Y_{in1} \dots Y_{inr}]) \quad (\text{Pers. 2.11.6})$$

dengan :

aktual = nilai aktual

$Y_{outr}$  = nilai prediksi

$f'([Y_{in1} \dots Y_{inr}])$  = turunan aktivasi keluaran untuk unit  $Y_r$

$\text{aktual} - Y_{outr}$  = selisih nilai aktual dan nilai prediksi

Langkah 8 : Menghitung delta bobot dan delta bias pada *output layer*

Selanjutnya dilakukan perhitungan delta bobot atau delta *weight* dan delta bias pada *output layer*. Perhitungan delta bobot ditunjukkan dengan persamaan:

$$\Delta W_{qr} = \eta \delta_r \partial [Z_{out1} \dots Z_{outq}] \quad (\text{Pers. 2.11.7})$$

dengan :

$\Delta W_{qr}$  = menghitung korelasi bobot

$\eta$  = *learning rate*

$\delta_r \partial [Z_{out1} \dots Z_{outq}]$  = *local gradient* dikalikan dengan nilai dari unit pada *hidden layer*.

Untuk menghitung nilai dari delta bias ditunjukkan dengan persamaan :

$$\Delta b_r = \eta \delta_r \quad (\text{Pers. 2.11.8})$$

dengan :

$\Delta b_r$  = menghitung delta bias

$\eta \delta_r$  = *learning rate* dengan *local gradient* pada *output*

Langkah 9 : Menghitung informasi *local gradient* di *hidden layer*

Pada *hidden layer* juga dilakukan perhitungan *local gradient* dan delta bobot, namun pada *hidden layer* terdapat perbedaan dalam perhitungan keduanya. Pada unit-unit *hidden layer* melakukan penjumlahan dari perkalian antara *local*

*gradient output layer* dengan *weight*. Persamaan untuk menghitung informasi *local gradient* di *hidden layer* ditunjukkan dengan persamaan berikut ini :

$$\delta_q = f'([Z_{in1} \dots Z_{inq}]) \cdot \sum_{r=1}^r w_{qr} \cdot \delta_r \quad (\text{Pers. 2.11.9})$$

dengan :

$W_{qr}$  = bobot *neuron hidden* ke *neuron output*  
 $\delta_r$  = menghitung *local gradient* pada *output layer*  
 $f'([Z_{in1} \dots Z_{inq}])$  = turunan fungsi aktivasi pada *hidden layer*

Langkah 10 : Selanjutnya dilakukan perhitungan delta bobot dan delta bias pada *hidden layer*. Perhitungan delta bobot dihitung dengan persamaan :

$$\Delta W_{pq} = \eta \delta_q [X_i \dots X_p] \quad (\text{Pers. 2.11.10})$$

dengan :

$\Delta W_{pq}$  = delta bobot pada *hidden layer*  
 $\eta$  = *learning rate*  
 $\delta_q [X_i \dots X_p]$  = *local gradient* dikalikan dengan *input data*

Untuk menghitung nilai dari delta bias ditunjukkan dengan persamaan :

$$\Delta b_p = \eta \delta_q \quad (\text{Pers. 2.11.11})$$

dengan :

$\Delta b_p$  = menghitung delta bias  
 $\eta$  = *learning rate*  
 $\delta_q$  = *local gradient* pada *hidden layer*

### **Fase 3: Update Bobot dan Bias**

Langkah 11 : Perubahan nilai bobot dan nilai bias

Pada perubahan bobot di *output layer* dan *hidden layer* dilakukan penjumlahan nilai dari bias lama dijumlahkan dengan nilai delta bobot. Pada setiap unit-unit *output layer* dilakukan perubahan bobot yang ditunjukkan dengan persamaan:

$$\Delta W_{qr \text{ baru}} = W_{qr \text{ lama}} + \Delta W_{qr} \quad (\text{Pers. 2.11.12})$$

dengan :

$\Delta W_{qr \text{ baru}}$  = delta bobot baru pada *output layer*  
 $W_{qr \text{ lama}}$  = nilai bobot lama pada *output layer*

$\Delta W_{qr}$  = delta bobot pada *output layer*

Untuk menghitung perubahan nilai bias ditunjukkan dengan persamaan :

$$\Delta b_{r \text{ baru}} = b_{r \text{ lama}} + \Delta b_r \quad (\text{Pers. 2.11.13})$$

dengan :

$\Delta b_{r \text{ baru}}$  = delta bias baru pada *output layer*

$b_{r \text{ lama}}$  = nilai bias lama pada *output layer*

$\Delta b_r$  = delta bias pada *output layer*

Pada *hidden layer* juga dilakukan perubahan nilai bobot dan perubahan nilai bias.

Pada setiap unit-unit *hidden layer* dilakukan perubahan nilai bobot yang ditunjukkan dengan persamaan:

$$\Delta W_{pq \text{ baru}} = W_{pq \text{ lama}} + \Delta W_{pq} \quad (\text{Pers. 2.11.14})$$

dengan :

$\Delta W_{pq \text{ baru}}$  = delta bobot baru pada *hidden layer*

$W_{pq \text{ lama}}$  = nilai bobot lama pada *hidden layer*

$\Delta W_{pq}$  = delta bobot pada *hidden layer*

Untuk menghitung perubahan nilai bias ditunjukkan dengan persamaan :

$$\Delta b_{p \text{ baru}} = b_{p \text{ lama}} + \Delta b_p \quad (\text{Pers. 2.11.15})$$

dengan :

$\Delta b_{p \text{ baru}}$  = delta bias baru pada *hidden layer*

$b_{p \text{ lama}}$  = nilai bias lama pada *hidden layer*

$\Delta b_p$  = delta bias pada *hidden layer*

#### Langkah 12 : Test kondisi berhenti

Setelah dilakukan perbaikan bobot dan bias dilakukan test kondisi berhenti. Selain berdasarkan nilai error seperti pada langkah 6, perhitungan akan berhenti ketika nilai *epoch* (iterasi) yang dilakukan sudah mencapai nilai maksimum *epoch* yang sudah ditentukan.

Kelebihan dari algoritma *Backpropagation* ini antara lain adalah :

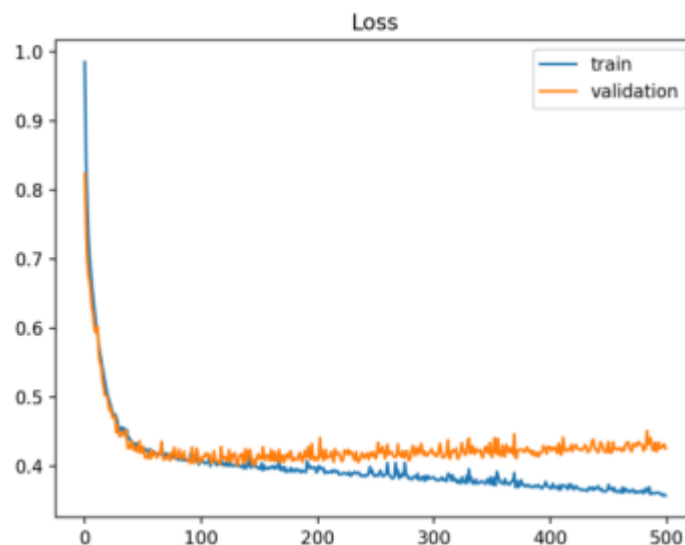
1. Algoritma *Backpropagation* mudah, cepat, dan simple untuk diprogram
2. Hanya perlu mengatur nilai inputan, tanpa mengatur parameter lain

3. Algoritma fleksibel dan efisien dalam penggunaannya.
4. Pengguna algoritma tidak perlu mempelajari fungsi spesial.

(Ishukatiyar, 2023)

### 2.11 Early Stopping

Pada saat melakukan pelatihan neural network, sangat penting dilakukan pencarian cara untuk menghindari *overfitting* pada data pelatihan. *Overfitting* mengacu pada model yang mempelajari data latih data latih dengan sangat baik (Mahyunis, 2022) . Keadaan *overfitting* dapat dilihat pada grafik nilai loss di bawah ini.



**Gambar 2. 5 Model Overfitting**

Berdasarkan **Gambar 2.5** di atas, *overfitting* terjadi ketika nilai *loss* pada data latih sudah membaik sedangkan nilai *loss* pada data uji mengalami kenaikan dibandingkan nilai *loss* data latih. Salah satu *hyperparameter* yang mempengaruhi keadaan model yang *overfitting* atau tidak adalah jumlah *epoch*. Jika jumlah *epoch* yang digunakan terlalu banyak, maka kemungkinan neural network akan *overfitting*. Di sisi lain, jika *epoch* yang digunakan terlalu sedikit, neural network memiliki kemungkinan untuk tidak memiliki kesempatan untuk mempelajari data latih sepenuhnya (Early Stopping, 2020). *Early stopping* adalah teknik optimasi yang digunakan untuk mengatasi *overfitting* tanpa mengorbankan akurasi model.

*Early stopping* akan menghentikan pelatihan ketika performansi model mulai memburuk yang dilihat berdasarkan nilai loss dan mulai terjadi *overfitting* (Mustafeez, 2023).

## 2.12 Performansi Model

Evaluasi hasil pada sebuah sistem adalah sebuah hal yang sangat penting untuk mengetahui tingkat keberhasilan dari suatu sistem. Pengukuran evaluasi hasil ini dapat dilakukan untuk menentukan model terbaik yang dapat digunakan. Untuk mengukur performa dari model prediksi yang digunakan adalah dengan membandingkan nilai aktual dan nilai prediksi.

### 2.12.1 Mean Square Error

*Mean Square Error* (MSE) merupakan model evaluasi yang digunakan untuk melakukan evaluasi terkait hasil yang diberikan oleh model dalam meramalkan nilai atribut dependen yang merupakan data numeric berdasarkan nilai atribut independen. Untuk melakukan perhitungan *Mean Square Error* dapat dilakukan dengan menggunakan persamaan berikut.

$$MSE = \frac{1}{n} \sum_{t=1}^n (\text{aktual} - \text{peramalan})^2 \quad (\text{Pers. 2.13.1})$$

*Mean Square Error* (MSE) digunakan untuk memeriksa kedekatan nilai prediksi atau *forecast* dengan nilai sebenarnya. Semakin rendah nilai MSE maka akan semakin menunjukkan kecocokan yang lebih baik antara nilai ramalan dengan nilai aktual.

### 2.12.2 Root Mean Square Error

*Root Mean Square Error* (RMSE) merupakan model evaluasi hasil yang menghitung tingkat kesalahan hasil prediksi dengan mencari nilai akar MSE antara nilai prediksi dengan nilai aktual yang diinterpretasikan dengan persamaan berikut ini.

$$RMSE = \sqrt{MSE} \quad (\text{Pers. 2.13.2})$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (aktual - peramalan)^2} \quad (\text{Pers. 2.13.3})$$

Semakin kecil nilai RMSE yang dihasilkan model, maka nilai prediksi semakin akurat (Andriyani, 2021). RMSE yang lebih tinggi menunjukkan bahwa ada penyimpangan yang besar dari residual ke kebenaran dasar.

### 2.13 Flask

Flask merupakan sebuah *web framework* yang diimplementasikan menggunakan bahasa pemrograman Python dan tergolong sebagai salah satu jenis *microframework* dikarenakan tidak memerlukan suatu alat atau pustaka tertentu dalam penggunaannya. Framework yang dipelopori oleh Armin Ronacher ini jauh lebih ringan dan cepat karena dibuat dengan ide menyederhanakan inti framework-nya seminimal mungkin. Flask berfungsi sebagai kerangka kerja aplikasi dan tampilan dari suatu web. Dengan menggunakan Flask dan bahasa Python, pengembang dapat membuat sebuah web terstruktur dan dapat mengatur *behavior* suatu *web* dengan lebih mudah (Somya, 2018).

Sebagian besar fungsi dan komponen umum seperti validasi form, database, dan sebagainya tidak terpasang secara default pada Flask dikarenakan fungsi dan komponen-komponennya sudah dapat menggunakan ekstensi yang membuat fitur dan komponen-komponen tersebut seakan diimplementasikan oleh flask sendiri.

Tahapan singkat mengenai pembuatan *website* sederhana menggunakan *flask* :

1. Melakukan instalasi *python*.
2. Melakukan *setting Environment* atau yang sering disebut dengan mengatur *virtual environment* secara sederhana.
3. Melakukan *instalasi flask* melalui pip pada *command prompt*.
4. Membuat *code server*
5. Membuka URL hasil *coding*.



## **BAB 3**

### **ANALISIS**

Pada bab ini akan dijelaskan mengenai analisis dan perancangan yang akan dilaksanakan dalam pengerjaan penelitian Tugas Akhir ini.

#### **3.1 Analisis Domain**

Penelitian ini bertujuan untuk menghasilkan model untuk melakukan prediksi jumlah pendaftar calon mahasiswa baru di Institut Teknologi Del. Hasil penelitian ini diharapkan dapat membantu pihak institusi untuk melakukan promosi pada sekolah-sekolah yang nilai pendaftar dari asal sekolah tersebut masih sedikit. Dengan menggunakan data pendaftar pada masa PMB IT Del tahun 2016 sampai 2022, maka akan dilakukan penggunaan *Machine Learning* untuk mempelajari data yang ada.

Algoritma *Machine Learning* yang digunakan dalam penelitian ini ada dua yaitu, algoritma *Support Vector Regression* (SVR) dan algoritma *Artificial Neural Network* (ANN). Kedua algoritma ini akan digunakan untuk melakukan prediksi dengan tujuan untuk melihat bentuk algoritma yang dapat mempelajari bentuk data yang diberikan. Bentuk data yang akan digunakan pada penelitian ini adalah bentuk data *time series*, dimana kedua model algoritma akan mempelajari pola jumlah pendaftar dari tahun sebelumnya dan meramalkan berdasarkan pola tersebut. Pembagian data dilakukan dengan menggunakan metode *K-Fold Cross-Validation* untuk menemukan jumlah perbandingan *train set* dan *test set* yang paling optimal untuk kedua algoritma. Setelah dilakukan pembentukan model dengan algoritma-algoritma tersebut, akan dilakukan perbandingan hasil berdasarkan nilai RMSE terkecil dan menentukan model algoritma yang terbaik berdasarkan nilai RMSE tersebut.

#### **3.2 Analisis Data**

Pada subbab ini akan menjelaskan tentang data yang akan digunakan untuk mengetahui prediksi jumlah pendaftar calon mahasiswa baru. Data yang

digunakan untuk memprediksi jumlah pendaftar merupakan data pendaftar calon mahasiswa baru yang diolah menjadi dataset yang dapat digunakan untuk pemodelan algoritma. Adapun sumber data dan keterangan lain dari data akan dijelaskan pada subbab berikut ini.

### 3.2.1 Sumber Data

Sumber data yang dibutuhkan dalam penelitian ini yang merupakan penentu metode serta analisis pengolahan data diperoleh dari pusat akademik (BAAK) Institut Teknologi Del. Data yang diperoleh dari pusat akademik Institut Teknologi Del adalah data informasi peserta calon mahasiswa baru yang mendaftar melalui jalur PMDK, USM 1, USM 2, USM 3, USM 4, dan jalur UTBK. Adapun informasi yang didapatkan dari data tersebut adalah nama peserta, alamat peserta, nama orang tua peserta, asal sekolah peserta, prodi pilihan peserta baik pilihan 1 maupun pilihan 2.

**Tabel 3. 1 Informasi yang diperoleh dari BAAK IT Del**

<b>Variabel</b>	<b>Type data</b>	<b>Deskripsi</b>
No Pendaftar	Integer	No peserta pendaftar calon mahasiswa baru
Nama Siswa	String	Nama peserta calon mahasiswa baru
Asal Sekolah	String	Sekolah asal peserta pendaftar calon mahasiswa baru
Pilihan I	Categorical	Jurusan pilihan pertama calon peserta
Pilihan II	Categorical	Jurusan pilihan kedua calon peserta
Jenis Kelamin	Categorical	Jenis kelamin pendaftar calon mahasiswa baru
No Telepon	Integer	Nomor telepon dari peserta calon mahasiswa baru

### 3.2.2 Seleksi Data

Dalam penelitian ini akan dilakukan penyeleksian data yang akan digunakan dalam prediksi. Seleksi data ini dilakukan untuk membentuk dataset yang akan

digunakan dalam membentuk model dan prediksi. Berdasarkan data pendaftar yang telah diterima, data yang digunakan adalah tahun pendaftar dan data asal sekolah peserta.

### 3.3 Preprocessing Data

Preprocessing data adalah suatu teknik yang dilakukan untuk mengolah data dengan tujuan data tersebut siap digunakan pada pemodelan dengan algoritma yang akan digunakan. Pada penelitian ini tahapan preprocessing data yang dilakukan peneliti adalah sebagai berikut.

#### 3.3.1 Data Integration

Pada tahapan ini peneliti melakukan penggabungan data jumlah pendaftar yang diperoleh dari jalur seluruh jalur yang dibuka oleh institusi dan data pendaftar dari tahun 2016 sampai 2022. Berdasarkan penggabungan data ini dihasilkan dataset yang berisi nama sekolah dan jumlah pendaftar dari asal sekolah untuk setiap tahunnya. Berdasarkan proses data *integration* yang telah dilakukan maka diperoleh dataset dengan tiga atribut sebagai berikut ini :

**Tabel 3. 2 Atribut pada dataset**

Atribut	Type Data	Deksripsi
Nama Sekolah	Character	Nama asal sekolah yang mendaftar
Jumlah Pendaftar pada tahun 2016	Numerik	Jumlah pendaftar pada tahun 2016
Jumlah Pendaftar pada tahun 2017	Numerik	Jumlah pendaftar pada tahun 2017
Jumlah Pendaftar pada tahun 2018	Numerik	Jumlah pendaftar pada tahun 2018
Jumlah Pendaftar pada tahun 2019	Numerik	Jumlah pendaftar pada tahun 2019
Jumlah Pendaftar pada tahun 2020	Numerik	Jumlah pendaftar pada tahun 2020

Jumlah Pendaftar pada tahun 2021	Numerik	Jumlah pendaftar pada tahun 2021
Jumlah Pendaftar pada tahun 2022	Numerik	Jumlah pendaftar pada tahun 2022

Sehingga Dataset yang dihasilkan adalah sebagai berikut:

**Tabel 3. 3 Dataset hasil integration**

Nama Sekolah	Jumlah Pendaftar						
	2016	2017	2018	2019	2020	2021	2022
SMA Bintang Timur Balige	70	54	53	110	46	48	53
SMA Bintang Timur Pematang Siantar	30	19	6	7	18	8	2
SMK Bintang Timur Pematang Siantar	7	6	-	1	2	1	1

### 3.3.2 Data Cleaning

Pada tahapan ini data yang telah digabungkan dilakukan pembersihan data. Pembersihan data atau data cleaning yang dilakukan oleh peneliti adalah *missing value*. *Missing value* dilakukan disaat adanya penemuan tidak ada data/nilai yang terdeteksi untuk suatu atribut atau fitur pada beberapa sampel pengamatan di dataset. Adanya *missing value* dapat memberikan pengaruh pada hasil prediksi. *Missing value* dapat diatasi dengan beberapa teknik, diantaranya yaitu menghapus kolom atau baris yang memiliki *missing value* dan mengisi *missing value*.

Pada penelitian ini, *missing value* dilakukan dengan memberikan nilai default di dalam data yang kosong. Nilai default yang diberikan adalah 0. Pada dataset yang dimiliki, kita bisa menemukan atribut yang memiliki nilai *missing value* adalah data yang memiliki tanda strip (-). Atribut-atribut yang memiliki nilai *missing value* antara lain adalah jumlah pendaftar 2016, jumlah pendaftar 2017, jumlah pendaftar 2018, jumlah pendaftar 2019, jumlah pendaftar 2020, jumlah pendaftar 2021, dan jumlah pendaftar 2022.

Tabel 3. 4 Dataset sebelum dilakukan missing value

Jumlah Pendaftar 2016	Jumlah Pendaftar 2017	Jumlah Pendaftar 2018	Jumlah Pendaftar 2019	Jumlah Pendaftar 2020	Jumlah Pendaftar 2021	Jumlah Pendaftar 2022
18	15	6	5	12	16	1
16	40	39	19	17	22	11
32	38	18	23	28	21	7
-	-	-	2	4	-	-
28	24	63	38	45	44	32
22	45	17	12	6	10	6
1	-	-	-	-	-	-
2	-	12	3	-	2	3
-	1	-	-	-	-	12
-	2	-	2	-	-	2
1	4	1	1	1	4	-
3	1	-	2	1	2	-
1	1	1	1	6	1	-
53	48	58	28	17	24	8
1	3	2	3	-	2	8
1	-	-	1	-	-	-
28	43	68	6	2	21	4

Tabel 3. 5 Dataset sesudah dilakukan missing value

Jumlah Pendaftar 2016	Jumlah Pendaftar 2017	Jumlah Pendaftar 2018	Jumlah Pendaftar 2019	Jumlah Pendaftar 2020	Jumlah Pendaftar 2021	Jumlah Pendaftar 2022
18	15	6	5	12	16	1
16	40	39	19	17	22	11
32	38	18	23	28	21	7
0	0	0	2	4	0	0
28	24	63	38	45	44	32
22	45	17	12	6	10	6
1	0	0	0	0	0	0
2	0	12	3	0	2	3
0	1	0	0	0	0	12
0	2	0	2	0	0	2
1	4	1	1	1	4	0
3	1	0	2	1	2	0
1	1	1	1	6	1	0
53	48	58	28	17	24	8
1	3	2	3	0	2	8
1	0	0	1	0	0	0
28	43	68	6	2	21	4

### 3.3.3 Data Encoding

Pada tahapan ini peneliti melakukan pengubahan variabel teks ke dalam bentuk numerik dengan tujuan agar variabel tersebut dapat diolah dalam pembelajaran mesin. Pada data yang ada, variabel yang diubah adalah variabel nama sekolah. Pengubahan bentuk data teks ini dilakukan dengan menggunakan teknik encoding, dimana data nama sekolah diubah ke dalam bentuk numerik 1,2,3,..., dst.

### 3.4 Analisis Algoritma SVR

SVR merupakan penerapan support vector machine (SVM) pada kasus regresi. Dalam kasus regresi output yang digunakan berupa bilangan riil atau continue. Dalam mengatasi permasalahan overfitting SVR dapat digunakan sehingga mendapatkan nilai performansi yang bagus. Model SVR sering digunakan untuk meminimalkan jumlah square error, yang mana SVR ini merupakan teori yang diadaptasi dari teori machine learning yang sudah digunakan untuk mengatasi masalah klasifikasi.

#### Perhitungan SVR

Untuk melakukan perhitungan terdapat beberapa langkah yang perlu diperhatikan antara lain :

1. Menginisialisasi awal untuk nilai  $\alpha$ , C, epsilon, gamma, lambda:

**Tabel 3. 6 Inisialisasi nilai parameter**

Inisialisasi Parameter	
C	1
alpha	0,5
epsilon	0,1
gamma	0,167
lambda	0,5
alpha star	1

Bias	0,5
------	-----

## 2. Masukkan data uji

Jumlah data yang digunakan untuk hitungan manual yaitu 5 data, dimana 4 data sebagai data latih dan 1 sebagai data uji dan jumlah variabel yang digunakan yaitu 6 variabel.

Data yang digunakan dapat dilihat pada tabel berikut:

**Tabel 3. 7 Data Uji**

<b>Nama Sekolah</b>	Jumlah Pendaftar 2016 (X1)	Jumlah Pendaftar 2017 (X2)	Jumlah Pendaftar 2018 (X3)	Jumlah Pendaftar 2019 (X4)	Jumlah Pendaftar 2020 (X5)	Jumlah Pendaftar 2021 (X6)	Jumlah Pendaftar 2022 (Y)
SMA Kampus FKIP HKBP Nomensen	0,29	0,00	0,00	0,08	0,20	0,18	0,00
SMA Kristen Kalam Kudus Pematang Siantar	0,19	1,00	0,58	0,47	0,32	0,35	0,32
SMA Budi Mulia Pematang Siantar	1,00	0,92	0,21	0,58	0,59	0,32	0,19
SMA Yadika 6	0,00	0,02	0,19	0,00	0,00	0,00	0,15
SMA Bintang Timur Pematang Siantar	0,79	0,36	1,00	1,00	1,00	1,00	1,00

Data yang digunakan adalah :

**Tabel 3. 8 Data Uji**

	<b>X1</b>	<b>X2</b>	<b>X3</b>	<b>X4</b>	<b>X5</b>	<b>X6</b>	<b>Y</b>
	<b>A1</b>	<b>A2</b>	<b>A3</b>	<b>A4</b>	<b>A5</b>	<b>A6</b>	<b>Y</b>
A1	0,29	0,00	0,00	0,08	0,20	0,18	0,00
A2	0,19	1,00	0,58	0,47	0,32	0,35	0,32
A3	1,00	0,92	0,21	0,58	0,59	0,32	0,19



3. Pada metode kernel, data tidak dipresentasikan secara individual, melainkan melakukan perbandingan antara sepasang data. Setiap data akan dibandingkan dengan dirinya sendiri dan data lainnya. Maka untuk data latih yang berjumlah 3 data dan data uji yang berjumlah 2 data, maka hasil perbandingan dapat dilihat pada tabel berikut:

**Tabel 3. 9 Pasangan Data**

	A1	A2	A3	A4	A5	A6
A1	K(A1,A1)	K(A1,A2)	K(A1,A3)	K(A1,A4)	K(A1,A5)	K(A1,A6)
A2	K(A2,A1)	K(A2,A2)	K(A2,A3)	K(A2,A4)	K(A2,A5)	K(A2,A6)
A3	K(A3,A1)	K(A3,A2)	K(A3,A3)	K(A3,A4)	K(A3,A5)	K(A3,A6)

Semua data dihitung dengan cara yang sama, baris x kolom dengan sehingga dikarenakan adanya 3 data uji, maka diperoleh matriks 3 x 6. Pada tugas akhir ini kernel yang digunakan yaitu kernel radial basis function, sehingga:

$$k(x, x') = \exp(-\gamma \|x - y\|^2) \quad (\text{Pers. 3.4.1})$$

Maka untuk perhitungan dengan kernel rbf yaitu:

$$\mathbf{K(A1,A1)} = \exp(-0,167\|0,29-0,29\|^2) + \exp(-0,167\|0,19-0,19\|^2) + \exp(-0,167\|1,00-1,00\|^2) = 3$$

$$\mathbf{K(A1,A2)} = \exp(-0,167\|0,29-0\|^2) + \exp(-0,167\|0,19-1\|^2) + \exp(-0,167\|1,00-0,92\|^2) = 3,881139032$$

maka untuk hasil perhitungan dengan menggunakan kernel rbf yaitu:

**Tabel 3. 10 Hasil Perhitungan RBF**

	A1	A2	A3	A4	A5	A6
A1	3	2,88120583	2,861995465	2,95062577	2,96814757	2,99798134
A2	2,881139302	3	2,885417662	2,93391731	2,900938976	2,868062698
A3	2,855984901	2,885417662	3	2,968299609	2,95227962	2,977779196

4. Menghitung matriks dengan rumus:

$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2) \quad (\text{Pers. 3.4.2})$$

Dimana:

$D_{ij}$  = elemen matriks ke-ij

$y_i$  = kelas data ke-i

$y_j$  = kelas data ke-j

$\lambda$  = batas teoritis yang akan diturunkan

Contoh perhitungan:

$$D_{11} = (0) (0) (4+0,25) = 0$$

Sehingga didapatkan hasil perhitungan matriks Hessian adalah sebagai berikut:

**Tabel 3. 11 Hasil Perhitungan Matriks Hessian**

	<b>A1</b>	<b>A2</b>	<b>A3</b>	<b>A4</b>	<b>A5</b>	<b>A6</b>
<b>A1</b>	0	0	0	0	0	0
<b>A2</b>	0,42302866 45	0,4352	0,423466768 6	0,428433132 3	0,4250561511	0,42168962 0
<b>A3</b>	0,14822605 49	0,14928857 76	0,153425	0,152280615 9	0,1517022943	0,15262282 9

5. Mencari nilai error menggunakan rumus:

$$E_i = \sum_{j=i}^i \alpha_j D_{ij} \quad (\text{Pers. 3.4.3})$$

Dimana  $E_i$  = nilai error data ke-i

$$E_1 = 0 + 0 + 0 + 0 + 0 + 0 = 0$$

Maka didapatkan setiap nilai error data:

**Tabel 3. 12 Hasil Perhitungan Nilai Error**

	<b>Error</b>
--	--------------

A1	0
A1	1,278437168
A3	0,4537726859

6. Menghitung nilai delta alpha

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\} \quad (\text{Pers. 3.4.4})$$

$$\delta\alpha_i = \min\{\max[0,167(1 - 0), -0,5], 1 - 0,5\}$$

$$\delta\alpha_i = \min\{\max(0,167, -0,5), 0,5\}$$

$$\delta\alpha_i = \min\{0,167, 0,5\}$$

$$\delta\alpha_i = 0,167$$

Hasil perhitungan delta alpha:

**Tabel 3. 13 Hasil Perhitungan Delta Alpha**

	<b>Delta Alpha</b>
<b>A1</b>	0,167
<b>A2</b>	-0,04649900706
<b>A3</b>	0,09121996145
<b>A4</b>	0,1195388175

Karena nilai maksimum delta alpha adalah 0,167 dan lebih dari epsilon (0,1) maka iterasi berlanjut.

7. Menghitung nilai  $\alpha$  baru dengan menggunakan rumus:

$$\alpha_i = \alpha_i + \delta_i\alpha \quad (\text{Pers. 3.4.5})$$

Contoh perhitungan:

$$\alpha_1 = 0,5 + 0,167 = 0,667$$

$$\alpha_2 = 0,5 + (-0,4649900706) = 0,035$$

**Tabel 3. 14 Hasil Perhitungan Nilai Alpha Baru**

Alpha	
A1	0,667
A2	0,4535009929
A3	0,5912199615

8. Melakukan pencarian nilai w dengan rumus:

$$w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) \quad (\text{Pers. 3.4.6})$$

Contoh perhitungan:

$$W1 = 0,667 - 1 = -0,333$$

$$W2 = 0,4535009929 - 1 = -0,5464990071$$

Maka hasil perhitungan w adalah sebagai berikut:

**Tabel 3. 15 Hasil Perhitungan Nilai Alpha Star**

Alpha Star	
A1	-0,333
A2	-0,5464990071
A3	-0,4087800386

Setelah nilai w ditemukan maka dapat dilakukan perhitungan dalam fungsi pada persamaan () yaitu:

$$y = \sum_{i=1}^n (\alpha_i - \alpha_i^*) K(x_i, x) + b \quad (\text{Pers. 3.4.7})$$

Nilai bias(b) merupakan nilai yang diambil secara random pada komputer. Misalkan inisialisasi nilai b = 0,5

Contoh perhitungan 2 data untuk melakukan testing:

Tabel 3. 16 Data Uji yang Ditesting

Nama Sekolah	Jumlah Pendaftar 2016 (X1)	Jumlah Pendaftar 2017 (X2)	Jumlah Pendaftar 2018 (X3)	Jumlah Pendaftar 2019 (X4)	Jumlah Pendaftar 2020 (X5)	Jumlah Pendaftar 2021 (X6)	Jumlah Pendaftar 2022 (Y)
SMA Kristen Kalam Kudus Pematang Siantar	0,19	1,00	0,58	0,47	0,32	0,35	0,32

Perhitungan kernel (*dot product* baris data testing dengan kolom data training), dengan nilai  $\alpha_i - \alpha_i^*$  yang telah didapatkan sebelumnya, maka diperoleh hasil menggunakan fungsi y sebagai berikut:

Tabel 3. 17 Hasil Alpha Star yang didapat sebelumnya

$\alpha_i - \alpha_i^*$	
A1	-0,333
A2	-0,5464990071
A3	-0,4087800386

Untuk data testing maka diperoleh hasil prediksi sebagai berikut:

$$y = (-0,333 * 0,19) + (-0,333 * 1,00) + (-0,333 * 0,58) + (-0,333 * 0,47) + (-0,333 * 0,32) + (-0,333 * 0,35) + 0,5 = 0,13603$$

Kemudian dilakukan proses denormalisasi, untuk mendapatkan nilai real kembali. Maka prediksi jumlah pendaftar calon mahasiswa baru tahun 2022 dari sekolah SMA Kalam Kudus Pematang Siantar adalah sebanyak 15 jumlah pendaftar calon mahasiswa.

### 3.5 Analisis Algoritma ANN Backpropagation

Backpropagation merupakan salah satu bentuk metode dalam *artificial neural network* (ANN) yang populer. Metode ini memiliki kemampuan yang unggul untuk belajar, beradaptasi akan suatu objek, dan memiliki toleransi terhadap *error*.

Pada prosesnya algoritma *Backpropagation* ini dimulai dari tahapan *feedforward propagation*, kemudian melakukan perhitungan *error* dari *output*, dan dilanjutkan tahapan *backpropagation*.

### 1. Inisialisasi Bias dan Bobot

Sebelum dilakukan pemodelan, perlu dilakukan inisialisasi bobot dan bias terlebih dahulu. Bobot dan bias dipilih secara acak yang diambil dari rentang 0 sampai 1, seperti yang tampak pada tabel berikut :

**Tabel 3. 18 Nilai bias dari input layer ke hidden layer**

(bias, <i>hidden layer</i> )	(,1)	(,2)	(,3)
(1, )	0,48	0,50	0,65

**Tabel 3. 19 Nilai bobot dari inputan layer ke hidden layer**

(input, <i>hidden layer</i> )	(,1)	(,2)	(,3)
(1, )	0,20	0,50	0,50
(2, )	0,15	0,20	0,40
(3, )	0,30	0,10	0,35
(4, )	0,40	0,25	0,15
(5, )	0,18	0,30	0,25
(6, )	0,22	0,45	0,30

Setelah bias dan bobot dari *input layer* ke *hidden layer* ditentukan selanjutnya kita menentukan nilai bias dan bobot dari *hidden layer* ke *output layer*.

**Tabel 3. 20 Nilai bias dari hidden layer ke output layer**

(bias, output)	(,1)
(2, )	0,50

Tabel 3. 21 Nilai bobot dari hidden layer ke output layer

(hidden layer, output)	(,1)
(1, )	0,60
(2, )	0,35
(3, )	0,50

## 2. Inisialisasi *Hyperparameter*

Setelah dilakukan inisialisasi *parameter* bobot dan bias, maka selanjutnya dilakukan inisialisasi *hyperparameter* seperti *learning rate*, *epoch*, dan target *error* (threshold). Nilai *hyperparameter* yang akan digunakan pada analisis ini adalah *learning rate*=0.1 , *epoch*=1, target error (*threshold*) = 0.3, dan jumlah neuron pada *hidden layer* =3 neuron.

## 3. Feedforward Propagation

Pada tahap ini akan dilakukan perhitungan dengan menjumlahkan nilai bobot dan bias yang telah ditentukan sebelumnya untuk memperoleh nilai keluaran yang dihasilkan di setiap layer berdasarkan nilai yang dikirim mulai dari input layer ke *hidden layer* dan kemudian dari *hidden layer* ke output layer. Berikut tahapan feed

forward propagation yang dilakukan:

### a. Feedforward Propagation dari Input Layer ke *Hidden layer*

Nilai data input pada baris kedua (data sekolah SMA Kalam Kudus Pematang Siantar) akan dikirimkan untuk menghasilkan keluaran pada output layer, yang mana sebelum sampai ke output layer, sinyal yang dikirimkan dari input layer akan terlebih dahulu melalui *hidden layer*. Untuk itu pada tahap ini akan dihitung berapa sinyal yang diterima oleh masing-masing unit yang ada di *hidden layer* dengan menggunakan persamaan 2.11.1 .

$$[Z_{in1} \ Z_{in2} \ Z_{in3}] = [X_1 \ X_2 \ X_3 \ X_4 \ X_5 \ X_6] \cdot \begin{bmatrix} W_{11} & W_{12} & W_{13} \\ W_{21} & W_{22} & W_{23} \\ W_{31} & W_{32} & W_{33} \\ W_{41} & W_{42} & W_{43} \\ W_{51} & W_{52} & W_{53} \\ W_{61} & W_{62} & W_{63} \end{bmatrix} + [b_{11} \ b_{12} \ b_{13}]$$

$$\begin{bmatrix} 0.20 & 0.50 & 0.50 \\ 0.15 & 0.20 & 0.40 \\ 0.30 & 0.10 & 0.35 \\ 0.40 & 0.25 & 0.15 \\ 0.18 & 0.30 & 0.25 \\ 0.22 & 0.45 & 0.30 \end{bmatrix}$$

$$[Z_{in1} \ Z_{in2} \ Z_{in3}] = [0.19 \ 1.00 \ 0.58 \ 0.47 \ 0.32 \ 0.35].$$

$$+ [0.48 \ 0.50 \ 0.65]$$

$$[Z_{in1} \ Z_{in2} \ Z_{in3}] = [0.685 \ 0.724 \ 0.954] + [0.48 \ 0.50 \ 0.65]$$

$$[Z_{in1} \ Z_{in2} \ Z_{in3}] = [1.165 \ 1.224 \ 1.604]$$

Setelah dilakukan penghitungan sinyal-sinyal inputan terbobot, selanjutnya akan dilakukan penghitungan fungsi aktivasi dari sinyal inputan terbobot. Untuk melakukan perhitungan fungsi aktivasi digunakan persamaan 2.11.2 .

$$\partial([Z_{out1} \ Z_{out2} \ Z_{out3}]) = \left[ \frac{1}{1 + e^{-Z_{in1}}} \quad \frac{1}{1 + e^{-Z_{in2}}} \quad \frac{1}{1 + e^{-Z_{in3}}} \right]$$

$$\partial([Z_{out1} \ Z_{out2} \ Z_{out3}]) = \left[ \frac{1}{1 + e^{-(1.165)}} \quad \frac{1}{1 + e^{-(1.224)}} \quad \frac{1}{1 + e^{-(1.604)}} \right]$$

$$\partial([Z_{out1} \ Z_{out2} \ Z_{out3}]) = [0.762 \ 0.773 \ 0.832]$$

$$[Z_{out1} \ Z_{out2} \ Z_{out3}] = [0.762 \ 0.773 \ 0.832]$$

#### b. Feedforward Propagation dari *Hidden layer* ke *Output Layer*

Dari hasil perhitungan *feedforward propagation* dari *input layer* ke *hidden layer*, akan dilanjutkan dengan menghitung nilai keluaran untuk masing-masing unit di *hidden layer* yang di kirimkan ke *output layer*. Persamaan yang digunakan untuk perhitungan ini adalah persamaan 2.11.3 .

$$[Y_{in}] = [Z_{out1} \ Z_{out2} \ Z_{out3}] \cdot \begin{bmatrix} W_{11} \\ W_{21} \\ W_{31} \end{bmatrix} + [b_{21}]$$

$$[Y_{in}] = [0.762 \ 0.773 \ 0.832] \cdot \begin{bmatrix} 0.60 \\ 0.35 \\ 0.50 \end{bmatrix} + [0.50]$$

$$[Y_{in}] = [1.144] + [0.50]$$

$$[Y_{in}] = [1.644]$$

Selanjutnya, sinyal yang diterima dari *hidden layer* ke *output layer* dihitung kembali dengan menggunakan fungsi aktivasi sigmoid seperti berikut, yang mana hasil dari perhitungan ini akan menjadi nilai y atau prediksi jumlah pendaftar terhadap data yang dilatih. Perhitungan ini menggunakan persamaan 2.11.4 seperti berikut :



$$\partial([Y_{out}]) = \left[ \frac{1}{1 + e^{-Y_{in}}} \right] = \left[ \frac{1}{1 + e^{-(1.644)}} \right] = 0.838$$

$$[Y_{out}] = 0.838$$

### c. Menghitung Nilai Error / Loss

Berdasarkan perhitungan *feedforward propagation* yang dilakukan, diperoleh nilai prediksi jumlah pendaftar untuk data yang dipilih adalah 0.838. Selanjutnya dengan menggunakan nilai prediksi yang dihasilkan, akan dilakukan perhitungan nilai *error* dengan menggunakan *Mean Square Error* (MSE). Bila nilai MSE yang diperoleh lebih kecil dengan nilai target error yang telah ditentukan sebelumnya, maka nilai prediksi tersebut dapat dipastikan akan menjadi hasil akhir. Namun, jika nilai MSE yang diperoleh lebih besar dari nilai target error, maka akan dilakukan pembaruan bobot dan bias dengan menggunakan *backpropagation*.

Dengan menggunakan persamaan 2.11. 5 , nilai *loss* dapat ditentukan seperti berikut ini.

$$\begin{aligned} loss &= \frac{1}{2} (Y_{out} - aktual)^2 = \frac{1}{2} (0.838 - 0.32)^2 \\ &= \frac{1}{2} (0.518)^2 \\ &= \frac{1}{2} (0.268) \\ &= 0.134 \end{aligned}$$

## 4. Backpropagation

Karena nilai *loss* < nilai target *error*, yaitu  $0.134 < 0.3$ , maka untuk tahapan *backpropagation* tidak perlu dilakukan. Sehingga nilai prediksi yang dilakukan pada tahapan sebelumnya merupakan hasil akhir prediksi.

### 3.6 Analisis Evaluasi Model

Setelah dilakukan analisis terhadap kedua model, selanjutnya akan dilakukan evaluasi terhadap kedua model tersebut. Pada evaluasi model akan menggunakan model RMSE (*Root Mean Square Error*). RMSE digunakan untuk mengukur tingkat kesalahan model dalam prediksi suatu nilai numerik. Seperti yang telah disampaikan pada **subbab 2.13.2** mengenai RMSE, bahwa semakin kecil nilai

RMSE yang dihasilkan model, maka nilai prediksi semakin akurat. Evaluasi model RMSE memiliki beberapa kelebihan diantaranya adalah :

1. RMSE memberikan hasil dalam satuan yang sama dengan target, sehingga mudah dipahami dan diinterpretasikan
2. RMSE memperhitungkan semua nilai residual (perbedaan nilai antara nilai aktual dan nilai prediksi) sehingga lebih sensitif terhadap perbedaan besar antara nilai aktual dan nilai prediksi.
3. RMSE menunjukkan performa model secara keseluruhan, bukan hanya performa pada titik tertentu.

### 3.7 Hipotesis Penelitian

Berdasarkan hasil literatur dan hasil analisis dari beberapa algoritma yang sudah dijelaskan diatas, maka peneliti dapat menyusun hipotesis sebagai berikut:

**Hipotesis :** Algoritma Artificial Neural Network (ANN) atau Jaringan Syaraf Tiruan merupakan suatu sistem pemrosesan informasi dengan suatu karakteristik menyerupai sistem saraf pada manusia yang dapat memecahkan masalah dengan melakukan training data yang besar, dan juga ANN memiliki kemampuan untuk mentoleransi kesalahan sehingga dapat menghasilkan prediksi yang baik (Putra & Ulfa Walmi, 2020). Berdasarkan penelitian yang dilakukan oleh (Abdel-Sattar et al., 2021) dengan judul penelitian ” Application of artificial neural network and support vector regresion in predicting mass of ber fruits (*Ziziphus mauritiana* Lamk) based on fruit axial dimensions ” dengan hasil penelitian yaitu algoritma ANN mendapatkan hasil yang lebih baik dibandingkan dengan algoritma Support Vector Regression dengan nilai RMSE 1.8479 dan nilai  $R^2$  dengan nilai 0.9771 pada dataset yang memiliki tingkat kompleksitas dan atribut yang kompleks. Sehingga dugaan pada penelitian ini algoritma Artificial Neural Network juga memiliki performa terbaik dalam melakukan prediksi jumlah pendaftar di Institut Teknologi Del.

### 3.7.1 Perumusan H0 dan H1

Berdasarkan Hipotesis yang sudah ditentukan, maka Hipotesis Awal (H0) dan Hipotesis Alternatif (H1) adalah sebagai berikut:

**H0** : model Algoritma Artificial Neural Network lebih baik dibandingkan model Support Vector Regression.

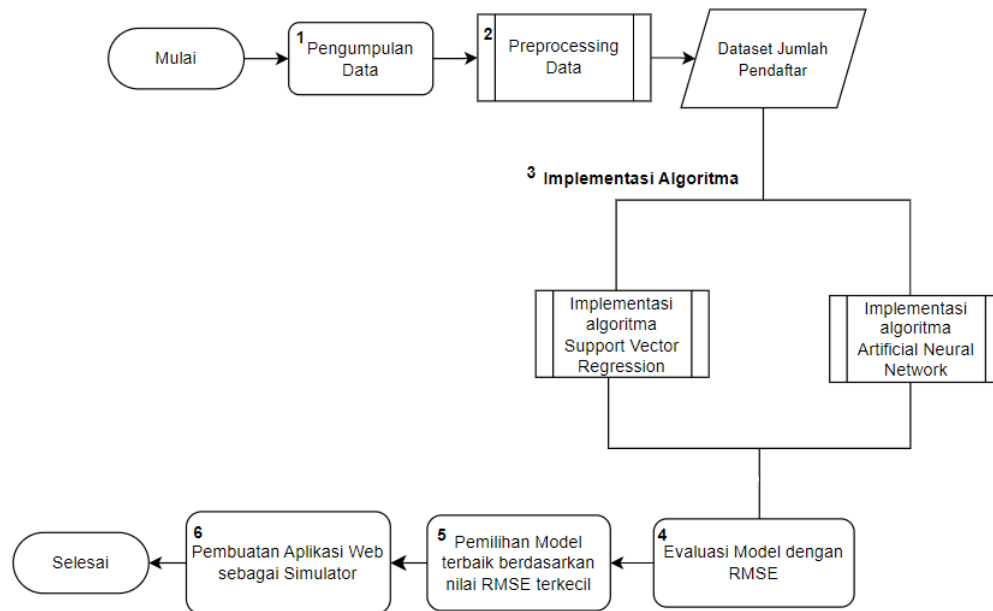
**H1** : model Algoritma Artificial Neural Network tidak lebih baik dibandingkan model Support Vector Regression.

## BAB 4

### PERANCANGAN

Pada bab ini akan dijelaskan mengenai rancangan penelitian yang dilakukan dalam melakukan pengerjaan Tugas Akhir.

#### 4.1 Rancangan Desain Penelitian



**Gambar 4. 1 Rancangan desain penelitian yang dilakukan**

Berikut penjelasan terkait rancangan penelitian yang akan dilakukan :

#### 1. Pengumpulan data

Pada penelitian ini data yang digunakan diperoleh dari kantor pusat akademik (BAAK) Institut Teknologi Del. Data yang diperoleh dari BAAK Institut Teknologi Del adalah data pendaftar calon mahasiswa baru yang mendaftar pada seluruh jalur pendaftaran yang dibuka institusi. Data pendaftar diperoleh secara terpisah berdasarkan jalur pendaftarannya. Data yang dikumpulkan merupakan data mulai dari 2016 sampai dengan 2022.

#### 2. Data Preprocessing

Pada tahapan ini data yang telah dikumpulkan akan dilakukan tahapan preprocessing data untuk mempersiapkan data sebelum digunakan dalam implementasi algoritma. Setelah melewati tahapan ini akan menghasilkan dataset yang telah siap untuk digunakan dalam pemodelan.

### 3. Implementasi Algoritma

Pada tahapan ini data akan dimasukkan ke dalam bentuk pemodelan algoritma. Pemodelan algoritma akan dilakukan secara terpisah antara pemodelan algoritma *Support Vector Regression* dan pemodelan algoritma *Artificial Neural Network*. Implementasi setiap algoritma akan dijelaskan pada subbab berikutnya.

### 4. Evaluasi Model

Pada tahapan ini akan dilakukan evaluasi terhadap kedua model algoritma yang telah dibuat pada tahapan sebelumnya. Evaluasi model ini akan dilakukan pencarian nilai RMSE dan akurasi dari setiap model algoritma.

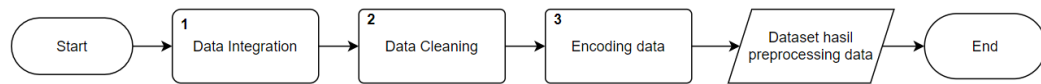
### 5. Pemilihan Algoritma Terbaik

Pada tahapan ini akan dilakukan pemilihan algoritma terbaik dari antara algoritma *Support Vector Regression* dan *Artificial Neural Network*. Penentuan algoritma terbaik ini dilihat berdasarkan nilai RMSE terkecil. Seperti yang dijelaskan pada **subbab 2.13.2**, bahwa semakin kecil nilai RMSE yang dihasilkan maka model tersebut akan semakin baik.

### 6. Pembuatan Simulator

Setelah dilakukan pembentukan model dan penentuan model algoritma terbaik, selanjutnya akan dilakukan pembangunan aplikasi web yang akan digunakan sebagai simulator. Pembentukan aplikasi ini dilakukan dengan menggunakan bahasa pemrograman *python* dan framework *flask*. Pada aplikasi akan terdapat fitur SVR dan ANN, dimana pada kedua fitur akan melakukan prediksi dengan bentuk modelnya masing-masing.

## 4.2 Rancangan Pre-Processing Data



**Gambar 4. 2 Rancangan pelaksanaan preprocessing data**

Berdasarkan **Gambar 4.2** di atas, terlihat bahwa rancangan preprocessing data yang dilakukan dengan penjelasan seperti berikut :

### 1. Data Integration

Pada tahapan ini data yang telah dikumpulkan akan digabungkan seperti pada **subbab 3.3.1**. Penggabungan data ini dilakukan secara manual oleh peneliti dan akan menghasilkan bentuk atribut seperti pada **tabel 3.2** sehingga bentuk dataset yang dihasilkan adalah seperti pada **tabel 3.3**.

### 2. Data Cleaning

Pada tahapan ini, peneliti melakukan data cleaning dengan menggunakan teknik *missing value*. Seperti yang sudah dijelaskan pada **subbab 3.3.2**, bahwa data cleaning dilakukan dengan melakukan pengisian nilai 0 pada data yang tidak memiliki nilai maupun bertanda strip (-).

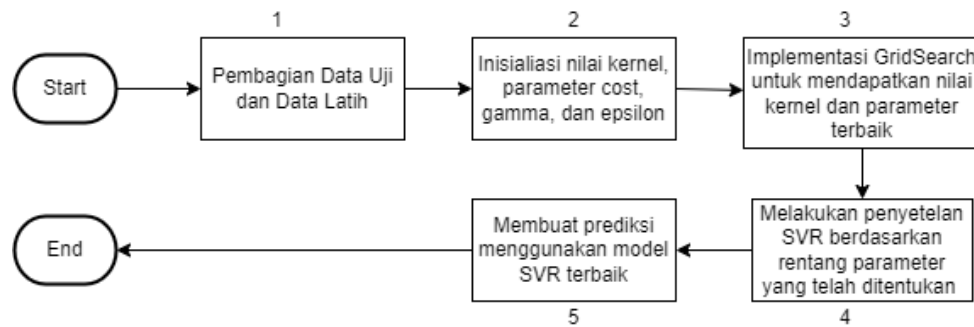
### 3. Encoding Data

Pada tahapan encoding data ini, peneliti melakukan perubahan bentuk data nama sekolah dari bentuk teks ke bentuk numerik dengan tujuan agar nama sekolah dapat digunakan dalam pemodelan. Pelaksanaan *encoding* data ini akan dijelaskan pada **subbab 5.3.1** dan hasil akan dijelaskan pada **subbab 6.1**. Hasil akhir yang akan diperoleh dari tahapan *encoding* data adalah dataset baru yang sudah dapat digunakan dan dipelajari oleh model algoritma.

## 4.3 Rancangan Implementasi Algoritma

Berikut ini adalah rancangan implementasi algoritma yang digunakan dalam penelitian.

### 4.3.1 Rancangan Implementasi Algoritma SVR



**Gambar 4. 3 Flow Algoritma Support Vector Regression**

Berdasarkan **Gambar 4.3** diatas, langkah-langkah pengerjaan algoritma *Support Vector Regression* dilakukan dengan beberapa tahapan dengan rincian masing-masing tahapan adalah sebagai berikut :

1. Melakukan pembagian dataset

Dataset yang digunakan akan dilakukan pembagian untuk data latih dan data uji. Pada proses pembagian dataset ini digunakan menggunakan teknik *K-Fold Cross Validation*, yang mana untuk rentang nilai k yang digunakan yaitu mulai nilai k=2 sampai nilai k=10.

2. Melakukan inisialisasi kernel dan parameter

Kernel yang digunakan pada SVR terdiri dari 4 kernel yaitu linear, RBF, Sigmoid, dan polynomial. Sedangkan untuk inisialisasi Parameter SVR terdiri dari beberapa parameter  $\epsilon$  (nilai epsilon untuk zona yang masih dapat diterima), C (cost),  $\gamma$  untuk masukan fungsi kernel yang digunakan dan jumlah iterasi maksimum. Untuk nilai parameter cost akan diinisialisasikan pada nilai 0,1, 1, 10, 100, kemudian parameter epsilon pada nilai 0,01, 0,1, 1 dan untuk parameter gamma berada pada nilai 0,001, 0,01, 0,1, dan 1 (Sepri & Fauzi, 2020).

3. Melakukan pencarian kernel dan parameter terbaik dengan menggunakan *Gridsearch*

*Gridsearch* akan digunakan untuk menemukan parameter dan kernel terbaik berdasarkan nilai yang telah diinisialisasikan untuk mendapatkan nilai error yang paling rendah.

4. Melakukan pemodelan dengan algoritma SVR

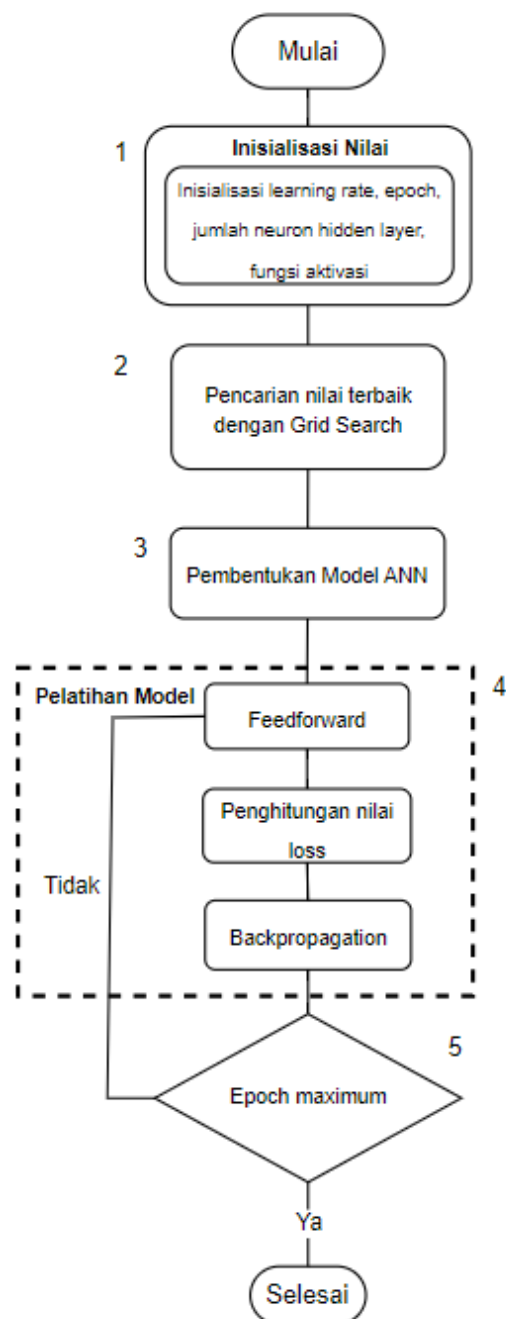
Kernel dan parameter terbaik yang telah didapatkan menggunakan *GridSearch* kemudian digunakan untuk melakukan pemodelan menggunakan algoritma SVR

5. Melakukan prediksi menggunakan SVR

Model terbaik telah didapatkan dan dapat digunakan untuk melakukan prediksi.



#### 4.3.2 Rancangan Implementasi Algoritma ANN Backpropagation



Gambar 4. 4 Flow Algoritma ANN Backpropagation

Berdasarkan Gambar di atas, langkah implementasi algoritma Backpropagation dilakukan dengan beberapa tahapan dengan rincian dari masing – masing tahapan adalah sebagai berikut:

#### 1. Inisialisasi Nilai

Pada tahapan ini, untuk memulai tahapan pemodelan algoritma perlu dilakukan inisialisasi nilai *hyperparameter* seperti nilai *learning rate*, *epoch*, jumlah neuron *hidden layer*, dan fungsi aktivasi. Beberapa *hyperparameter*, seperti fungsi aktivasi, jumlah neuron pada *hidden layer*, dan *learning rate*, akan dilakukan pencarian nilai terbaik diantara beberapa nilai yang akan digunakan dengan menggunakan teknik *gridsearch*. Berikut adalah nilai-nilai *hyperparameter* yang diinisialisasikan untuk model.

**Tabel 4. 1 Inisialisasi Hyperparameter ANN**

<i>Hyperparameter</i>	Nilai
Learning rate	0.01, 0.001
Jumlah neuron <i>hidden layer</i>	12, 16, 24, 32, 64, 128
Fungsi Aktivasi <i>hidden layer</i>	Sigmoid, Tanh, ReLu

Jika dilihat pada **Tabel 4.1** di atas, bahwa *hyperparameter learning rate*, jumlah neuron pada *hidden layer*, dan fungsi aktivasi didefinisikan dengan beberapa nilai. *Learning rate* diinisialisasikan dengan nilai 0.01 dan 0.001 ditentukan dari nilai learning rate yang pernah digunakan pada penelitian sebelumnya (Mahyunis, 2022). Berlandaskan dengan jurnal yang sama dengan *learning rate*, penggunaan fungsi aktivasi yang diinisialisasi juga berdasarkan penelitian sebelumnya, dimana peneliti menggunakan percobaan pada tiga fungsi aktivasi seperti ReLU, sigmoid, dan tahn. Dari keempat bentuk fungsi aktivasi tersebut, peneliti mencoba mencari fungsi aktivasi yang optimal digunakan pada bentuk dataset yang akan dimodelkan. Inisialisasi *hyperparameter* selanjutnya yang diinisialisasi adalah jumlah neuron yang akan digunakan pada *hidden layer*. Nilai-nilai tersebut diambil

dari penelitian yang sebelumnya (Rusdi et al., 2021) dan (Rizal, 2014) yang juga melakukan percobaan nilai epoch dengan nilai-nilai tersebut.

## 2. Pencarian nilai terbaik dengan *Gridsearch*

Pada tahapan ini, nilai-nilai yang sudah diinisialisasi pada tahapan sebelumnya akan diproses oleh library *GridSearch* untuk mencari nilai-nilai yang terbaik dari seluruh nilai yang telah diinisialisasi untuk setiap *hyperparameter*. Parameter yang akan diolah dengan teknik *gridsearch* adalah *learning rate*, jumlah neuron pada *hidden layer*, dan fungsi aktivasi.

## 3. Pembentukan Model

Pada tahapan ini dilakukan pembentukan model ANN yang mana model terdiri dari input layer, *hidden layer*, dan output layer. Dalam pembentukan model nilai-nilai *hyperparameter* yang telah diinisialisasikan pada tahapan sebelumnya dipanggil untuk digunakan dalam training model.

## 4. Pelatihan Model

Pada tahapan ini akan dilakukan pelatihan model sesuai dengan bentuk model yang dibuat dan nilai *hyperparameter* yang telah diinisialisasi. Di dalam pembentukan model, akan terjadi tahapan feedforward, perhitungan loss, dan backpropagation yang tidak terlihat secara langsung ketika model dilatih.

Pada tahapan feedforward nilai inputan data akan dimasukkan ke dalam input layer dan akan dihitung nilai output untuk setiap neuron dengan memperhitungkan bobot, bias, dan fungsi aktivasi. Nilai bobot dan bias pada proses ini akan dihasilkan secara otomatis oleh model. Kemudian setiap output akan disimpan untuk digunakan pada tahap berikutnya.

Pada tahapan perhitungan loss, nilai output yang dihasilkan pada tahapan feedforward akan dicari nilai *loss*-nya terhadap nilai target yang diinginkan. Nilai loss ini menggambarkan sejauh mana prediksi jaringan mendekati nilai target sebenarnya.

Pada tahapan backpropagation akan terjadi perhitungan gradien loss terhadap bobot dan bias yang dihasilkan langsung oleh model. Gradien ini mengindikasikan seberapa besar perubahan yang harus diterapkan pada bobot dan bias untuk mengurangi loss.

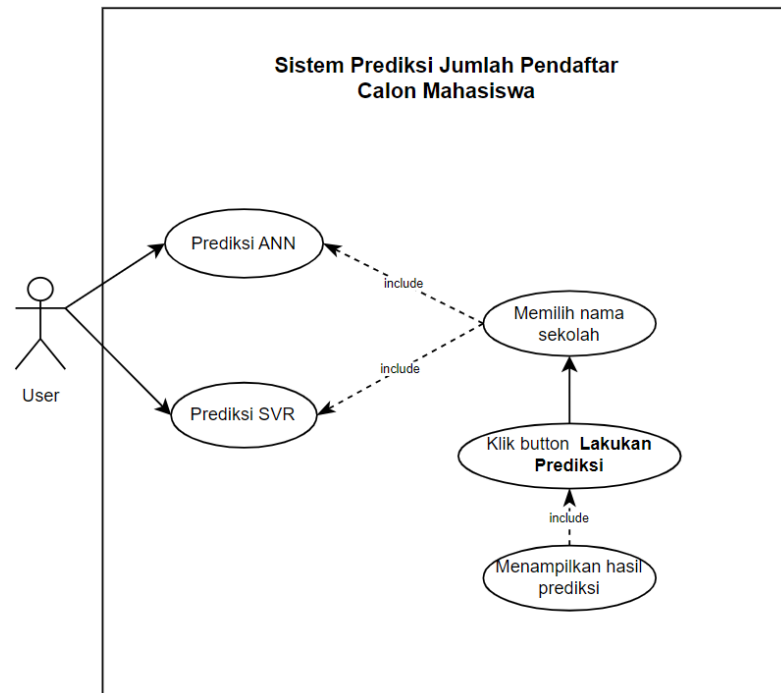
#### 5. Kondisi Waktu Berhenti

Pada tahapan ini, selama proses train akan dilakukan pengecekan nilai *epoch* dimana ketika iterasi yang terjadi selama proses *training* sudah mencapai nilai *epoch* yang ditentukan maka proses *training* akan dihentikan dan ketika belum mencapai nilai *epoch* maksimum maka *training* akan terus dilakukan

### 4.4 Rancangan Aplikasi Del Predict

Del Predict merupakan aplikasi web yang digunakan untuk melakukan prediksi terhadap jumlah pendaftar calon mahasiswa baru. Pada aplikasi akan terdapat dua fitur yaitu fitur SVR untuk melakukan prediksi dengan algoritma *support vector regression* dan fitur ANN untuk melakukan prediksi dengan algoritma *artificial neural network*. Pembangunan aplikasi ini akan menggunakan tools HTML dan Bootstrap dalam membuat halaman tampilan dan akan menggunakan python dalam membuat server.

#### 4.4.1 Use Case Diagram



**Gambar 4. 5 Use Case Diagram Aplikasi**

Pada **Gambar 4.5** merupakan gambaran fungsionalitas dari aplikasi web yang akan digunakan sebagai simulator sistem prediksi jumlah pendaftar calon mahasiswa baru. Tahapan yang terjadi pada use case diagram di atas akan dijelaskan di dalam tabel berikut berikut ini.

**Tabel 4. 2 Deskripsi Use Case Diagram**

No	Use Case	Deskripsi
1	Prediksi ANN	Use case ini menggambarkan bahwa aplikasi dapat melakukan prediksi dengan menggunakan algoritma ANN.
2	Prediksi SVR	Use case ini menggambarkan bahwa aplikasi dapat melakukan prediksi dengan menggunakan algoritma SVR.
3	Memilih nama sekolah	Use case ini menggambarkan bahwa aplikasi melakukan pemilihan nama sekolah yang akan

		diprediksi
4	Klik button <b>Lakukan Prediksi</b>	Use case ini menunjukkan bahwa setelah user memilih nama sekolah, user mengklik button <i>lakukan prediksi</i> untuk memulai proses prediksi.
5	Menampilkan Hasil Prediksi	Use case ini menggambarkan bawa aplikasi akan menampilkan hasil prediksi pada halaman fitur yang dipilih.

#### 4.4.2 Use Case Scenario

Pada subab ini akan menjelaskan mengenai prosedur dari sistem yang akan dibangun. Berikut ini adalah use case scenario dari aplikasi Del Predict.

**Tabel 4. 3 Use Case Scenario Prediksi ANN**

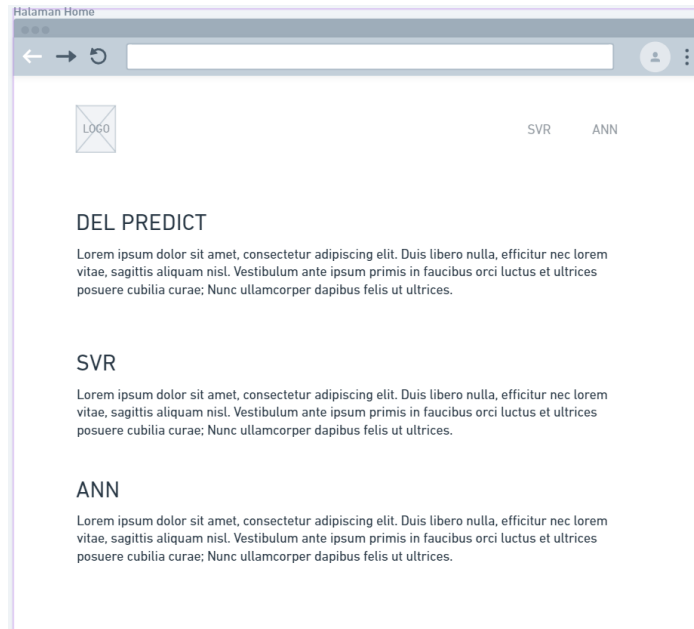
Use Case	:	Prediksi ANN
Deskripsi	:	Use case ini menggambarkan bahwa aplikasi dapat melakukan prediksi ANN
Aktor	:	User
Kondisi Awal	:	Sistem menampilkan halaman home Del Predict
Kondisi Akhir	:	Sistem menampilkan hasil prediksi
Skenario		
Aksi Aktor		Reaksi Sistem
1.User memilih fitur <b>ANN</b>		
		2.Menampilkan halaman prediksi ANN
3.Memilih nama sekolah yang akan diprediksi		
4. Mengklik button <b>Lakukan Prediksi</b>		
		5.Menampilkan nilai hasil prediksi

**Tabel 4. 4 Use Case Scenario Prediksi SVR**

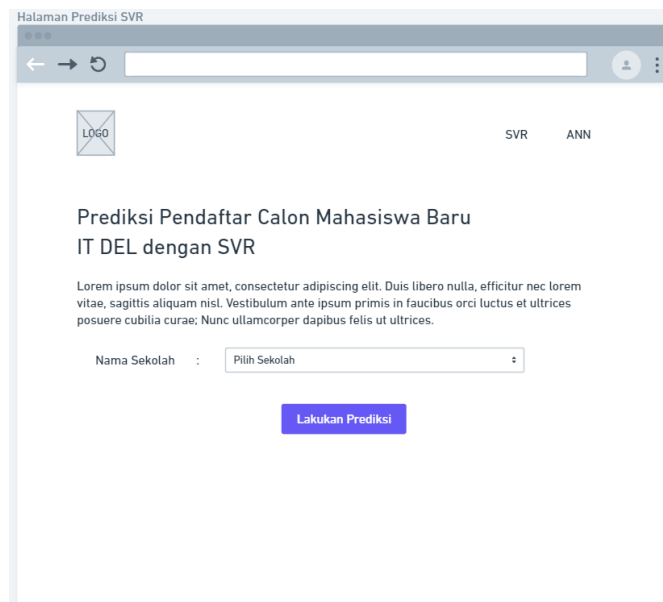
Use Case	:	Prediksi SVR
Deskripsi	:	Use case ini menggambarkan bahwa aplikasi dapat melakukan prediksi SVR
Aktor	:	User
Kondisi Awal	:	Sistem menampilkan halaman home Del Predict
Kondisi Akhir	:	Sistem menampilkan hasil prediksi
Skenario		
Aksi Aktor		Reaksi Sistem
1.User memilih fitur <b>SVR</b>		
		2.Menampilkan halaman prediksi SVR
3.Memilih nama sekolah yang akan diprediksi		
4. Mengklik button <b>Lakukan Prediksi</b>		
		5.Menampilkan nilai hasil prediksi

#### 4.4.3 Design Aplikasi

Berikut ini merupakan *design* tampilan dari aplikasi yang akan dijadikan sebagai simulator dari pemodelan algoritma.



**Gambar 4. 6 Halaman Home**



**Gambar 4. 7 Halaman Prediksi SVR**



Halaman Prediksi SVR

← → ↺

LOGO SVR ANN

### Prediksi Pendaftar Calon Mahasiswa Baru IT DEL dengan SVR

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis libero nulla, efficitur nec lorem vitae, sagittis aliquam nisl. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nunc ullamcorper dapibus felis ut ultrices.

Nama Sekolah : SMA BINTANG TIMUR BALIGE

Lakukan Prediksi

Hasil Prediksi Jumlah Pendaftar Calon Mahasiswa SMA BINTANG TIMUR BALIGE adalah 52 orang

**Gambar 4. 8 Halaman Hasil Prediksi SVR**

Halaman Prediksi ANN

← → ↺

LOGO SVR ANN

### Prediksi Pendaftar Calon Mahasiswa Baru IT DEL dengan ANN

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis libero nulla, efficitur nec lorem vitae, sagittis aliquam nisl. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nunc ullamcorper dapibus felis ut ultrices.

Nama Sekolah : Pilih Sekolah

Lakukan Prediksi

**Gambar 4. 9 Halaman Prediksi ANN**

The screenshot shows a web browser window with the title 'Halaman Prediksi ANN'. The browser's address bar is empty. The main content area has a heading 'Prediksi Pendaftar Calon Mahasiswa Baru IT DEL dengan ANN'. Below the heading is a paragraph of placeholder text: 'Lorem ipsum dolor sit amet, consectetur adipiscing elit. Duis libero nulla, efficitur nec lorem vitae, sagittis aliquam nisl. Vestibulum ante ipsum primis in faucibus orci luctus et ultrices posuere cubilia curae; Nunc ullamcorper dapibus felis ut ultrices.' Below this text is a form with a label 'Nama Sekolah :' and a dropdown menu showing 'SMA BINTANG TIMUR BALIGE'. A blue button labeled 'Lakukan Prediksi' is positioned below the form. At the bottom, a light blue box displays the prediction result: 'Hasil Prediksi Jumlah Pendaftar Calon Mahasiswa SMA BINTANG TIMUR BALIGE adalah 52 orang'.

**Gambar 4. 10 Halaman Hasil Prediksi ANN**

## BAB 5

### IMPLEMENTASI

Pada bab ini akan dijelaskan mengenai lingkungan implementasi, batasan, pengumpulan data, preprocessing data, pembentukan model untuk prediksi, dan pembuatan aplikasi web sebagai simulator prediksi.

#### 5.1 Lingkungan Implementasi

Sub-bab ini akan menjelaskan mengenai spesifikasi *hardware* (perangkat keras) dan *software* (perangkat lunak) yang digunakan dalam proses implementasi selama masa penelitian.

##### a. Hardware

Spesifikasi *hardware* yang digunakan antara lain :

**Tabel 5. 1 Spesifikasi Hardware**

No	Hardware	Spesifikasi
1	Laptop	Asus Laptop X507UF
2	<i>Processor</i>	Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz(8CPU)
3	RAM	8 GB

##### b. Software

Spesifikasi *software* yang digunakan antara lain :

**Tabel 5. 2 Spesifikasi Software**

No	Software	Spesifikasi
1	Sistem Operasi	Windows 10
2	Development Tools	Google Colaboratory, Jupyter Notebook
3	Bahasa Pemograman	Python

4	Web Browser	Google Chrome
5	Dokumentasi	Ms. Word, Google docs.

## 5.2 Batasan Implementasi

Batasan implementasi yang ditetapkan dalam implementasi Tugas Akhir ini adalah :

1. Bahasa pemrograman yang digunakan dalam melakukan implementasi arsitektur dan pembentukan model prediksi jumlah pendaftar adalah *python*.
2. Implementasi yang dilakukan oleh peneliti sampai pada tahap pembuatan aplikasi web untuk melakukan prediksi jumlah pendaftar calon mahasiswa IT Del yang akan mendaftar pada tahun depan.
3. Implementasi yang dilakukan berfokus pada pembangunan model dengan algoritma *Support Vector Regression* dan *Artificial Neural Network*. Dari kedua algoritma akan dilakukan penentuan algoritma dengan akurasi terbaik yang akan digunakan pada aplikasi web yang disiapkan.
4. Implementasi dilakukan mulai dari tahapan *encoding data*, dikarenakan tahapan *preprocessing* lain seperti data integrasi dan data cleaning sudah dilakukan secara manual dengan cara yang telah dijelaskan pada **subbab 3.3.1** dan **subbab 3.3.2**.

## 5.3 Implementasi Preprocessing

Tahapan implementasi *preprocessing* data yang dilakukan adalah *encoding* data dimana data yang digunakan merupakan data yang telah melalui tahapan integrasi data dan *cleaning* data.

### 5.3.1 Pengecekan Outlier

Sebelum dilakukan tahapan *pre-processing* data untuk mencegah adanya terdeteksi nilai yang tidak wajar yang dapat mengakibatkan kesalahan pada analisis data dan prediksi. Outlier terjadi ketika nilai berada di luar range nilai pada umumnya. Berdasarkan implementasi yang telah dilakukan untuk pengecekan outlier, diperoleh hasil sebagai berikut:

Total outliers pada tahun 2016: 168
-------------------------------------

Total outliers pada tahun 2017: 172
Total outliers pada tahun 2018: 163
Total outliers pada tahun 2019: 169
Total outliers pada tahun 2020: 171
Total outliers pada tahun 2021: 170
Total outliers pada tahun 2022: 166

Berdasarkan hasil outlier yang diterima dari hasil pengecekan, bahwa sebagian besar variabel mengalami outlier. Hal ini disebabkan karena nilai pendaftar pada setiap sekolah berbeda-beda dan tidak adanya kemungkinan jumlah pendaftar antara sekolah A, sekolah B, dan sekolah C berada pada *range* nilai yang sama. Oleh karena sifat data jumlah pendaftar antara sekolah yang satu dengan yang lain dapat berada pada rentang yang jauh, maka peneliti tidak melakukan tindakan untuk mengatasi *outlier* yang terjadi.

### 5.3.2 Encoding

Sesuai dengan analisis data yang telah dilakukan pada **subbab 3.3.3** bahwa pada dataset yang akan digunakan dalam pemodelan, terdapat variabel "nama sekolah" yang perlu diubah ke dalam bentuk numerik. Hal ini digunakan agar nilai "nama sekolah" dapat dipelajari, dikarenakan *machine learning* hanya dapat mengolah data dengan bentuk numerik. Berikut ini adalah bentuk implementasi encoding data yang dilakukan.

Input	: Dataset file .csv yang telah dilakukan data integrasi dan data cleaning.
Output	: Dataset baru .csv yang telah dilakukan <i>encoding</i> data.
Library	: Labelencoder yang digunakan untuk melakukan <i>encoding</i>
Tahapan	: 1.Lakukan import dataset dan library yang akan digunakan. 2.Memisahkan variabel yang akan di encoding dengan variabel lain. 3.Lakukan <i>encoding</i> pada variabel yang telah dipisahkan

## 5.4 Implementasi Algoritma Support Vector Regression

### 5.4.1 Pencarian nilai parameter terbaik

Pada tahapan implementasi algoritma *support vector regression* dilakukan dengan menggunakan data yang sudah mengalami proses Encoding pada tahapan sebelumnya. Sebelum dilakukan pembentukan model, peneliti melakukan pencarian *parameter* yang akan digunakan dalam pembentukan model. Pencarian parameter dilakukan dengan menggunakan teknik *gridsearch*. Sesuai dengan penjelasan pada **subbab 2.8**, penggunaan teknik GridSearch akan menghasilkan parameter terbaik dari nilai-nilai yang telah diinisialisasi. Untuk nilai dari kernel dan parameter-parameter diinisialisasi sebagaimana sudah dijelaskan pada **subbab 4.4.1** dengan nilai-nilai berikut.

**Tabel 5. 3 Nilai parameter algoritma support vector regression**

Parameter	Value
Kernel	Linear, RBF, Sigmoid, Poly
C	0.1, 1, 10, 100
Epsilon	0.01, 0.1, 1
Gamma	0.001, 0.01, 0.1, 1

Dengan menggunakan GridSearch nilai-nilai dari setiap kernel dan parameter yang telah diinisialisasikan akan didapatkan nilai kernel dan parameter terbaik. *Grid Search* akan mencari kombinasi parameter satu per satu dan membandingkan nilai galat terkecil pada parameter dan kernel tersebut (Saputra et al., 2019). Berikut ini adalah tahapan implementasi *gridsearch* yang dilakukan.

Input	: Data yang telah melalui pre-processing
Output	: Nilai Kernel dan parameter terbaik menggunakan <i>Grid Search</i>
Library	: <ul style="list-style-type: none"> <li>- Pandas digunakan untuk memuat dataset dan membuat dataframe.</li> <li>- KFold digunakan untuk mengatur pembagian data</li> <li>- GridSearchCV digunakan untuk menemukan parameter dan kernel terbaik</li> </ul>

Tahapan	: 1.Melakukan import dataset dan library yang akan digunakan 2. Menentukan variabel input dan output 3. Melakukan pre-processing dataset 4. Melakukan inisialisasi nilai parameter dan kernel 5. Melakukan pemodelan menggunakan algoritma SVR 6. Melakukan fungsi objek pada <i>Grid Search</i> 7. Menampilkan output yaitu kernel dan parameter terbaik pada algoritma SVR berdasarkan nilai yang telah diinisialisasi.
---------	---

#### 5.4.2 Percobaan variasi nilai k-fold Cross Validation

Setelah mendapatkan nilai kernel dan parameter terbaik menggunakan *Grid Search* proses selanjutnya yang dilakukan yaitu melakukan penyetelan algoritma SVR menggunakan parameter dan kernel terbaik yang telah didapatkan dengan variasi nilai k-fold cross validation yang digunakan. Kemudian melakukan pengecekan nilai RMSE berdasarkan nilai dari k-fold cross validation. Untuk rentang yang diberikan yaitu mulai dari nilai  $k = 2$  sampai dengan nilai  $k = 10$ .

Adapun langkah-langkah yang dilakukan adalah sebagai berikut.

Input	: Kernel dan parameter terbaik dan nilai $k=2$ sampai 10
Output	: Evaluasi model menggunakan RMSE
Library	: <ul style="list-style-type: none"> <li>- Pandas digunakan untuk membuat dataset dan membuat dataframe</li> <li>- Stats digunakan untuk membantu menghitung statistik deskriptif.</li> <li>- Numpy digunakan untuk melakukan komputasi ilmiah atau scientific computing</li> <li>- Label Encoder digunakan untuk melakukan encoding terhadap dataset</li> </ul>

	<ul style="list-style-type: none"> <li>- K-Fold digunakan untuk melakukan pembagian data ke dalam bentuk data latih dan data uji</li> <li>- SVR digunakan untuk implementasi algoritma support vector regression.</li> <li>- mean_squared_error digunakan untuk menemukan evaluasi model menggunakan RMSE</li> </ul>
Tahapan	: 1. Melakukan import dataset dan library yang akan digunakan. 2. Menentukan variabel input dan output 3. Melakukan pre-processing dataset 4. Melakukan inisialisasi nilai kernel dan parameter dan dicari nilai terbaik dari masing – masing kernel dan parameter tersebut. 5. Melakukan inisialisasi nilai k untuk penggunaan K-Fold. 6. Melakukan pemodelan menggunakan algoritma SVR menggunakan nilai kernel dan parameter terbaik 7. Melakukan evaluasi model yang telah dibangun menggunakan RMSE 8. Melakukan prediksi jumlah pendaftar

Berdasarkan nilai RMSE yang didapatkan pada setiap nilai k, nilai RMSE yang paling kecil yang akan digunakan untuk melakukan pemodelan algoritma dan digunakan untuk melakukan prediksi jumlah pendaftar

### 5.5 Implementasi Algoritma Artificial Neural Network

Pada implementasi algoritma *Artificial Neural Network* data yang digunakan adalah data yang telah menyelesaikan tahapan pre-processing. Sebelum dilakukan implementasi algoritma, peneliti melakukan pencarian *hyperparameter* sama seperti yang dilakukan algoritma *support vector regression* yang dijelaskan pada **subbab 5.4** yaitu dengan menggunakan teknik *gridsearch*. Agar *gridsearch* dapat mempelajari nilai *hyperparameter* pada model yang akan digunakan, maka perlu



dilakukan pembentukan model terlebih dahulu. Pembentukan model harus disesuaikan dengan bentuk model yang akan dilakukan pada saat training data. Setelah model dibentuk maka dilakukan inisialisasi nilai *hyperparameter* seperti pada **Tabel 4.1**.

Setelah dilakukan inisialisasi *hyperparameter* selanjutnya akan dideklarasikan fungsi *gridsearch* untuk menjalankan setiap nilai *hyperparameter* ke dalam model dan untuk menghasilkan *hyperparameter* terbaik. Implementasi teknik ini digambarkan pada langkah berikut :

Input	: Data yang telah melalui pre-processing
Output	: Nilai <i>hyperparameter</i> terbaik yang dihasilkan oleh teknik <i>gridsearch</i> .
Library	: <ul style="list-style-type: none"> <li>- Pandas digunakan untuk memuat dataset dan membuat dataframe.</li> <li>- Tensorflow digunakan untuk membantu dalam proses pelatihan <i>neural network</i>.</li> <li>- GridSearch digunakan untuk mencari nilai parameter terbaik yang menghasilkan performa model yang optimal.</li> <li>- Keras digunakan untuk membuat bentuk pemodean <i>deep learning</i>.</li> </ul>
Tahapan	: <ol style="list-style-type: none"> <li>1. Lakukan import dataset dan library yang akan digunakan.</li> <li>2. Menentukan variabel input dan output.</li> <li>3. Melakukan pre-processing dataset</li> <li>4. Buat struktur model ANN yang digunakan.</li> <li>5. Inisialisasi nilai-nilai <i>hyperparameter</i> yang akan digunakan.</li> <li>6. Inisialisasi objek Gridsearch dengan memasukkan <i>hyparameter</i> yang akan dijalankan.</li> <li>7. Menjalankan fungsi objek gridsearch</li> <li>8. Menampilkan output <i>best_params</i> sebagai nilai</li> </ol>

*hyperparameter* yang lebih optimal.

Nilai-nilai *hyperparameter* yang telah diterima sebagai nilai yang paling optimal, maka selanjutnya nilai-nilai tersebut akan digunakan di dalam model algoritma *Artificial Neural Network* yang akan dilatih. Berikut adalah implementasi dari pemodelan *Artificial Neural Network*.

Input	: Data yang telah melalui proses pre-processing
Output	: Nilai hasil evaluasi model seperti RMSE dan Akurasi
Library	: <ul style="list-style-type: none"> <li>- Pandas digunakan untuk memuat dataset dan membuat dataframe.</li> <li>- Tensorflow digunakan untuk membantu dalam proses pelatihan <i>neural network</i>.</li> <li>- K-Fold digunakan untuk melakukan pembagian data ke dalam bentuk data latih dan data uji.</li> <li>- Keras digunakan untuk membuat bentuk pemodean <i>deep learning</i>.</li> <li>- Numpy digunakan untuk melakukan komputasi ilmiah atau <i>scientific computing</i>.</li> <li>- Matplotlib digunakan untuk visualisasi data.</li> </ul>
Tahapan	: <ol style="list-style-type: none"> <li>1. Lakukan import dataset dan library yang akan digunakan.</li> <li>2. Menentukan variabel input dan output.</li> <li>3. Melakukan pre-processing dataset</li> <li>4. Menginisialisasikan nilai <i>hyperparameter</i> yang akan dicari nilai terbaiknya.</li> <li>5. Menginisialisasikan nilai <i>k</i> untuk penggunaan <i>K-Fold</i>.</li> <li>6. Melakukan train data sebanyak nilai iterasi (<i>epoch</i>) kali dengan membagi variabel X dan y ke dalam <i>X_train</i>,</li> </ol>

$X_{test}$ ,  $y_{train}$ ,  $y_{test}$  sebanyak  $k$  bagian. Pada saat train juga dilakukan pembentukan model ANN dan pemanggilan *hyperparameter* yang akan digunakan. Setelah dilakukan pembentukan model, selanjutnya model yang dijalankan, kemudian model kita latih, dan kita lakukan evaluasi model.

7. Menampilkan RMSE data latih dan data uji
8. Menampilkan nilai RMSE dari hasil pembentukan model
9. Menampilkan visualisasi dari nilai RMSE model.

Pada pembentukan model algoritma, peneliti membentuk model dengan 1 input layer, 2 *hidden layer*, dan 1 output layer sehingga banyak layer yang dibentuk adalah 4 layer. Setiap layer memiliki jumlah neuron yang berbeda-beda seperti yang dijelaskan pada **subbab 4.4.2**. Nilai bias dan bobot pada setiap layer akan diinisialisasikan secara otomatis dengan menggunakan fungsi *kernel\_initializer=uniform* seperti yang dapat dilihat pada **Source Code Pemodelan Algoritma SVR**. Untuk nilai aktivasi yang digunakan pada layer juga berbeda dan disesuaikan dengan fungsi aktivasi yang lebih optimal sesuai dengan hasil pada *gridsearch*. Setelah dilakukan pembentukan model, dilakukan konfigurasi proses pelatihan model. Pelatihan model ini melibatkan *loss function*, optimasi, dan metric evaluasi yang digunakan. *Loss function* ini digunakan untuk melihat perbedaan antara nilai prediksi dan nilai sebenarnya. Optimasi digunakan untuk mengatur mengoptimalkan parameter model selama pelatihan. Pengoptimalan yang dimaksud adalah perbaikan nilai bobot saat melakukan pelatihan dengan menggunakan nilai laju pembelajaran yang digunakan. Metric evaluasi digunakan untuk memantau kinerja model selama pelatihan model. Setelah model dikonfigurasi, model siap untuk dilatih dengan menggunakan perintah *model.fit()*. Dalam proses ini, model akan dilatih sesuai dengan nilai *epoch* yang telah diinisialisasikan. Proses pelatihan akan dilakukan selama nilai iterasi *epoch* belum terpenuhi. Tahapan terakhir dari proses pelatihan adalah

evaluasi model. Pada tahapan ini model yang telah dilatih akan dilakukan evaluasi, dimana dari hasil evaluasi akan menghasilkan nilai RMSE ketika proses pelatihan, RMSE ketika proses validasi, dan nilai RMSE dari metric evaluasi model yang dilakukan. Pada pelaksanaan implementasi algoritma, peneliti melakukan percobaan untuk mencari nilai  $k$  dan nilai *epoch* terbaik. Dalam percobaan ini nilai-nilai *epoch* akan digunakan pada setiap nilai  $k$ , sehingga dengan itu peneliti dapat melihat bagaimana nilai RMSE pada setiap *epoch* dengan pembagian  $k$  data. Selain melihat nilai RMSE, peneliti juga melihat hasil pelatihan apakah hasil pelatihan mengalami *overfitting*, *underfitting*, atau *good fit*. Pada hasil yang diperoleh hanya akan menerima hasil pelatihan yang *good fit* dan memiliki nilai RMSE terendah. Dalam implementasi algoritma ANN ada beberapa percobaan yang dilakukan. Percobaan ini akan dijelaskan pada subbab berikut.

### 5.5.1 Pencarian Nilai Epoch

Untuk melakukan pencarian nilai epoch yang akan digunakan dalam pemodelan, peneliti menggunakan teknik *early stopping*. Percobaan ini dilakukan peneliti untuk melihat nilai epoch yang optimal digunakan pada model yang telah dirancang. Pada percobaan ini pertama kali dilakukan inisialisasi nilai epoch yang dipilih secara acak. Nilai epoch yang digunakan dalam percobaan ini adalah 50, 100, 250, 500, dan 1000. Pada bagian implementasi, teknik *early stopping* digunakan dengan penambahan fungsi *early\_stop* yang diambil dari library *Keras*. Seperti yang sudah dijelaskan pada **subbab 2.12**, bahwa proses pelatihan akan berhenti ketika sudah mulai akan terjadi *overfitting*, sehingga dari hasil percobaan ini akan dihasilkan nilai-nilai epoch yang optimal pada nilai  $k$  dan nilai *epoch* tertentu. Berdasarkan data hasil percobaan pada lampiran **Hasil Implementasi Early Stopping** yang merupakan hasil implementasi yang dilakukan, nilai *epoch* optimal berada di antara nilai 25 sampai dengan 100. Berlandaskan dari *range* nilai *epoch* tersebut maka peneliti melakukan percobaan kedua dengan membuat epoch baru dalam dalam rentang nilai tersebut dan nilai *epoch* yang ditentukan memiliki nilai range yang sama. Penentuan nilai *epoch* dengan range yang sama

memiliki tujuan untuk melihat perbedaan hasil evaluasi model dengan jarak perbedaan nilai *epoch* yang sama.

### 5.5.2 Percobaan Variasi Nilai Epoch

Seperti yang telah dijelaskan pada **subbab 5.5.1** di atas, bahwa akan dilakukan percobaan kedua untuk melihat nilai *epoch* terbaik dari antara nilai variasi nilai *epoch* lain yang diambil dari rentang nilai 15 sampai 100. Untuk menjaga adanya kemungkinan nilai RMSE terbaik di luar epoch hasil *early stopping*, nilai epoch ditambah hingga rentang 120 sehingga nilai *epoch* yang diuji adalah 15, 30, 45, 60, 75, 90, 105, 120. Dengan nilai dari nilai-nilai *epoch* tersebut maka akan dicari nilai *epoch* yang menghasilkan nilai RMSE yang terbaik, dan nilai *epoch* yang menghasilkan RMSE terbaik akan digunakan sebagai nilai *epoch* pada model yang akan digunakan di dalam simulator. Variasi nilai *epoch* ini akan dikombinasi dengan nilai *k*.

### 5.5 Uji Validasi Algoritma

Setelah melakukan percobaan pemodelan dengan algoritma *Support Vector Regression* dan *Artificial Neural Network*, dilakukan uji validasi hasil pembentukan model. Pada uji validasi, dataset yang dilakukan *training* adalah dataset dengan jumlah pendaftar dari tahun 2016-2021, kemudian data jumlah pendaftar tahun 2022 dipisahkan dengan maksud memvalidasi hasil pembentukan model dalam melakukan prediksi jumlah pendaftar tahun 2022. Model yang digunakan untuk melakukan prediksi adalah model dengan nilai performansi terbaik dari kedua algoritma.

## BAB 6

### HASIL DAN PEMBAHASAN

Pada bab ini akan dilakukan penjelasan mengenai hasil implementasi yang telah dilakukan.

#### 6.1. Hasil dan Pembahasan Preprocessing Dataset

Pada sub-bab ini akan menjelaskan hasil pre-processing yang didapatkan. Adapun tahapan pre-processing yang dilakukan yaitu dengan menggunakan Label Encoder dan mendapatkan hasil sebagai berikut:

**Tabel 6. 1 Dataset yang sudah dilakukan pre-proceessing**

Nama Sekolah	nama_sekolah _encoded	2016	2017	2018	2019	2020	2021	2022
SMA ABDI SEJATI PERDAGANGAN	0	1	2	0	1	0	0	2
SMA ADVENT PEMATANG SIANTAR	1	1	0	1	1	1	0	0
SMA ASISI PEMATANG SIANTAR	2	1	0	0	3	5	1	3
SMA BINTANG TIMUR BALIGE	3	70	54	53	110	46	48	53
SMA BINTANG TIMUR PEMATANG SIANTAR	4	30	19	6	7	18	8	2
SMA BRIGJEN KATAMSO 1 MEDAN	5	1	1	4	1	0	0	0
SMA BUDI MULIA PEMATANG SIANTAR	6	44	47	39	36	16	30	10
SMA BUDI MURNI 1 MEDAN	7	8	26	12	2	5	11	3
SMA BUDI MURNI 2 MEDAN	8	2	2	1	3	3	3	1
SMA CAHAYA MEDAN	9	14	11	6	1	2	1	0
SMA CINTA RAKYAT PEMATANG SIANTAR	10	5	3	0	1	2	1	1
SMA FRANSISKUS SIBOLGA	11	2	1	9	1	1	0	0
SMA GLOBAL PERSADA MANDIRI	12	1	0	0	1	0	0	0
SMA KALAM KUDUS MEDAN	13	1	3	11	1	1	6	0
SMA KAMPUS FKIP HKBP NOMENSEN	14	1	1	1	0	0	0	0
SMA KARTIKA JAYA 1-4	15	1	1	0	0	0	0	0
SMA KATOLIK KABANJAHE	16	1	0	0	3	2	4	2
SMA KATOLIK SIBOLGA	17	10	10	2	4	17	4	4
SMA KATOLIK TRISAKTI MEDAN	18	9	7	0	3	4	4	0
SMA KRISTEN IMMANUEL	19	1	1	2	3	0	0	0
SMA KRISTEN KALAM KUDUS PEMATANG SIANTAR	20	7	10	4	3	4	4	0
SMA MANDIRI TANGERANG	21	1	0	0	1	2	1	1
SMA MARKUS MEDAN	22	3	2	1	4	2	6	3
SMA MARS PEMATANG SIANTAR	23	3	0	0	0	0	0	0
SMA METHODIST 1 MEDAN	24	3	11	6	1	0	0	0
SMA METHODIST 2 MEDAN	25	6	11	12	2	0	0	0
SMA METHODIST 8 MEDAN	26	1	8	5	1	3	2	0
SMA METHODIST-AN PANCUR BATU	27	4	0	2	2	4	5	2
SMA NASRANI 1 MEDAN	28	1	0	1	1	3	2	0

Berdasarkan hasil pre-processing dataset yang didapatkan yang merujuk pada **Tabel 6.1**, variabel "nama sekolah" yang sebelumnya bertipe kategorikal tidak dapat dimasukkan kedalam pemodelan, kemudian melalui hal itu diperlukan tindakan konversi untuk mengubah tipe dari variabel "nama sekolah" agar berbentuk numerik. Teknik yang dilakukan yaitu Encoding. Adapun encoding yang digunakan yaitu Label Encoder yang mana didapatkan kolom atau variabel

baru dengan nama "nama\_sekolah\_encoded". Dan kemudian hasilnya digunakan untuk melakukan pemodelan.

## 6.2. Hasil dan Pembahasan Algoritma Support Vector Regression

Setelah membangun model prediksi jumlah pendaftar dengan menggunakan algoritma Support Vector Regression, maka didapatkan hasil sebagai berikut:

**Tabel 6. 2 Kernel dan Parameter Terbaik**

kernel	gamma	cost	Epsilon
Linear	0,001	100	0,01

Setelah proses *pre-processing* dataset dilakukan seperti pada Gambar 6.1, kemudian dilakukan pemodelan menggunakan dataset yang sudah siap pakai, dan hal pertama yang dilakukan yaitu menemukan kernel dan parameter terbaik berdasarkan inisialisasi kernel dan parameter yang diberikan pada **subbab 4.4.1**. Setelah melakukan pemilihan kernel dan parameter terbaik menggunakan grid search, grid search akan bekerja dengan cara melakukan kombinasi terhadap nilai kernel dan parameter yang telah diinisialisasikan untuk mendapatkan nilai error yang paling rendah. Melalui cara kerja tersebut, didapatkan hasil bahwa kernel terbaik yaitu kernel linear, hasil ini menunjukkan bahwa dataset yang digunakan dalam penelitian lebih mudah dipisahkan secara linier dengan menggunakan hyperplane. Kemudian setelah mendapatkan nilai dari kernel yang terbaik, dilanjutkan dengan mencari nilai terbaik dari setiap parameter yaitu parameter gamma dengan nilai 0,001, parameter cost dengan nilai 100, parameter epsilon dengan nilai 0,01. Untuk parameter gamma menghasilkan nilai 0,001 menunjukkan bahwa radius kesamaan yang besar yang menghasilkan lebih banyak titik yang dikelompokkan bersama, namun dikarenakan kernel terbaik yang didapatkan adalah kernel liner, maka untuk parameter gamma tidak terlalu berpengaruh terhadap nilai error dan juga hasil prediksi. Kemudian parameter cost sendiri menghasilkan nilai 100 menunjukkan bahwa model memiliki deviasi yang kecil dari parameter epsilon yang masih dapat ditoleransi. Dan untuk parameter epsilon menghasilkan nilai terbaik pada nilai 0,01 mengindikasikan bahwa

toleransi kesalahan yang lebih rendah ataupun ketat memberikan hasil yang lebih baik dalam hal performa berdasarkan metrik evaluasi yang digunakan.

Setelah mendapatkan nilai terbaik dari setiap kernel dan parameter, setiap nilai tersebut digunakan dalam pemodelan dan dilakukan evaluasi model dengan menggunakan evaluasi model Root Mean Squared Error, dengan menggunakan kombinasi pada rentang nilai k pada k-fold cross validation mulai dari rentang nilai 2 sampai dengan 10, setelah dilakukan analisis, didapatkan kesimpulan bahwa nilai k pada k-fold cross validation berpengaruh pada nilai error model, kemudian didapatkan nilai RMSE terendah pada nilai k=7, dengan nilai 7.2893. Adapun hasil yang didapatkan adalah sebagai berikut:

**Tabel 6. 3 Evaluasi RMSE menggunakan nilai k**

<b>k</b>	<b>kernel</b>	<b>gamma</b>	<b>Cost</b>	<b>Epsilon</b>	<b>RMSE</b>
2	Linear	0,001	100	0,01	8.87419
3	Linear	0,001	100	0,01	8.51308
4	Linear	0,001	100	0,01	8.15710
5	Linear	0,001	100	0,01	7.82429
6	Linear	0,001	100	0,01	7.81281
7	Linear	0,001	100	0,01	7.28936
8	Linear	0,001	100	0,01	7.37291
9	Linear	0,001	100	0,01	7.65842
10	Linear	0,001	100	0,01	7.33568

### **6.3. Hasil dan Pembahasan Algoritma Artificial Neural Network**

Pada subbab ini akan membahas tentang hasil yang diperoleh setelah melakukan percobaan. Setelah melakukan percobaan mulai dari tahapan pencarian nilai



*hyperparameter* dengan menggunakan *gridsearch* hingga melakukan pemodelan algoritma. Berdasarkan percobaan yang dilakukan untuk mencari nilai *hyperparameter* yang optimal digunakan, diperoleh nilai-nilai *hyperparameter* seperti berikut :

1. Jumlah neuron pada *hidden layer* yang digunakan adalah 12 neuron.
2. Fungsi aktivasi yang digunakan pada *hidden layer* adalah ReLU.
3. Nilai *learning rate* yang digunakan adalah 0.01.

Nilai-nilai *hyperparameter* yang telah diperoleh tidak akan mempengaruhi nilai-nilai data yang diproses. Nilai-nilai ini hanya berperan dalam mengatur bagaimana model belajar agar dapat bekerja secara optimal. Penggunaan nilai *hyperparameter* pada model memiliki dampak yang cukup berpengaruh pada kinerja model yang sedang dikembangkan. Untuk menilai kinerja model digunakan metrik RMSE untuk melihat ukuran kinerja model. Pemilihan metrik RMSE digunakan sebagai tolak ukur kinerja model dikarenakan tujuan model yang dibuat adalah melakukan prediksi nilai-nilai di masa depan (*continues target*). Metrik ini hanya akan mengukur besar kesalahan prediksi yang mungkin terjadi pada model. Semakin kecil nilai RMSE yang diperoleh maka akan semakin kecil kemungkinan kesalahan yang dilakukan oleh model, sehingga dapat dikatakan bahwa model tersebut akan semakin baik.

Setelah memperoleh nilai *hyperparameter* seperti di atas, nilai-nilai tersebut selanjutnya digunakan ke dalam model algoritma untuk dilakukan pembentukan model. Seperti yang telah disampaikan pada **subbab 5.5** bahwa pada saat melakukan pemodelan dilakukan percobaan juga terkait nilai *k* dan nilai *epoch* yang menghasilkan nilai RMSE terbaik. Dari hasil percobaan yang telah dilakukan, diperoleh nilai *k* dan nilai *epoch* terbaik adalah  $k = 4$  dan  $epoch = 105$  dengan nilai RMSE yang dihasilkan adalah 0.04929599755078. Berdasarkan hasil percobaan yang dilakukan bahwa kombinasi nilai *k* dan nilai *epoch* juga dapat mempengaruhi kinerja model yang dibangun. Oleh karena itu, selain nilai *learning rate*, fungsi aktivasi, dan jumlah neuron, pemilihan nilai *k* dan nilai *epoch* juga perlu diperhatikan untuk menghasilkan kinerja model yang lebih baik.

#### 6.4. Pembahasan Evaluasi Model

Berdasarkan hasil percobaan dan evaluasi model yang telah diperoleh dari kedua algoritma maka didapatkan bahwa bentuk pemodelan terbaik dihasilkan pada algoritma *artificial neural network* dengan nilai RMSE yang diperoleh adalah 0.04929599755078. Berdasarkan hasil evaluasi model ini, terbukti hipotesis **H0** yang disampaikan pada **Subbab 3.7** bahwa kinerja model ANN lebih baik dibandingkan dengan model SVR dalam melakukan prediksi. Hasil ini disebabkan karena algoritma ANN dapat melakukan adaptasi dan generalisasi dengan baik terhadap data pelatihan. Dalam proses pelatihan, algoritma ANN akan menyesuaikan nilai bobot dan bias secara iteratif untuk mencapai performa yang optimal pada data pelatihan.

#### 6.5. Hasil dan Pembahasan Uji Validasi

Berdasarkan hasil percobaan uji validasi yang dilakukan pada 9 sekolah didapatkan hasil sebagai berikut:

**Tabel 6. 4 Hasil Uji Validasi dengan model SVR**

<b>Nama Sekolah</b>	<b>Aktual</b>	<b>Prediksi</b>
SMA Abdi Sejati Perdagangan	2	1
SMA Advent Pematang Siantar	0	1
SMA Asisi Pematang Siantar	3	1
SMA Bintang Timur Balige	53	11
SMA Bintang Timur Pematang Siantar	2	5
SMA Brigjen Katamso 1 Medan	0	1
SMA Budi Mulia Pematang Siantar	10	11
SMA Budi Murni 1 Medan	3	4
SMA Budi Murni 2 Medan	1	1

**Tabel 6. 5 Hasil Uji Validasi dengan Model ANN**

<b>Nama Sekolah</b>	<b>Target</b>	<b>Prediksi</b>
SMA Abdi Sejati Perdagangan	2	0
SMA Advent Pematang Siantar	0	0
SMA Asisi Pematang Siantar	3	56
SMA Bintang Timur Balige	53	2658
SMA Bintang Timur Pematang Siantar	2	574
SMA Brigjen Katamso 1 Medan	0	0
SMA Budi Mulia Pematang Siantar	10	1793
SMA Budi Murni 1 Medan	3	586
SMA Budi Murni 2 Medan	1	186

Berdasarkan hasil uji validasi yang dilakukan pada kedua model, bahwa kedua model masih belum dapat digunakan untuk memprediksi jumlah pendaftar calon mahasiswa baru, dikarenakan selisih kesalahan hasil prediksi dengan nilai target masih sangat besar. Seperti pada prediksi jumlah pendaftar calon mahasiswa baru asal SMA Bintang Timur Balige yang mana hasil prediksi jumlah pendaftar calon mahasiswa baru adalah 2658 pendaftar sedangkan nilai targetnya adalah 53 pendaftar. Kesalahan ini muncul dikarenakan jumlah data yang dipelajari oleh model masih sedikit yaitu hanya mempelajari data jumlah pendaftar dari tahun 2016 sampai 2021 untuk prediksi 2022. Padahal seharusnya, model dapat memprediksi hasil dengan baik dengan mempelajari banyak data pada dataset.

## **6.6. Hasil Implementasi Simulator**

Hasil dari pembangunan aplikasi yang telah dibangun sesuai dengan design prototype yang telah dibuat, aplikasi sudah dapat berjalan dengan baik dan sudah dapat menghasilkan hasil prediksi.

## BAB 7

### KESIMPULAN DAN SARAN

#### 7.1. Kesimpulan

Berdasarkan hasil penelitian dan implementasi dapat diberi kesimpulan sebagai berikut:

1. Untuk melakukan pengolahan data mentah ke dalam bentuk dataset yang akan diolah ke dalam model, dimulai dari bentuk dataset yang ingin diolah baik dalam bentuk dataset time series ataupun dataset tabular. Selain itu juga perlu diperhatikan tahapan *pre-processing* yang perlu dilakukan untuk menyiapkan dataset.
2. Untuk melakukan pemodelan algoritma *Artificial Neural Network* dan *Support Vector Regression* yang optimal, perlu dilakukan pencarian nilai *hyperparameter* yang dapat membantu kinerja model agar bekerja dengan baik. Dengan menghasilkan bentuk pemodelan yang optimal, maka kemungkinan kesalahan prediksi akan semakin kecil.
3. Berdasarkan hasil percobaan yang telah dilakukan pada kedua model algoritma yaitu algoritma *Artificial Neural Network* dan *Support Vector Regression*, model yang memiliki nilai RMSE terbaik dengan nilai terendah model terbaik ditunjukkan pada model algoritma *Artificial Neural Network* dengan nilai  $RMSE = 0.04929599755078$ .
4. Berdasarkan dari hasil uji validasi model yang telah dibuat, bahwa kedua model belum dapat melakukan prediksi dengan baik untuk tahun selanjutnya. Hal ini diperkirakan karena banyak data yang dipelajari masih belum cukup banyak untuk dipelajari agar dapat melakukan prediksi dengan baik untuk tahun selanjutnya.

## **7.2. Saran**

Berdasarkan hasil penelitian yang telah dilakukan ada beberapa saran yang perlu dilakukan pada penelitian selanjutnya:

1. Pada penelitian selanjutnya dapat dilakukan pemodelan pada dataset yang memiliki jumlah data yang lebih banyak untuk membantu proses pembelajaran pola nilai untuk melakukan prediksi.

## LAMPIRAN

### Lampiran A Source Code

#### Source Code 1. Encoding Data

```
#Melakukan Import Library
library pandas
library numpy
library LabelEncoder
library matplotlib.pyplot

data <- data_breakdown.xlsx

#Membuat fungsi encoding
encoder <- LabelEncoder()

#Melakukan encoding pada variabel 'nama sekolah' dan menyimpan
hasil encoding ke dalam variabel 'nama_sekolah_encoded'
data['nama_sekolah_encoded'] <- encoder.fit
transform(data['Nama Sekolah'])

#Mengurutkan kolom agar lebih mudah menentukan variabel
feature dan variabel target
coloum_order <- ['Nama Sekolah', 'nama_sekolah_encoded',
2016, 2017, 2018, 2019, 2020, 2021, 2022]

#import dataset hasil encoding
data_prep.to_csv('data_prep.csv')
```

#### Source Code 2. *Grid Search* ANN

```
#Import Library yang akan digunakan dalam GridSearch
library pandas
library tensorflow
library Dense
library Sequential
library GridSearch
library mean_squared_error
library KerasClassifier
```

```

#Penentuan Variabel Input dan variabel Output
data <- data.prep.csv
X <- data.iloc[:,1:].values
y <- data.iloc[:, -1].values

#Pembentukan model ANN
create_model(num_neuron_hidden_layer, activation,
             learning_rate){
  model <- Sequential()
  add(Dense(Z.shape[1], kernel_initializer <- "uniform",
            input_shape=X.shape[1]))
  add(Dense(num_neurons_hidden_layer, kernel_initializer <-
            "uniform", activation))
  add(Dense(1)) #jumlah keluaran / output 1

  return model
}

#Inisialisasi beberapa nilai parameter
param_grid <- {
  num_neuron_hidden_layer : 12,16,24,32,64,128
  activation : relu, tanh, sigmoid
  learning_rate : 0.01, 0.001
}

#Membungkus model ANN kedalam variabel model
model <- KerasClassifier(build_func = create_model)

#Menjalankan bentuk model dengan grid_search
grid_search <- GridSearch(model, param_grid)
grid_search.fit(X, y)

#Mengeluarkan nilai parameter terbaik
grid_search.best_params_

```

### Source Code 3. *Grid Search* SVR

```

from google.colab import drive
# Mengimport modul GD pada Gcolab ke /content/drive
drive.mount('/content/drive')
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split,
GridSearchCV, KFold
from sklearn.metrics import mean_squared_error,
mean_absolute_error,

```

```

# Melakukan Load data
data = pd.read_csv("/content/drive/My
Drive/Tugas_Akhir/data-prep.csv")
# Penentuan Variabel Input(X) dan Variabel Output(y)
X <- data.iloc[:, 1:].values
y <- data.iloc[:, -1].values.reshape(-1, 1)

# Inisialisasi Parameter
parameters = {
kernel <- linear, rbf, sigmoid, polynomial
cost <- 0.1, 1, 10, 100
epsilon <- 0.01, 0.1, 1
gamma <- 0.001, 0.01, 1
}
# Melakukan pemodelan menggunakan SVR
model <- SVR()
cv = KFold(n_splits= 2 to 10 , shuffle=True,
random_state=42)
grid_search = GridSearchCV(model, parameters, cv=cv,
scoring="neg_root_mean_squared_error")
grid_search.fit(X, y)
Parameter terbaik <- grid_search.best_params_

```

#### Source Code 4. Pemodelan Algoritma ANN

```

#Import Library yang akan digunakan dalam pembentukan model
ANN
library pandas
library numpy
library tensorflow
library Dense
library Sequential
library GridSearch
library mean_squared_error
library KerasRegressor
library matplotlib.pyplot
library model_from_json
library KFold

```



```

#Penentuan Variabel Feature dan variabel Target
data <- data.prep.csv
X <- data.iloc[:,1:].values
y <- data.iloc[:, -1].values

#Checking outliers
data_check=data.drop(columns='Nama Sekolah')
mean = np.mean(data_check)
std_dev = np.std(data_check)
Mean <- mean
Standar deviasi <- std_dev
# count z-score
z_score <- (data_check - mean) / std_dev
print('Nilai z-score : ', z_score)
# identify outliers depends z-score value
outliers <- data_check[(z_score < batas_bawah) | (z_score >
batas_atas)]
Jumlah outliers <- len(outliers)

#PEMODELAN ANN
#Mendefinisikan nilai hyperparameter terbaik
learning rate <- 0.01
num_epoch <- 105
batch_size <- 16
num_neuron_hidden_layer <- 12
num_neuron_output <- 1

#Mendefinisikan nilai k-fold
n_folds = 4
kf <- KFold(n_splits=n_folds)

#Membuat variabel untuk menyimpan nilai RMSE pada setiap fold
rmse_scores = []

for train_idx, test_idx in kf.split(X,y):
    X_train, X_test = X[train_idx], X[test_idx]
    y_train, y_test = y[train_idx], y[test_idx]

    #Membentuk arsitektur ANN

```

```

model <- sequential()
model.add(Dense(X.shape[1],kernel_initializer='uniform'
, input_dim=X.shape[1]))
model.add(Dense(num_neurons_hidden_layer,
kernel_initializer='uniform', activation='relu'))
model.add(Dense(num_neurons_output, activation='linear'
))

#Menjalankan model yang dibentuk
model.compile(loss='mean_squared_error',
optimizer=tf.keras.optimizers.RMSprop(learning_rate),
metrics=[tf.keras.metrics.RootMeanSquaredError()])

#Melakukan pelatihan model
history = model.fit(X_train, y_train,
epochs=num_epochs, batch_size=batch_size,
validation_data=(X_test,y_test))

#Melakukan evaluasi model pada bagian train dan pada
bagian test
train_pred <- predict(X_train)
train_rmse <- rmse(y_train, train_pred)

test_pred <- predixt(X_test)
test_rmse <- rmse(y_test, test_pred)

#Menghitung nilai RMSE data testing pada setiap fold
rmse_scores.append(test_rmse)

output(mean(rmse_scores))

model.save('ann.h5')

```

### Source Code 5. Pemodelan Algoritma SVR

```

library(pandas)
library(numpy)
library(matplotlib)

```

```

# Load dataset
Data <- read_csv("/content/drive/My Drive/Tugas_Akhir/data-final-ta.csv")

# Checking outliers
data_check=data.drop(columns='Nama Sekolah')
mean = np.mean(data_check)
std_dev = np.std(data_check)
Mean <- mean
Standar deviasi <- std_dev
# count z-score
z_score <- (data_check - mean) / std_dev
print('Nilai z-score : ', z_score)
# identify outliers depends z-score value
outliers <- data_check[(z_score < batas_bawah) | (z_score > batas_atas)]
Jumlah outliers <- len(outliers)
# svr implementation
# initialization parameter
Kernel <- linear, polynomial, sigmoid, poly
C <- 0.1, 1, 10, 100
Epsilon <- 0.01, 0.1, 1
Gamma <- 0.01, 0.1, 1
# K-Fold Implementation
cv <- KFold(n_splits= 2 to 10, shuffle=True, random_state=42)
# Grid Search Implementation
# Implementasi GridSearch
grid_search <- GridSearchCV(model, parameters, cv=cv,
scoring="neg_root_mean_squared_error")
grid_search.fit(X, y)
Best Score <- grid_search.best_score
Best parameter <- grid_search.best_params
# model evaluation
rmse_scores = []
for train_index, test_index in cv.split(X, y):
    X_train, X_test <- X[train_index], X[test_index]
    y_train, y_test <- y[train_index], y[test_index]

    # Melakukan training model pada setiap fold

```

```
best_model.fit(X_train, y_train)

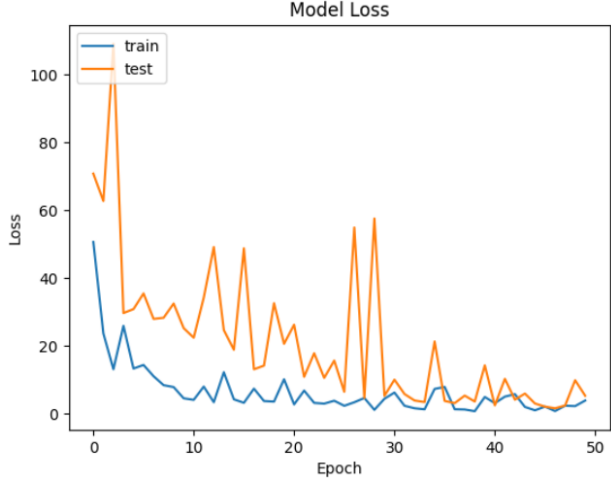
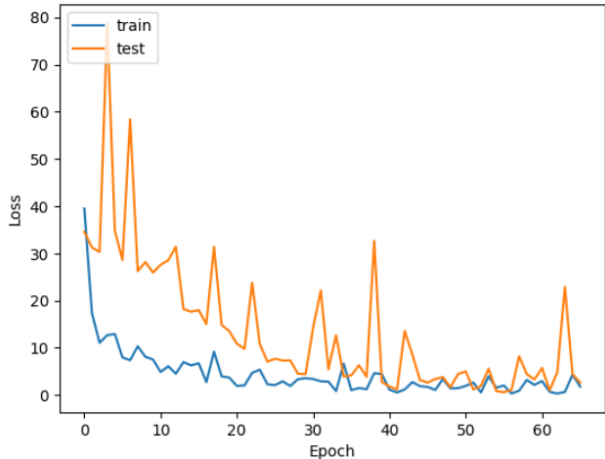
# Melakukan prediksi pada data testing
y_pred <- best_model.predict(X_test)

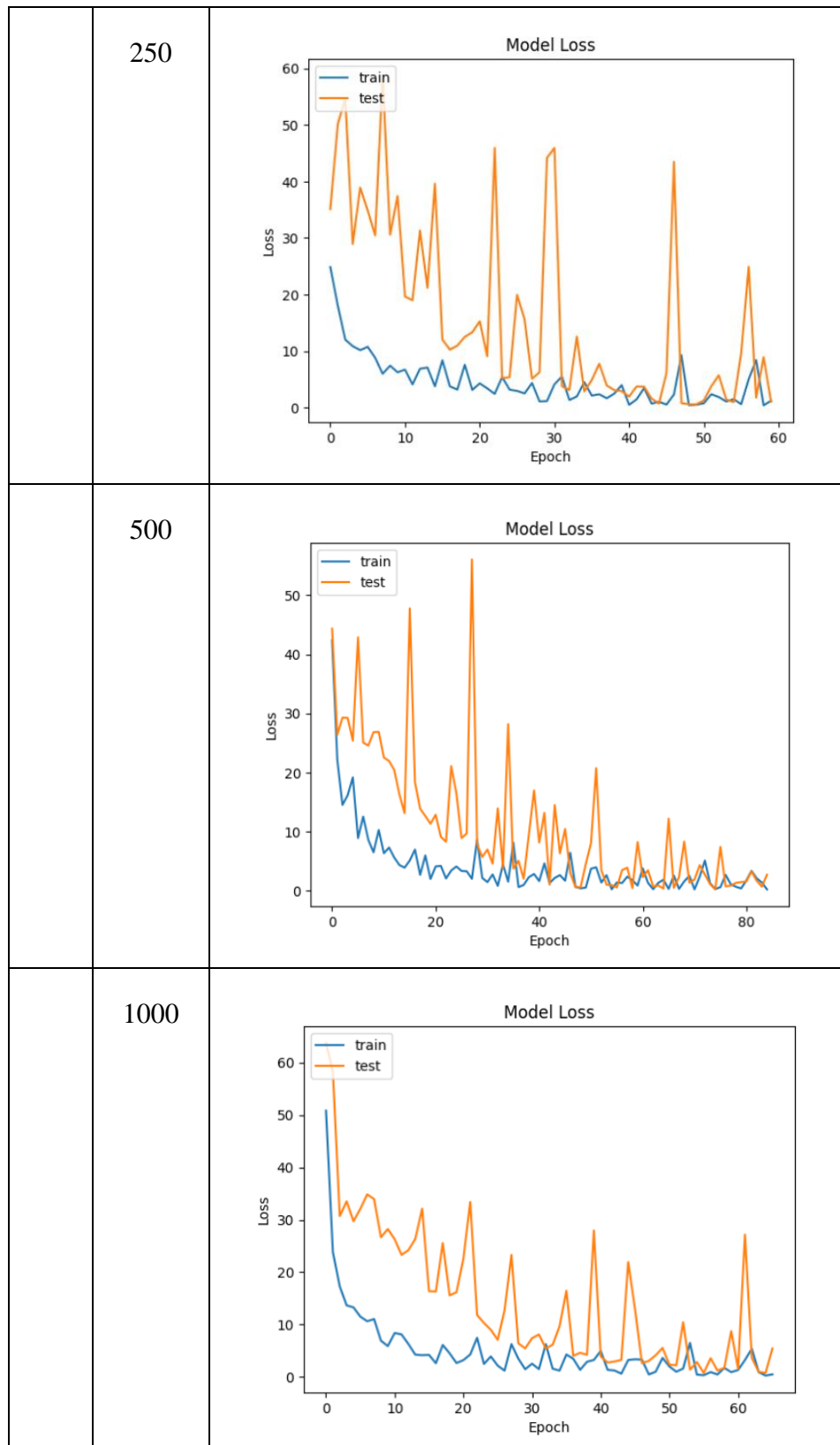
# Menghitung RMSE
rmse <- np.sqrt(mean_squared_error(y_test, y_pred))
rmse_scores.append(rmse)

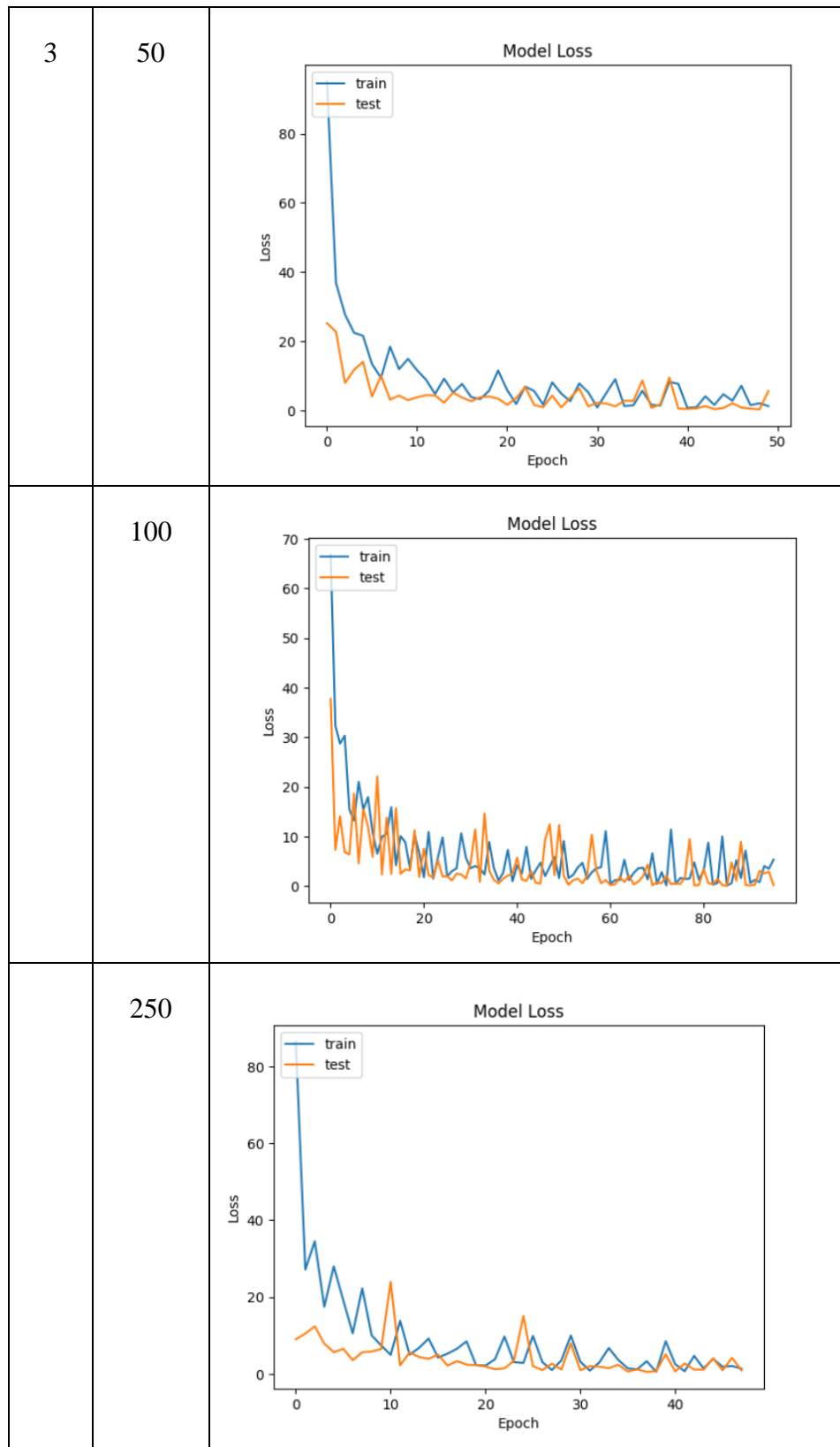
# Menghitung rata-rata nilai RMSE
avg_rmse = np.mean(rmse_scores)
RMSE <- avg_rmse
# predicting total registrans
```

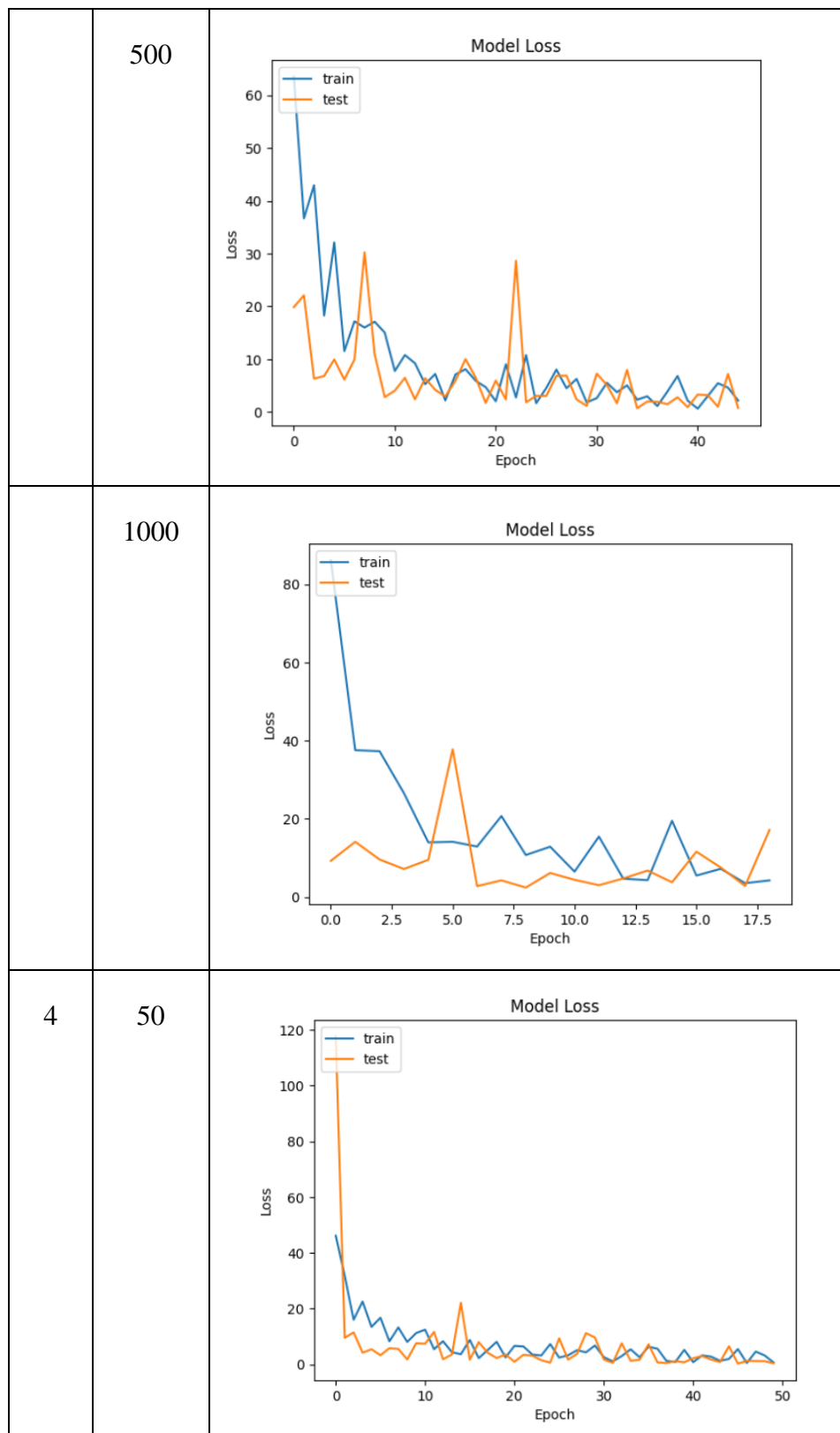
## Lampiran B Hasil Implementasi

### B.1. Hasil Implementasi Early Stopping

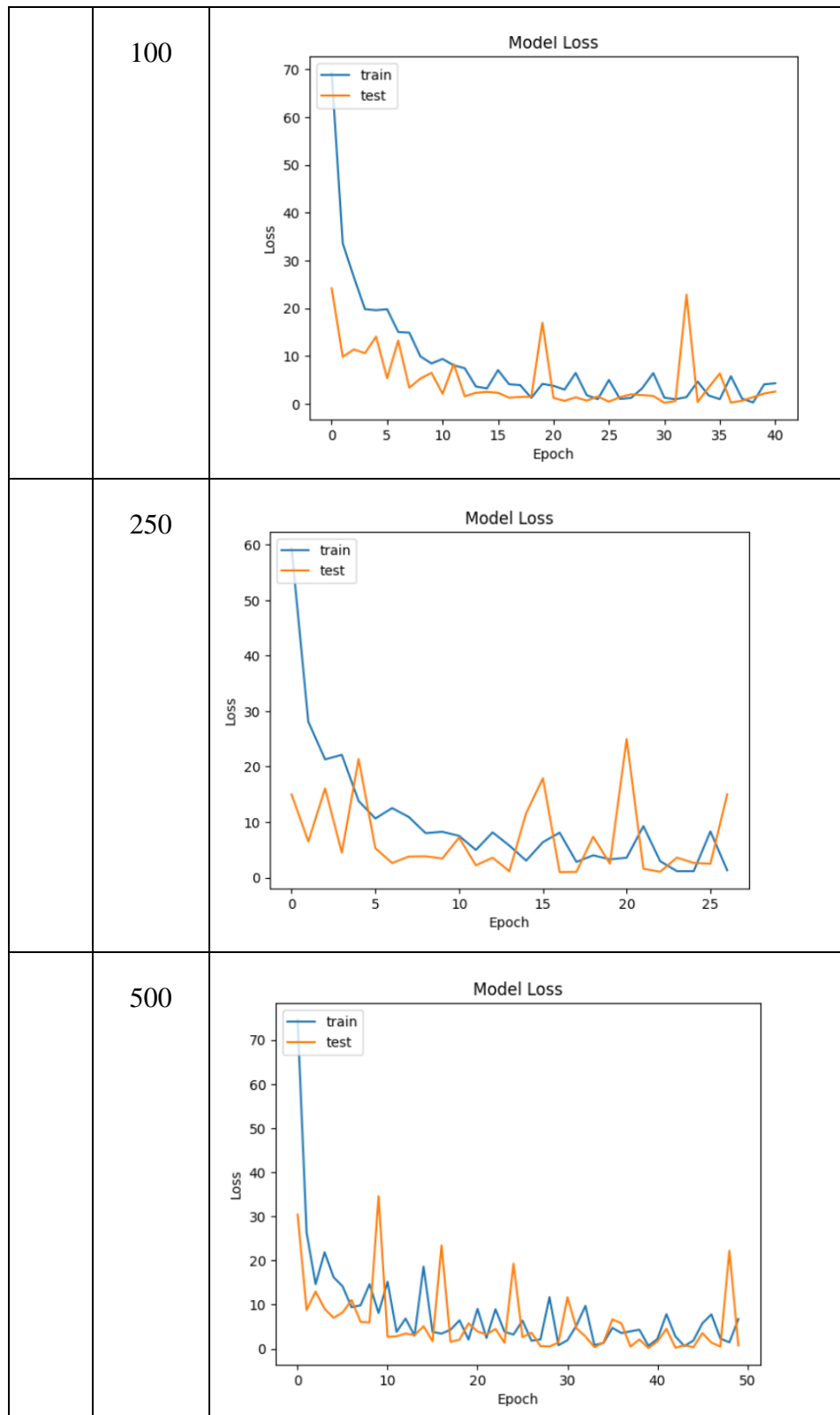
K	Epoch	Grafik Loss
2	50	
	100	

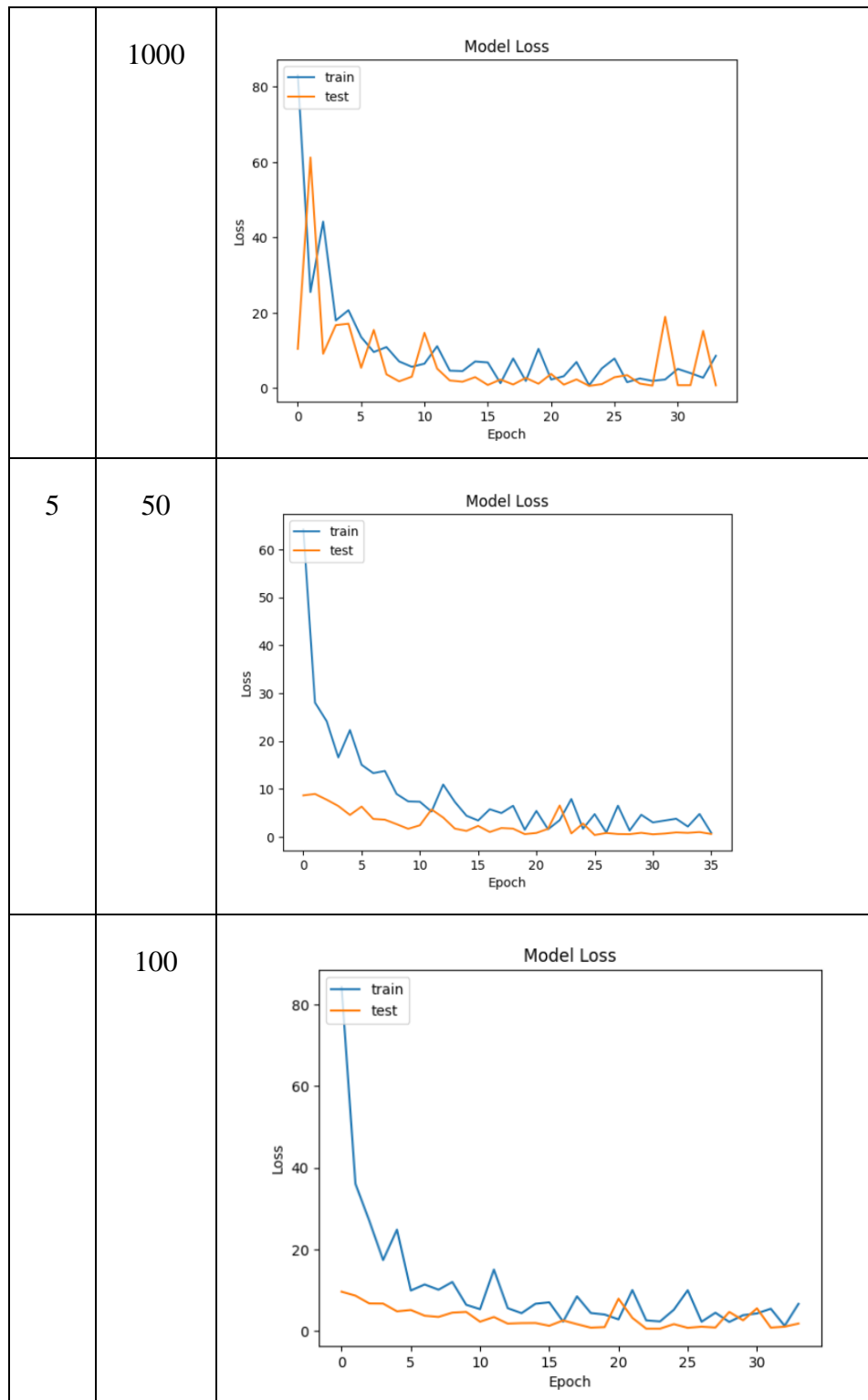


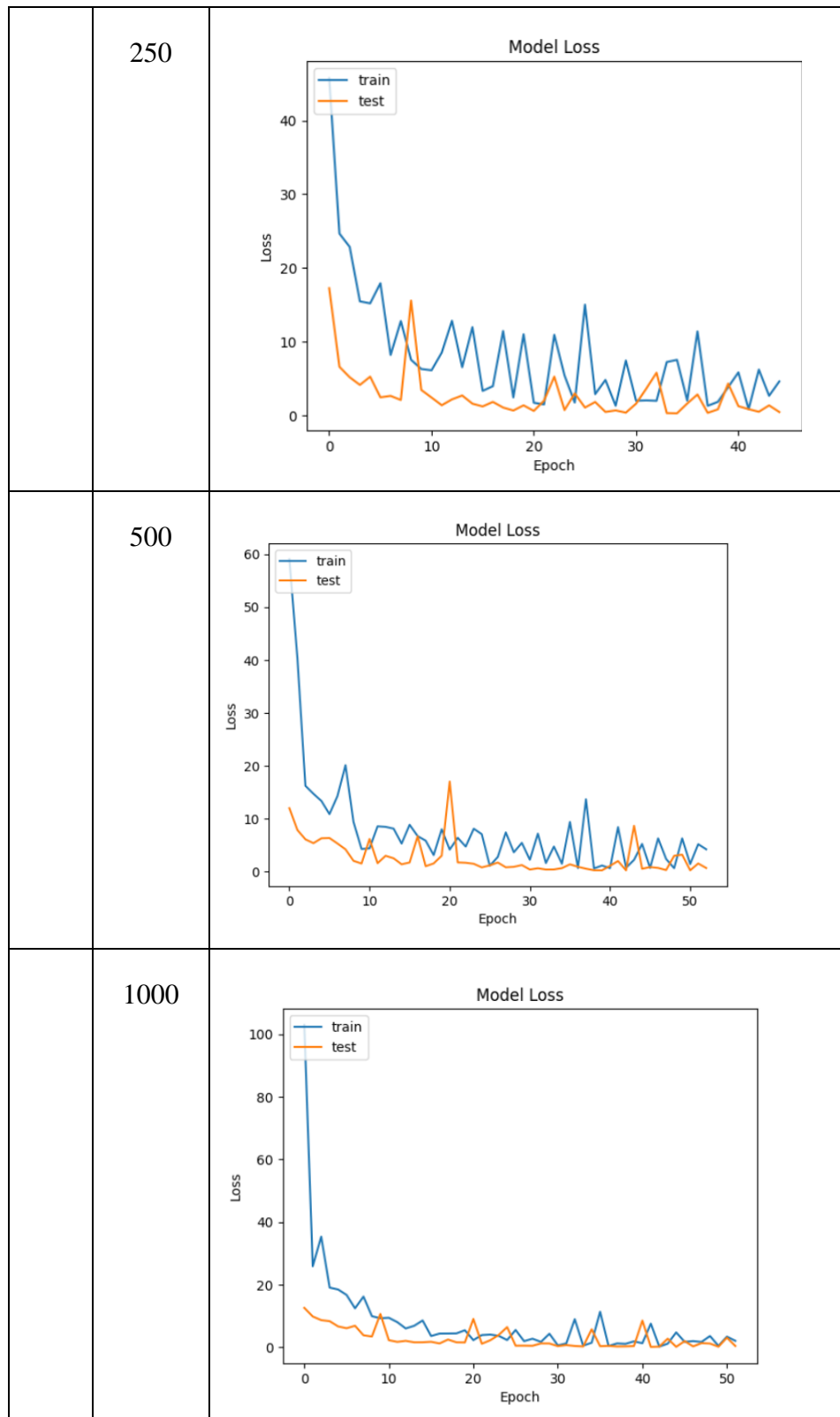


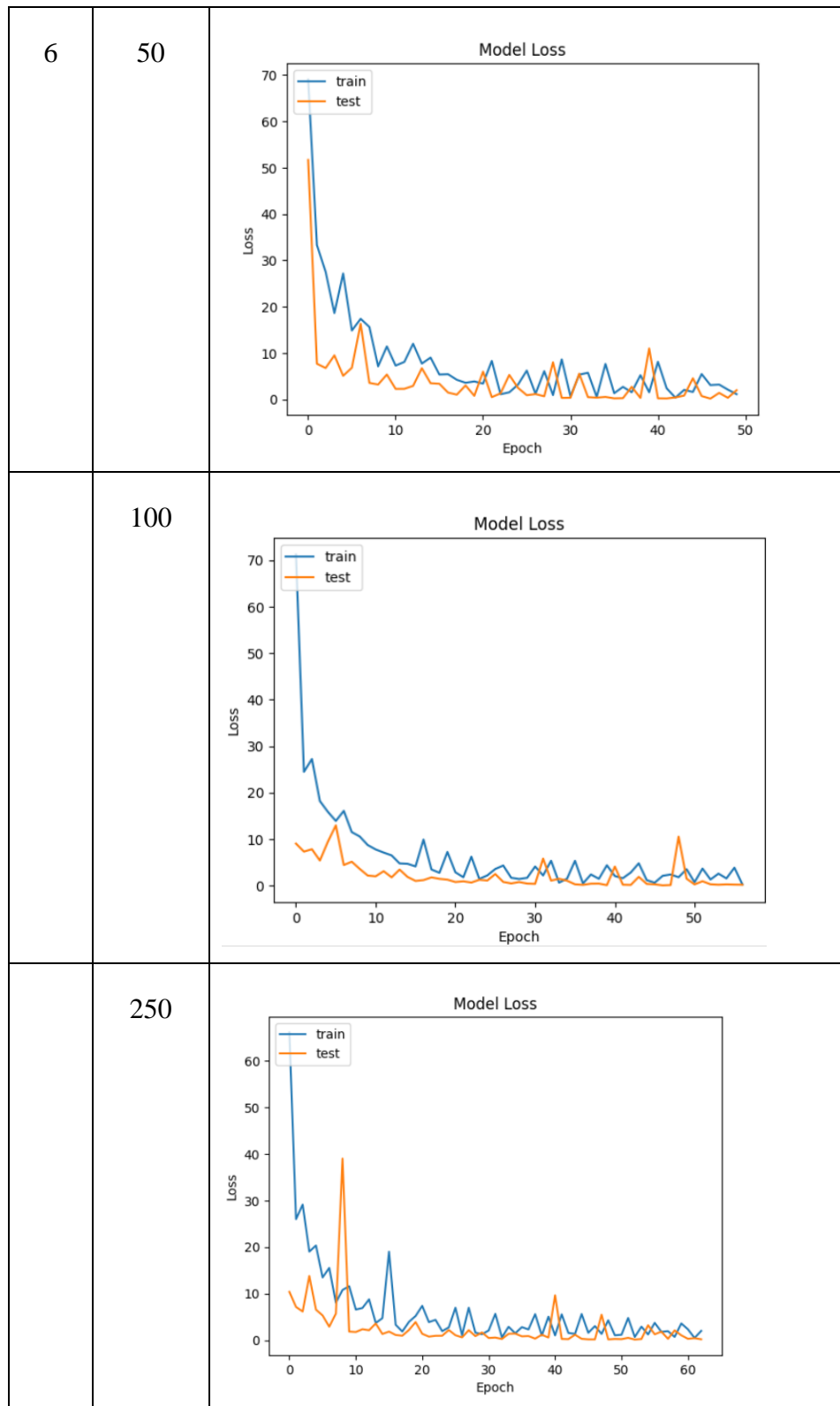


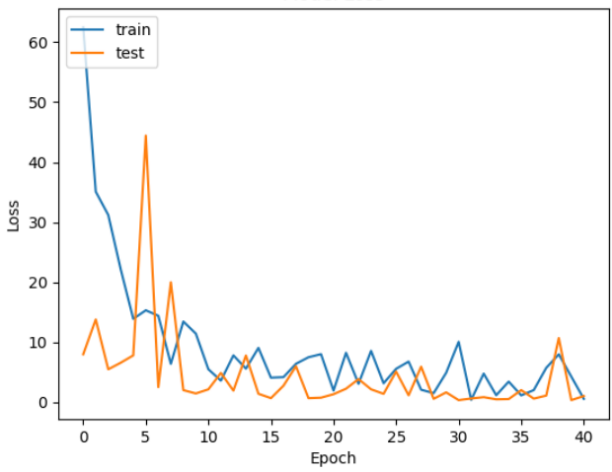
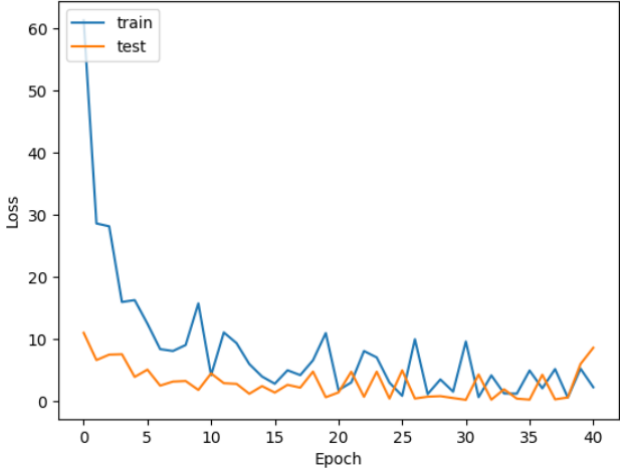
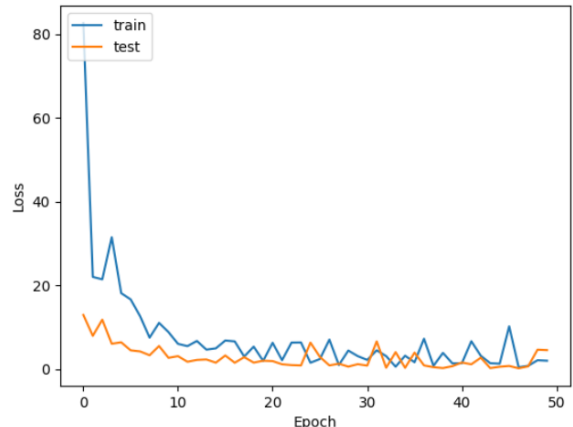


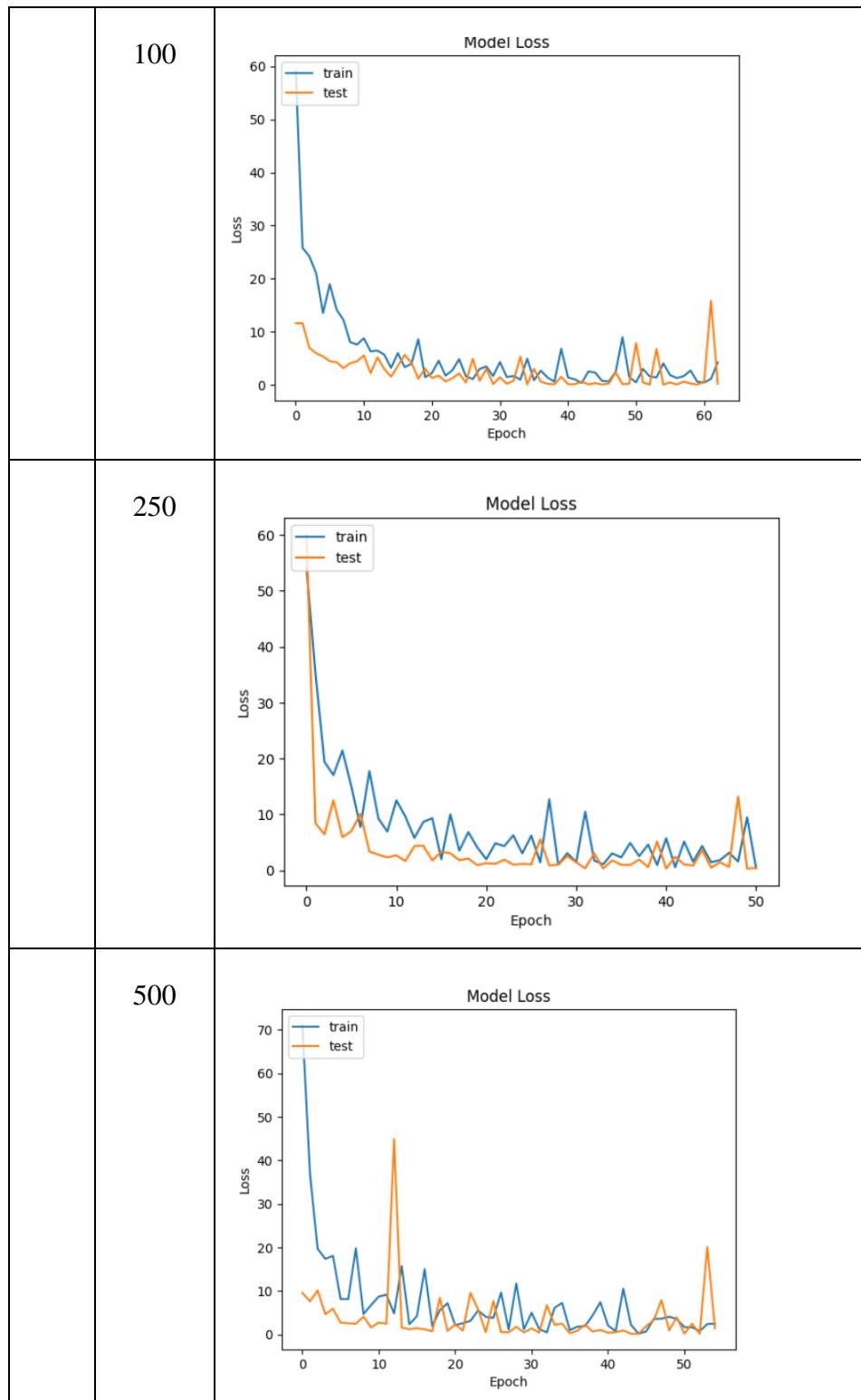


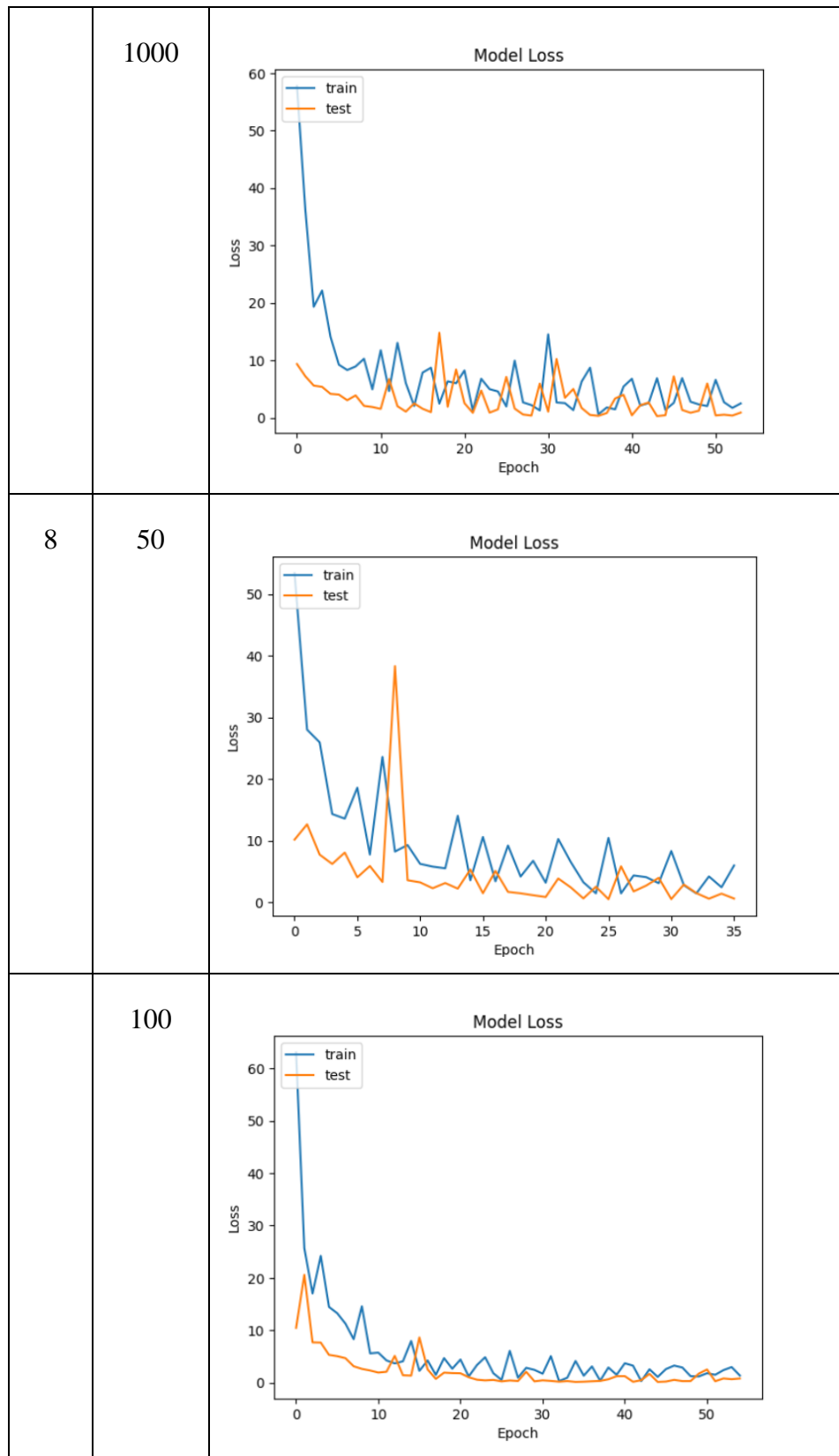


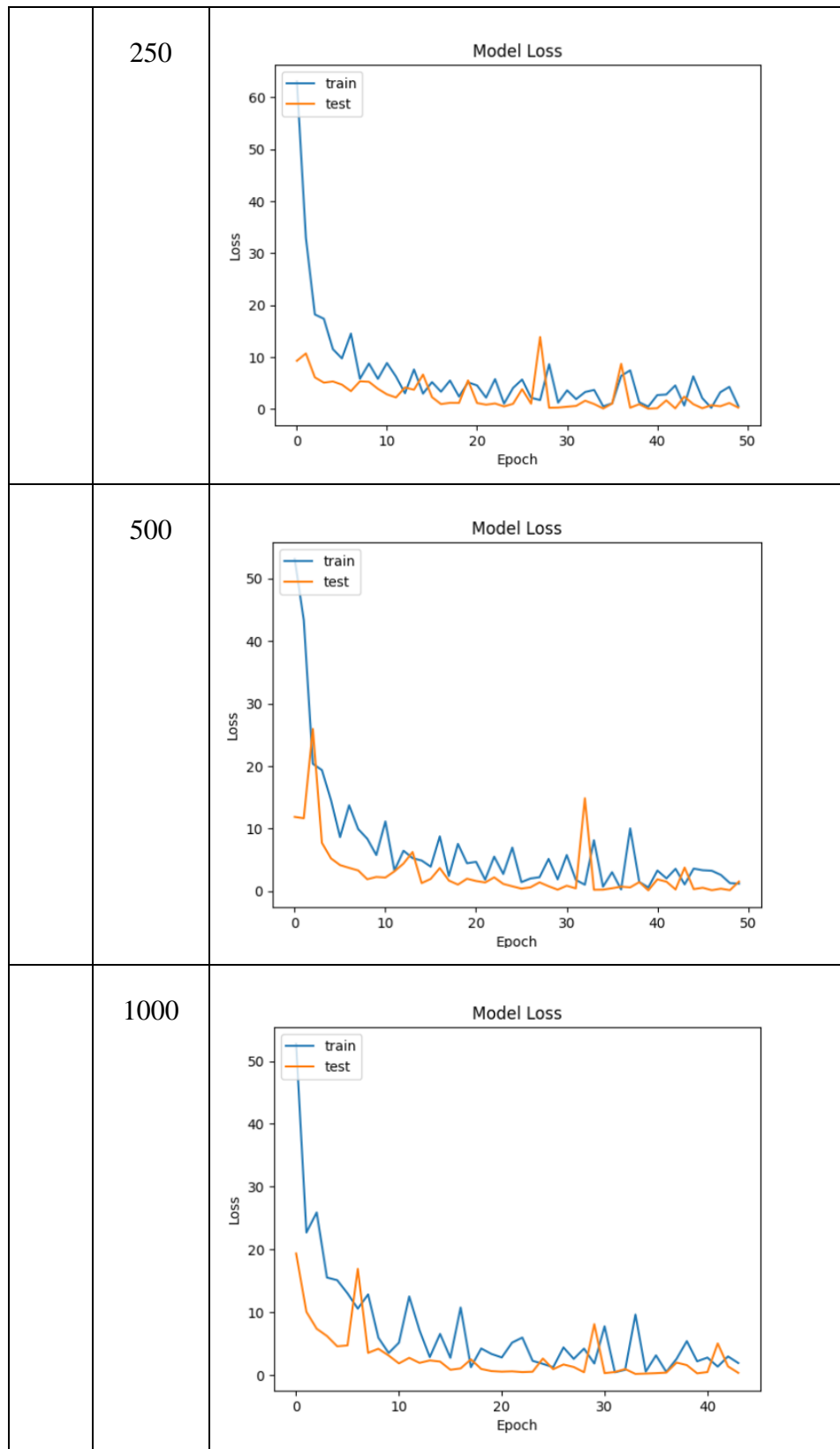




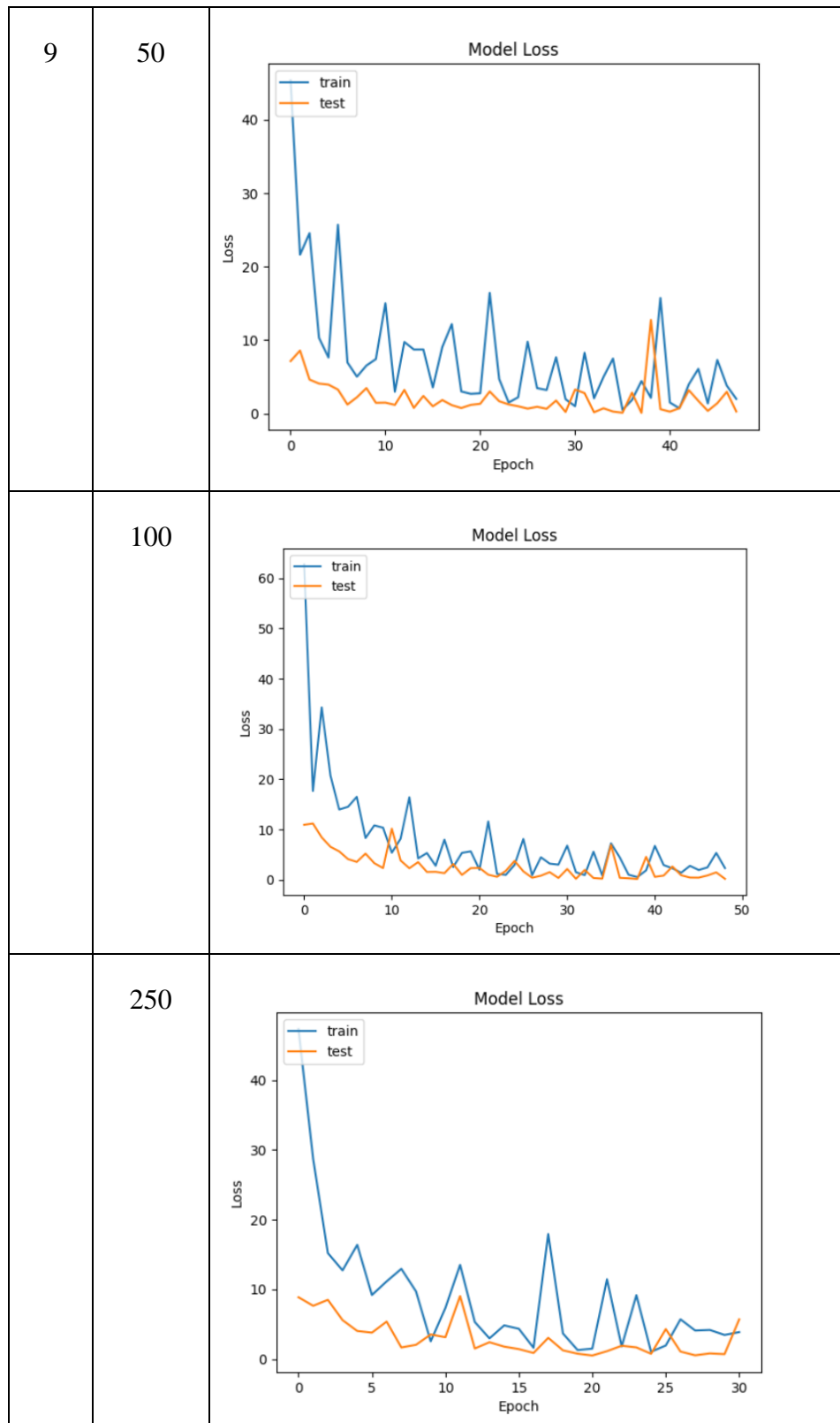
	500	<p>Model Loss</p>  <p>Loss</p> <p>Epoch</p>
	1000	<p>Model Loss</p>  <p>Loss</p> <p>Epoch</p>
7	50	<p>Model Loss</p>  <p>Loss</p> <p>Epoch</p>

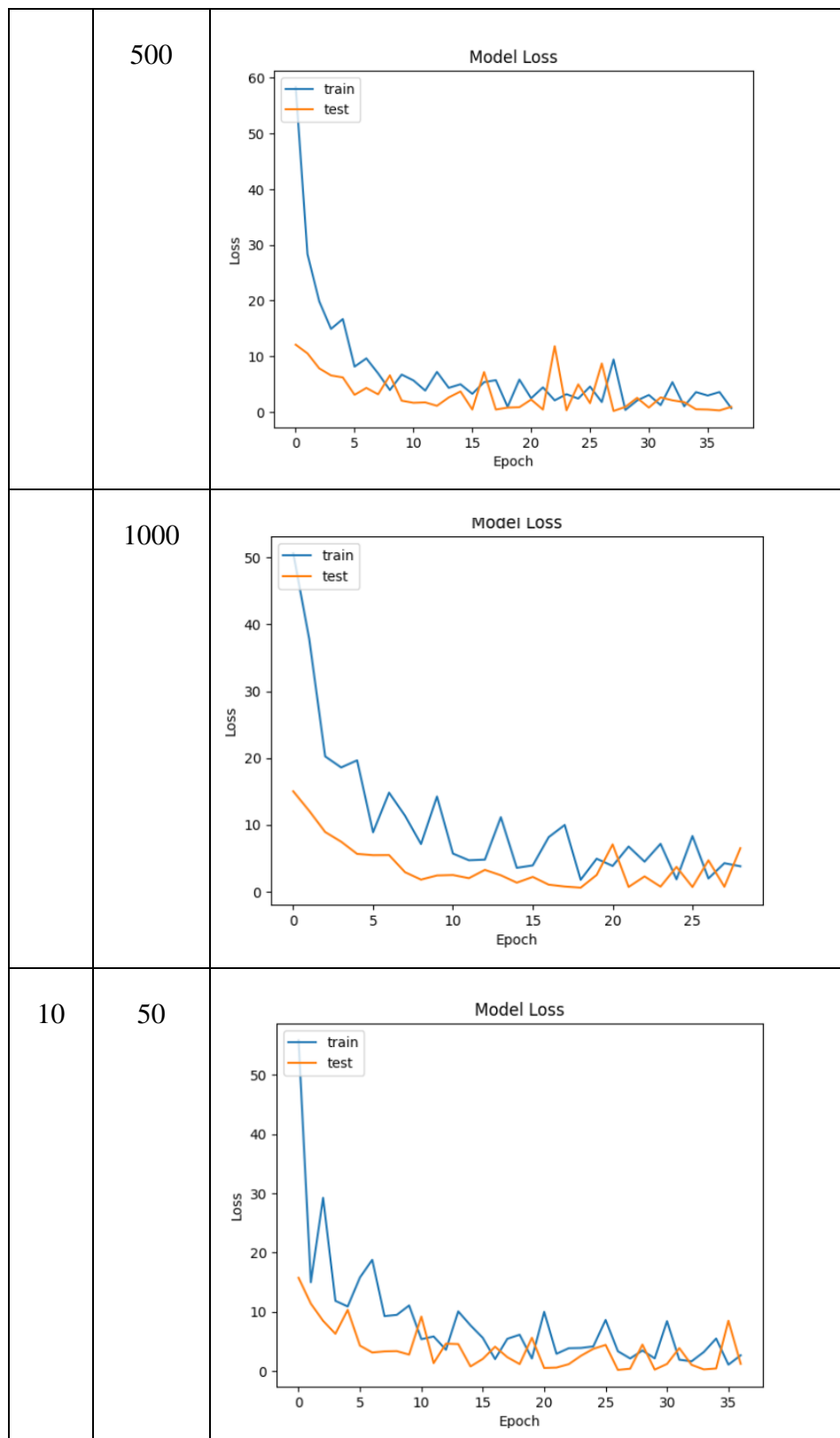


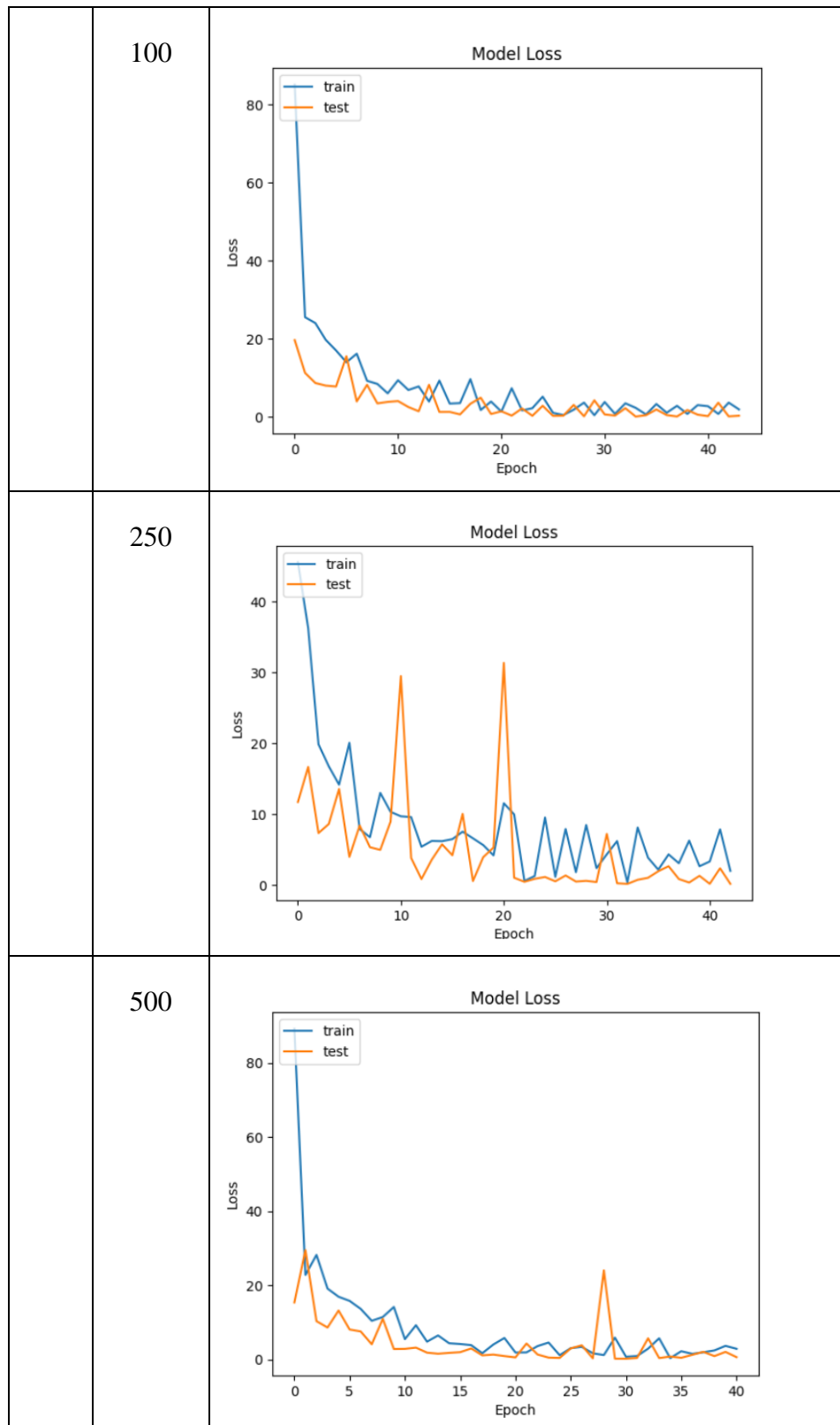


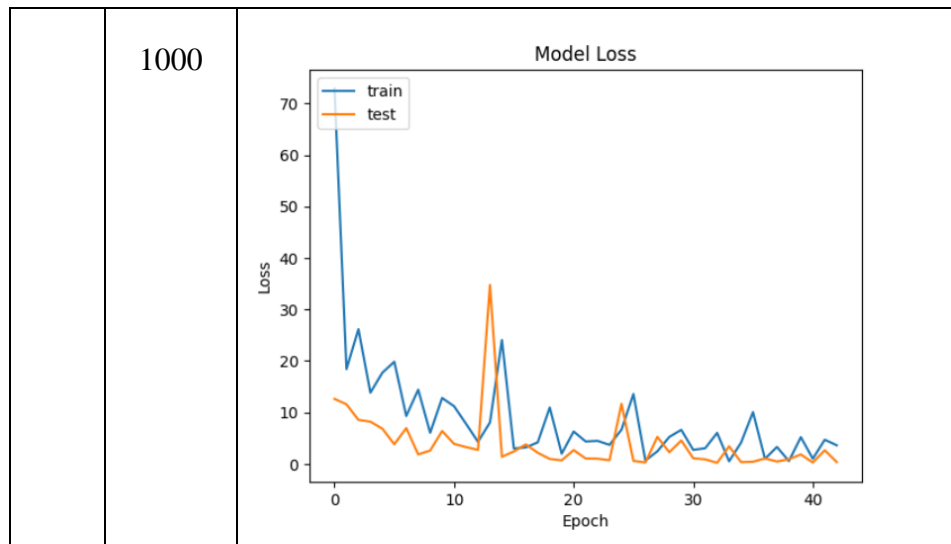












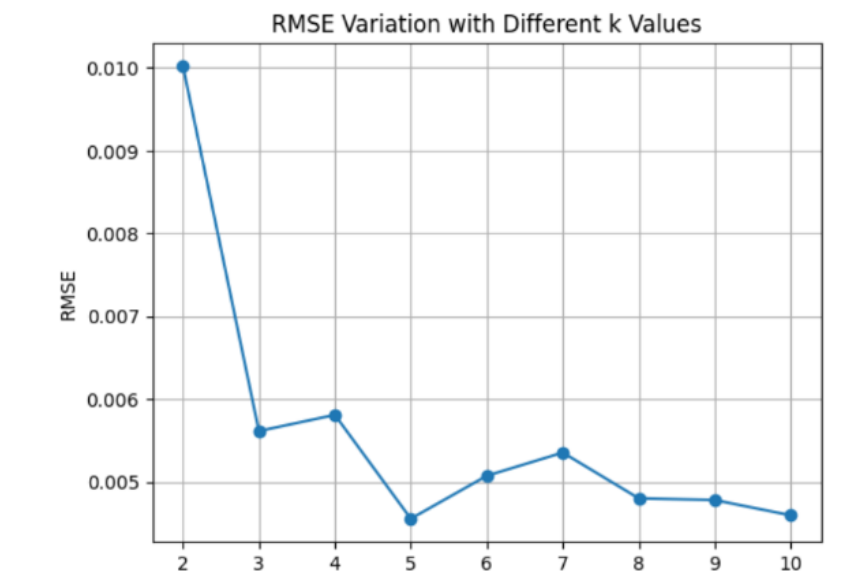
## B.2. Hasil Implementasi Artificial Neural Network

K	Epoch	RMSE
2	15	5.077675111568252
	30	3.308905963296935
	45	3.1125351998708717
	60	2.328428960954137
	75	2.3410310405597707
	90	2.8894081457364864
	105	0.9145345710529202
3	120	2.486604785602238
	15	4.349816690851296
	30	1.185050901482351
	45	1.419123984131077
	60	1.103317147379233
	75	1.2677344372812165

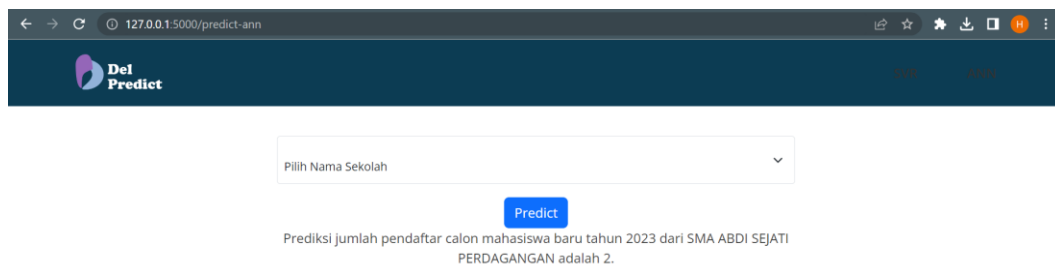
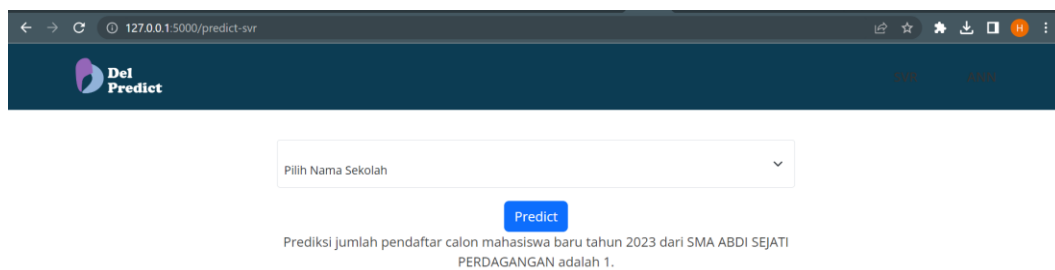
	90	0.7816953365196024
	105	0.8258383592647051
	120	1.3874999333541886
4	15	2.128592160228251
	30	3.6652698115711777
	45	1.602501021701588
	60	1.6238574635755945
	75	1.0774873164147554
	90	1.6037173323680127
	105	0.4929599799755078
	120	0.56299675144724
5	15	2.157844079885183
	30	1.1991356852105577
	45	0.9760183937827757
	60	1.326654052605502
	75	0.7243277702605629
	90	1.3416754523879937
	105	1.1916578445469415
	120	0.746181864831084
6	15	1.8877872681858854
	30	1.382243885936311
	45	1.0863240335818662
	60	1.319727154869274
	75	2.6038311594193204
	90	1.3773723994213725
	105	0.7115632375692152
	120	0.934355266523096
7	15	1.7597646597941885
	30	1.414998445687379
	45	1.7869493907399876

	60	1.5598473580813
	75	1.3347538608344744
	90	1.2210885548148813
	105	1.053785568079155
	120	0.88649628812792
8	15	1.7609847531652885
	30	1.2451541394536363
	45	1.7670223697444563
	60	1.5158721338459302
	75	0.8971200214387018
	90	0.5262559060639354
	105	1.66984355373789
	120	0.701008972141199
9	15	2.2511837984439484
	30	0.9656183070206177
	45	1.1292643517931757
	60	1.0534243143864066
	75	1.8833592365320015
	90	0.917752278888994
	105	0.5878529966755875
	120	1.3109461093875563
10	15	2.239774558701246
	30	1.8325242129539727
	45	1.4313917647647296
	60	1.774891534478397
	75	1.2483366230262396
	90	1.0172893094154662
	105	0.7662119998966219
	120	0.7320914619043708

### B.3. Hasil Implementasi Support Vector Regression



## Lampiran C Hasil Implementasi





## DAFTAR REFERENSI

- Abdel-Sattar, M., Aboukarima, A. M., & Alnahdi, B. M. (2021). Application of artificial neural network and support vector regression in predicting mass of ber fruits (*Ziziphus mauritiana* Lamk.) based on fruit axial dimensions. *PLoS ONE*, 16(1 January), 1–15. <https://doi.org/10.1371/journal.pone.0245228>
- Andreas, F., Mikhael, & Enri, U. (2022). *Perbandingan Algoritma Backpropagation Neural Network dan Long Short-Term Memory dalam Prediksi Harga Bitcoin*. 8(12), 547–558.
- Andriyani, W. (2021). *Apa perbedaan Root Mean Square Error (RMSE) dan Standard Deviation? Bagaimana cara penggunaannya?* Quora. <https://id.quora.com/Apa-perbedaan-Root-Mean-Square-Error-RMSE-dan-Standard-Deviation-Bagaimana-cara-penggunaannya#:~:text=RMSE digunakan untuk mengukur tingkat,nilai Profit berdasarkan nilai Sales.>
- Barus, T. E., & Simamora, N. S. P. (2020). Analisis Persebaran Pendaftar di Institut Teknologi Del Berdasarkan Asal Wilayah. *Prosiding Industrial Research ...*, 26–27. <https://jurnal.polban.ac.id/proceeding/article/view/2193>
- Early Stopping*. (2020). DeepLearning4J. <https://deeplearning4j.konduit.ai/v/en-1.0.0-beta7/getting-started/tutorials/early-stopping>
- Gunawan, M. I., Sugiarto, D., & Mardianto, I. (2020). Peningkatan Kinerja Akurasi Prediksi Penyakit Diabetes Mellitus Menggunakan Metode Grid Search pada Algoritma Logistic Regression. *Jurnal Edukasi Dan Penelitian Informatika (JEPIN)*, 6(3), 280. <https://doi.org/10.26418/jp.v6i3.40718>
- Handayani, A., Jamal, A., & Septiandri, A. A. (2017). 350-565-1-Sm. 6(4), 394–403.
- Ishukatiyar. (2023). *Backpropagation in Data Mining*. GeeksForGeeks. <https://www.geeksforgeeks.org/backpropagation-in-data-mining>
- Itedel. (2022). *Pendaftaran Mahasiswa Baru IT Del*. <http://spmb.del.ac.id/>
- Khoirudin, K., Nurdiyah, D., & Wakhidah, N. (2019). Prediksi Penerimaan Mahasiswa

- Baru Dengan Multi Layer Perceptron. *Jurnal Pengembangan Rekayasa Dan Teknologi*, 14(1), 1. <https://doi.org/10.26623/jprt.v14i1.1212>
- Lembong, R. I. (2022). Machine Learning Menggunakan Perbandingan Dua Algoritma Antara Support Vector Regression Dan Decision Tree Untuk Memprediksi Harga Saham Agro. In *upj.ac.id* (Vol. 33, Issue 1).
- Mahyunis, R. V. (2022). Penerapan Artificial Neural Network (ANN) Menggunakan Algoritma Backpropagation dengan Membandingkan Empat Fungsi Aktivitas dalam Memprediksi Harga Emas. *Braz Dent J.*, 33(1), 1–12.
- Maryana, T., Kusriani, K., & Fatta, H. Al. (2019). Analisis Perbandingan Prediksi Obat Dengan Menggunakan Metode Abc Analisis Dan Svr Pada Aplikasi “Morbis.” *Jurnal Teknologi Informasi*, 3(2), 174. <https://doi.org/10.36294/jurti.v3i2.1016>
- Muhadzdzab, H., Asfi, M., & Putri, T. E. (2020). Sistem Prediksi untuk Menentukan Jumlah Pendaftaran Mahasiswa Baru pada Universitas Catur Insan Cendekia Menggunakan Metode Least Square. *Jurnal Informatika Universitas Pamulang*, 5(3), 350. <https://doi.org/10.32493/informatika.v5i3.6598>
- Mustafeez, A. Z. (2023). *What is Early Stopping*. Educative.Io. <https://www.educative.io/answers/what-is-early-stopping>
- Nafi'iyah, N. (2016). Perbandingan Regresi Linear , Backpropagation Dan Fuzzy Mamdani Dalam Prediksi Harga Emas. *Seminar Nasional Inovasi Dan Aplikasi Teknologi Di Industri*, 291–296.
- Primartha, R. (2018). *Belajar Machine Learning Teori dan Praktik*.
- Putra, H., & Ulfa Walmi, N. (2020). Penerapan Prediksi Produksi Padi Menggunakan Artificial Neural Network Algoritma Backpropagation. *Jurnal Nasional Teknologi Dan Sistem Informasi*, 6(2), 100–107. <https://doi.org/10.25077/teknosi.v6i2.2020.100-107>
- Rais, Z. (2022). Analisis Support Vector Regression (Svr) Dengan Kernel Radial Basis Function (Rbf) Untuk Memprediksi Laju Inflasi Di Indonesia. *VARIANSI: Journal of Statistics and Its Application on Teaching and Research*, 4(1), 30–38.

<https://doi.org/10.35580/variensiunm13>

- Ridwan, D. I., Setianingsih, C., & Murti, M. A. (2021). *Prediksi Penggunaan Energi Listrik Support Vector Regression Prediction of Electricity Using Support Vector Regression Method*. 8(6), 12135–12144.
- Rizal, N. A. (2014). *Implementasi Metode Hybrid Jst-Som Pada Prediksi Churn Pelanggan Seluler: Studi Kasus Pt . Telekomunikasi Seluler Implementation Hybrid Nn-Som Method for Prediction Customer Churn: Study Case Pt . Telekomunikasi Seluler*.
- Rusdi, Z., Lubis, C., & Tjandra, V. G. (2021). *Prediksi Kurs Mata Uang dengan Metode Long Short Term Memory Berbasis Attention*.
- Saadah, S., Z, F. Z., & Z, H. H. (2021). Support Vector Regression (SVR) Dalam Memprediksi Harga Minyak Kelapa Sawit di Indonesia dan Nilai Tukar Mata Uang EUR/USD. *Journal of Computer Science and Informatics Engineering (J-Cosine)*, 5(1), 85–92. <https://doi.org/10.29303/jcosine.v5i1.403>
- Santoso, H., & Putri, R. A. (2023). *Deteksi Komentar Cyberbullying pada Media Sosial Instagram Menggunakan Algoritma Random Forest Cyberbullying Comment Detection on Instagram Social Media Using Random Forest Algorithm*. 13(April), 62–72.
- Saputra, G. H., Wigena, A. H., & Sartono, B. (2019). Penggunaan Support Vector Regression Dalam Pemodelan Indeks Saham Syariah Indonesia Dengan Algoritme Grid Search. *Indonesian Journal of Statistics and Its Applications*, 3(2), 148–160. <https://doi.org/10.29244/ijsa.v3i2.172>
- Sepri, D., & Fauzi, A. (2020). Prediksi Harga Cabai Merah Menggunakan Support Vector Regression. *Computer Based Information System Journal*, 8(2), 1–5. <https://doi.org/10.33884/cbis.v8i2.1921>
- Setyoningrum, N. R., Rahimma, P. J., Teknologi, S. T., Tanjungpinang, I., & Tanjungpinang, K. (2022). Implementasi Algoritma Regresi Linear Dalam Sistem Prediksi Pendaftar Mahasiswa Baru Sekolah Tinggi Teknologi Indonesia Tanjungpinang. *Prosiding Seminar Nasional Ilmu Sosial Dan Teknologi*

(*SNISTEK*), 4, 13–18.  
<https://ejournal.upbatam.ac.id/index.php/prosiding/article/view/5200>

Situmorang, Y. S. (2022). Analisis Pengaruh Holt Winter Exponential Smoothing sebagai Metode Optimasi Pemulusan Data Pada Model Peramalan Backpropagation Neural Network. *IT Del*, 1.

Somya, R. (2018). Perancangan Aplikasi Chatting Berbasis Web di PT. Pura Barutama Kudus menggunakan Socket.IO dan Framework Foundation. *Khazanah Informatika : Jurnal Ilmu Komputer Dan Informatika*, 4(1), 8–15.  
<https://doi.org/10.23917/khif.v4i1.5979>

Yuniarti, D., & Gunawan, B. (2021). Penerapan Data Mining Untuk Memprediksi Jumlah Siswa Diktukpa (Pendidikan Pembentukan Perwira) TNI AL Dengan Metode Artificial Neural Network Dan Algoritma Backpropagation. 5(4), 938–954.  
<https://doi.org/10.52362/jisamar.v5i4.574>

