

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

DAFTAR ISI.....	ii
DAFTAR GAMBAR.....	iv
DAFTAR TABEL	iv
BAB 1 PENDAHULUANN.....	1
1.1 Latar Belakang.....	1
1.2 Rumusan Masalah	4
1.3 Tujuan Penelitian.....	4
1.4 Ruang Lingkup Penelitiann	4
1.5 Hasil yang Diharapkann	5
1.6 Tahapan Penelitiann	5
1.7 Sistematika Penyajian.....	6
BAB 2 TINJAUAN PUSTAKA.....	9
2.1 Penelitian Terkait.....	9
2.2 Pendaftaran Calon Mahasiswa Baru.....	11
2.3 Machine Learningg.....	12
2.4 Regression	14
2.5 Support Vector Machine (SVM)	15
2.6 Support Vector Regression (SVR)	16
2.7 Grid Search.....	19
2.8 Artificial Neural Network.....	19
2.9 Backpropagation.....	21
2.10 Data Preprocessing	27
2.11 Evaluasi Hasil.....	28
2.11.1 Mean Square Error	28
2.11.2 Root Mean Square Error	29
2.12 Normalisasi.....	29
2.13 Flask	30
BAB 3 ANALISIS.....	31
3.1 Analisis Data	31
3.1.1 Sumber Data.....	31
3.1.2 Seleksi Data.....	32
3.2 Preprocessing Data	32
3.2.1 Data Integration.....	32
3.2.2 Data Cleaning.....	33
3.2.3 Pengubahan Variabel Teks ke Bentuk Numerik dengan Teknik Encoding	36
3.2.4 Normalisasi Dataset	36
3.3 Analisis Algoritma SVR.....	38
3.3.1 Perhitungan SVR.....	39
3.4 Analisis Algoritma ANN Backpropagation.....	45
3.4.1 Inisialisasi Bias dan Bobot	45
3.4.2 Inisialisasi Hyperparameter.....	46
3.4.3 Feedforward Propagation	46
3.4.4 Backpropagation Propagation	49
3.5 Analisis Evaluasi Model.....	49
BAB 4 PERANCANGAN	51

4.1 Rancangan Desain Penelitian	51
4.2 Rancangan Pre-Processing Data.....	53
4.3 Rancangan Implementasi Algoritma	54
4.3.1 Implementasi Algoritma SVR.....	54
4.3.2 Implementasi Algoritma ANN Backpropagation.....	55
4.4 Use Case Diagram	57
4.5 Desain Prototype	57
BAB 5 IMPLEMENTASI.....	60
5.1 Lingkungan Implementasi	60
5.2 Batasan Implementasi.....	61
5.3 Implementasi Pengumpulan Data.....	61
5.4 Implementasi Preprocessing	61
5.4.1. Data Integration.....	62
5.4.2. Data Cleaning.....	63
5.4.3. Encoding	63
5.4.2. Normalisasi	63
5.5 Implementasi Algoritma	64
5.5.1. Support Vector Regression	64
5.5.2. Artificial Neural Network	65
BAB 6 HASIL DAN PEMBAHASAN	67
6.1. Pembahasan Preprocessing Data	67
6.1.1. Pembahasan Encoding	67
6.1.2. Pembahasan Normalisasi	68
6.2. Pembahasan Algoritma.....	70
6.2.1. Pembahasan dan Hasil Pemodelan Support Vector Regression	70
6.2.2. Pembahasan dan Hasil Pemodelan Artificial Neural Network	73
6.3. Pembahasan Hasil Evaluasi Model	74
BAB 7 KESIMPULAN DAN SARAN	75
7.1. Kesimpulan.....	75
7.1. Saran	75
LAMPIRAN.....	76
Lampiran A Source Code	76
Source Code 1. Encoding Data	76
Source Code 2. Grid Search ANN	76
Source Code 3. Grid Search SVR	78
Source Code 4. Pemodelan Algoritma ANN	79
Source Code 5. Pemodelan Algoritma SVR	83
Lampiran B Hasil Implementasi Algoritma	85
B.1. Hasil Implementasi Artificial Neural Network	85
B.2. Hasil Implementasi Support Vector Regression.....	101
Lampiran C Aplikasi Web Sistem Prediksi Del Predict.....	102
DAFTAR REFERENSI	103

DAFTAR GAMBAR

Gambar 2. 1 Jumlah Pendaftar Calon Mahasiswa Baru IT Del 2016 - 2022	11
Gambar 2. 2 Ilustrasi SVM	15
Gambar 2. 3 Ilustrasi SVR	18
Gambar 2. 4 Arsitektur Algoritma <i>Artificial Neural Network</i>	20
Gambar 4. 1 Rancangan desain penelitian yang akan dilakukan	51
Gambar 4. 2 Rancangan pelaksanaan preprocessing data.....	53
Gambar 4. 3 Flow algoritma <i>Support Vector Regression</i>	54
Gambar 4. 4 Flow algoritma ANN <i>Backpropagation</i>	55
Gambar 4. 5 Rancangan use case aplikasi web.....	57
Gambar 4. 6 Rancangan halaman prediksi.....	58
Gambar 4. 7 Rancangan halaman hasil prediksi	59
Gambar 5. 1 Data mentah yang diperoleh dari BAAK	62
Gambar 5. 2 Data hasil integration	62

DAFTAR TABEL

Tabel 3. 1 Informasi yang diperoleh dari BAAK IT Del	31
Tabel 3. 2 Atribut pada dataset	32
Tabel 3. 3 Dataset hasil integration.....	33
Tabel 3. 4 Dataset sebelum dilakukan missing value	34

Tabel 3. 5 Dataset sesudah dilakukan missing value	35
Tabel 3. 6 Sampel Dataset sebelum normalisasi.....	36
Tabel 3. 7 Sampel Dataset setelah normalisasi	37
Tabel 3. 8 Inisialisasi nilai parameter	39
Tabel 3. 9 Data Uji yang sudah dinormalisasikan	39
Tabel 3. 10 Data Uji	40
Tabel 3. 11 Pasangan Data.....	40
Tabel 3. 12 Hasil Perhitungan RBF	41
Tabel 3. 13 Hasil Perhitungan Matriks Hessian.....	41
Tabel 3. 14 Hasil Perhitungan Nilai Error	42
Tabel 3. 15 Hasil Perhitungan Delta Alpha	42
Tabel 3. 16 Hasil Perhitungan Nilai Alpha Baru	43
Tabel 3. 17 Hasil Perhitungan Nilai Alpha Star.....	43
Tabel 3. 18 Data Uji yang Ditetes	44
Tabel 3. 19 Hasil Alpha Star yang didapat sebelumnya	44
Tabel 3. 20 Nilai bias dari input layer ke hidden layer	45
Tabel 3. 21 Nilai bobot dari inputan layer ke hidden layer.....	45
Tabel 3. 22 Nilai bias dari hidden layer ke output layer	46
Tabel 3. 23 Nilai bobot dari hidden layer ke output layer	46

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 tingkat akurasi yang diperoleh setelah melakukan prediksi dengan kedua algoritma tersebut ?

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 hasil prediksi yang dilakukan oleh algoritma *Support Vector Regression* dan *Artificial Neural Network* (ANN) dan menentukan evaluasi model kedua algoritma.

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 lain nama sekolah, wilayah kabupaten 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. Penelitian ini menggunakan model normalisasi *min-max normalization*.
5. Sistem prediksi yang dilakukan hanya akan memprediksi jumlah pendaftar untuk 1 tahun selanjutnya.
6. Simulator algoritma akan dibuat dengan membuat aplikasi website.
7. 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 evaluasi 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 penggerjaan 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).

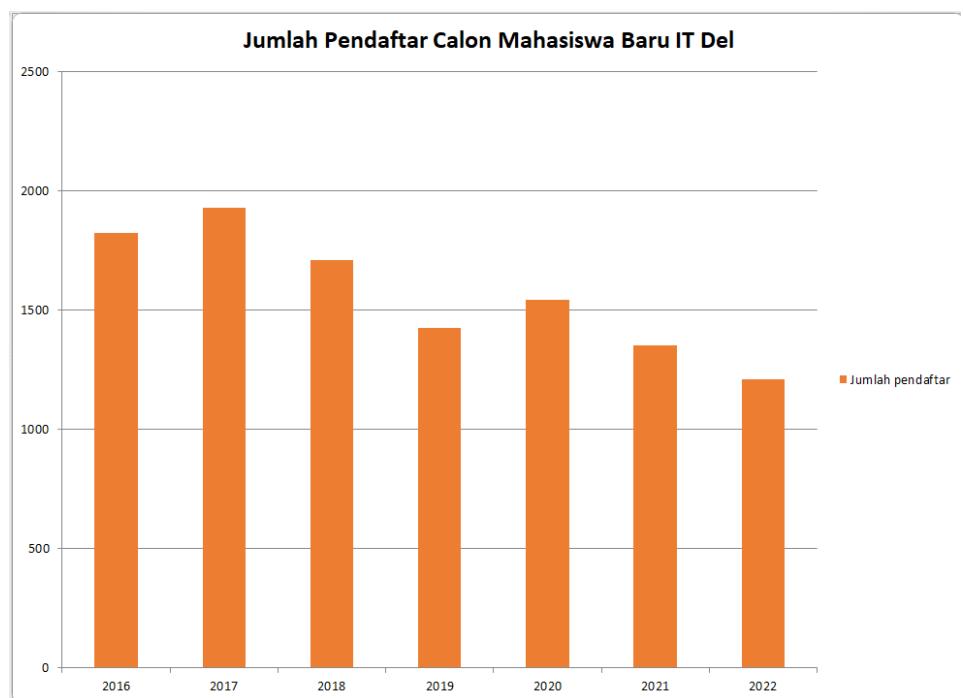
4. Perbandingan Peramalan Jumlah Penumpang Bus Damri Yogyakarta Menggunakan Metode Artificial Neural Network Dan Support Vector Regression. Penelitian dilakukan dengan menggunakan algoritma Artificial Neural Network Dan Support Vector Regression. Dari hasil penelitian ini didapatkan bahwa ANN mendapatkan model terbaik pada jumlah neuron hidden ke 30 dan target error yang tercapai adalah 0,0143 dan nilai MAPE 14.27%. dan algoritma SVR dengan menggunakan model kernel RBF yaitu nilai MAPE sebesar 2,46%. Berdasarkan didapatkan hasil bahwa algoritma SVR merupakan metode terbaik untuk meramalkan atau memprediksi jumlah penumpang bus DAMRI untuk hari kedepannya karena memiliki tingkat error paling kecil yaitu 2.46% (Maulani, 2019).

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 meuron-neuron terkoneksi satu sama lain sehingga membentuk sebuah jaringan neuron yang sangat rumit.

(Primartha, 2018)

2.4 Regression

Regression (regresi) adalah metode analisis statistika yang digunakan untuk menggambarkan suatu model hubungan keterkaitan antara dua variabel atau lebih. Dalam model keterkaitan tersebut variabel yang terlibat dikelompokkan menjadi Variabel bebas (independen) sebagai variabel penduga (*predictor*) yang biasanya disimbolkan dengan huruf “X” dan variabel terikat (dependen) sebagai variabel respon biasanya dilambangkan dengan huruf “Y”. Hubungan antara variabel-variabel ini dapat direpresentasikan sebagai fungsi linier maupun fungsi non-linier. Model ini juga dapat digunakan dalam proses optimisasi, misalkan: untuk mendapatkan nilai suhu yang mampu memaksimalkan kualitas produk hasil olahan, biaya iklan yang tepat untuk mendapatkan volume penjualan optimal dan sebagainya. Kelebihan regresi antara lain dapat digunakan untuk memprediksi nilai variabel respon dengan derajat kepercayaan tertentu jika nilai variabel bebas diketahui.

Regression (Regresi) merupakan metode analisis statistika yang berkaitan dengan suatu variabel tak bebas (dependent variable) pada satu atau lebih variabel yang bertujuan untuk melakukan perkiraan atau peramalan nilai – nilai dari variabel yang telah diketahui. Variabel yang menerangkan dinamakan variabel bebas (independent variable). Dalam regresi linier terdapat variabel bebas X dan Y. Tujuan regresi linier yaitu menentukan satu persamaan serta garis yang menunjukkan hubungan antara variabel bebas dan variabel yang tidak bebas yang mana merupakan persamaan penduga yang berfungsi untuk memprediksi variabel tidak bebas. Pembelajaran hubungan antara setiap variabel terdiri dari dua bentuk, yaitu:

1. Analisis regresi sederhana (simple analisis regresi)

Regresi sederhana merupakan hubungan antara dua variabel yaitu variabel bebas (independent variable) dan variabel tidak bebas (dependent variable).

2. Analisis regresi berganda (multiple analisis regresi)

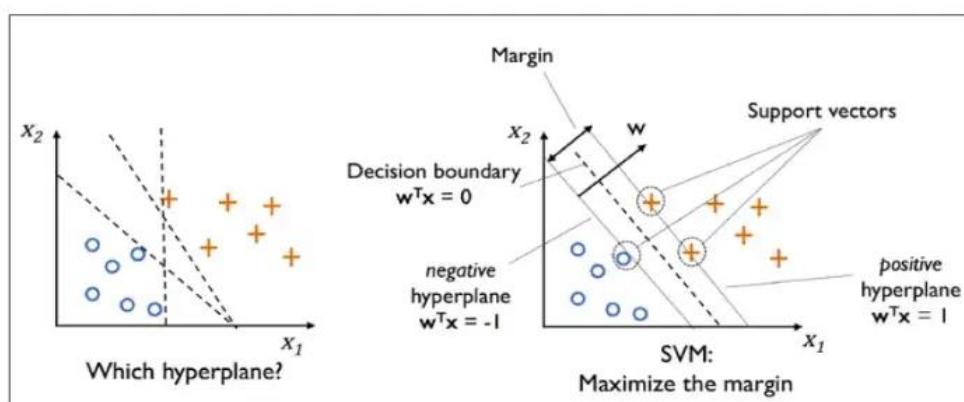
Regresi berganda merupakan hubungan antara 3 variabel atau lebih, yaitu sekurang-kurangnya 2 variabel dengan satu variabel bebas.

(Syahputra et al., 2020)

2.5 Support Vector Machine (SVM)

Support Vector Machine (SVM) merupakan algoritma pada *Supervised Learning* yang digunakan untuk mengatasi pemodelan pada kasus regresi dan klasifikasi. SVM merupakan sistem pembelajaran dengan menggunakan ruang hipotesis berupa fungsi - fungsi linier dalam sebuah fitur yang memiliki dimensi tinggi dan dilatih menggunakan algoritma pembelajaran yang didasarkan pada teori optimasi (Sugara & Subekti, 2019).

Penggunaan algoritma SVM bertujuan untuk mendapatkan nilai *hyperplane* terbaik dengan cara memaksimalkan jarak antar kelas. *Hyperplane* merupakan sebuah fungsi yang digunakan untuk memisahkan kelas. Pengklasifikasian kelas dalam fungsi 2-D disebut sebagai *line whereas*, pengklasifikasian kelas dalam fungsi 3-D disebut sebagai *plane similarity*, sedangkan pengklasifikasian pada ruang kelas lebih tinggi disebut sebagai *hyperplane*.



Gambar 2. 2 Ilustrasi SVM

Sumber Gambar : Penjelasan Sederhana tentang Apa Itu SVM | by Samsudiney | Medium

Hyperplane yang didapatkan pada *support vector machine* (SVM) diilustrasikan seperti pada *gambar 2.2* di atas. Posisi dari *hyperplane* berada diantara dua kelas, yang artinya jarak antara *hyperplane* dengan objek data berbeda dengan kelas yang berdekatan (ditandai dengan bulat positif dan negatif). Objek data terluar yang paling dekat dengan *hyperplane* pada SVM disebut dengan *support vector*. *Support vector* ini sulit untuk diklasifikasikan karena posisinya yang hampir tumpang tindih (*overlap*) dengan kelas lainnya. Sifat *support vector* ini termasuk kritis, oleh karena itu hanya objek ini yang dapat diperhitungkan untuk menemukan *hyperplane* yang optimal pada SVM (Samsudiney, 2019).

2.6 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 pinalti 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:

$$f(x) = w \cdot x + b \quad (2.6.1)$$

dengan :

- w = vektor pembobot
- x = vektor input berupa data
- b = nilai bias

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

$$f(x) = w\varphi(x) + b \quad (2.6.2)$$

dengan :

- $\varphi(x)$ = fungsi yang memetakan x pada sebuah dimensi
- w = vektor bobot
- b = bias

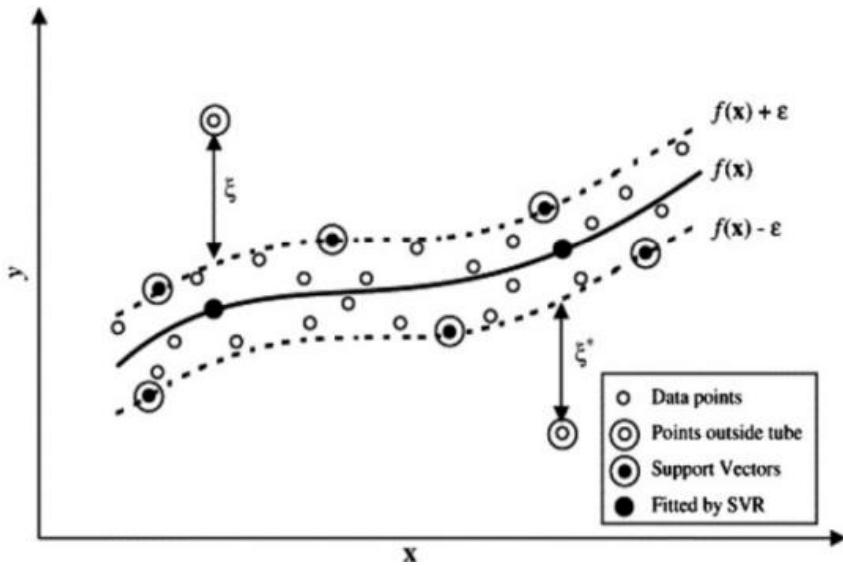
$$\min = \frac{1}{2} \| w \|^2 \quad (2.6.3)$$

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 $\varepsilon - \text{insensitive loss}$ agar dapat melakukan generalisasi dengan baik dengan batasan yang harus dipenuhi adalah titik harus berada pada rentang ε seperti pada persamaan berikut:

$$y_i - wx_i - b \leq \varepsilon \quad (2.6.4)$$

$$wx_i + b - y_i \leq \varepsilon, \quad \text{dimana } i=1,2,\dots,\lambda \quad (2.6.5)$$

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 ξ_i, ξ_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:

$$\min \frac{1}{2} \| \mathbf{w} \|^2 + C \sum_{i=1}^N (\xi_i + \xi_i^*), \quad C > 0 \quad (2.6.6)$$

dengan :

C = nilai penalti

N = jumlah seluruh data

ξ_i = titik di luar margin yang berada di atas hyperplane

ξ_i^* = titik di luar margin yang berada berikut hyperplane

dengan ketentuan:

$$y_i - w\mathbf{x}_i - b \leq \varepsilon + \xi_i \quad (2.6.7)$$

$$w\mathbf{x}_i + b - y_i \leq \varepsilon + \xi_i^* \quad (2.6.8)$$

$$\xi_i, \xi_i^* \geq 0, \text{ dimana } i=1,2,\dots,N \quad (2.6.9)$$

Sehingga diperoleh fungsi linear SVR seperti berikut:

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot \mathbf{x}_i \cdot \mathbf{x} + b \quad (2.6.10)$$

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

$$y = \sum_{i=1}^N (\alpha_i - \alpha_i^*) \cdot k(\mathbf{x}_i, \mathbf{x}) + b \quad (2.6.11)$$

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

$$k(x, x') = x^T y + c \quad (2.6.12)$$

2. Kernel Polynomial

$$k(x, x') = (\alpha x^T y + c)^d \quad (2.6.13)$$

3. Kernel Sigmoid

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

$$k(x, y) = \tanh(\alpha x \cdot y + c) \quad (2.6.14)$$

4. Kernel Radial Basis Function (RBF)

Kernel ini dapat mengatasi permasalahan yang bersifat non-linear.

$$k(x, x') = \exp(-\gamma \|x - y\|^2) \quad (2.6.15)$$

Penelitian ini menggunakan kernel RBF dikarenakan memiliki tingkat error yang paling kecil dibandingkan dengan kernel lain.

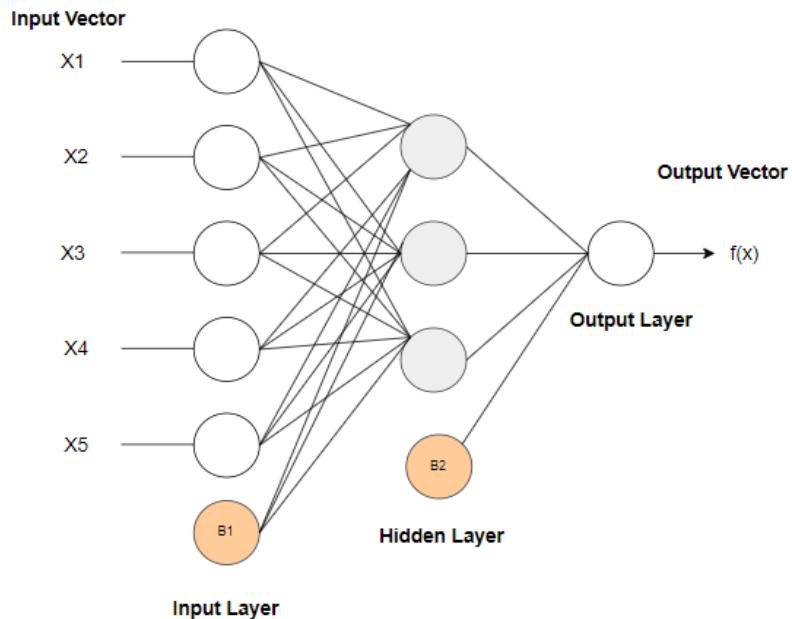
2.7 Grid Search

Metode grid search digunakan untuk mendapatkan model SVR dengan parameter Cost, epsilon, dan gamma yang paling optimal. Algoritma 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 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

Sumber : (Putra & Ulfa Walmi, 2020)

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).

2.9 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 disebarluaskan 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*. Algoritma *Backpropagation* ini digunakan fungsi aktivasi *sigmoid biner*, karena *output* yang diharapkan bernilai antara 0 sampai 1 (Situmorang, 2022).

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

Fase 0 :

Langkah 1 : Melakukan inisialisasi bobot dan bias dengan menggunakan nilai bilangan kecil yang dipilih secara *random*.

Langkah 2 : Jika kondisi berhenti tidak dipenuhi, maka kerjakan langkah 3 dan seterusnya, yang mana dikatakan berhenti jika nilai *error* “ \leq ” dari nilai target *error* (*threshold*) atau jika jumlah *epoch* “ \geq ” 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 :

$$[Z_{in1} \dots Z_{inp}] = [X_i \dots X_p] \cdot \begin{bmatrix} W_{11} & \dots & W_{1q} \\ \dots & \dots & \dots \\ W_{p1} & \dots & W_{pq} \end{bmatrix} + [b_{XZ_1} \dots b_{XZ_q}] \quad (2.7.1)$$

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 :

$$\partial[Z_{out1} \dots Z_{outq}] = f([Z_{in1} \dots Z_{inq}]) = \left[\frac{1}{1 + e^{-z_{in1}}} \dots \frac{1}{1 + e^{-z_{inq}}} \right] \quad (2.7.2)$$

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 :

$$[Y_{in1} \dots Y_{inr}] = [Z_{out1} \dots Z_{outq}] \cdot \begin{bmatrix} W_{11} & \dots & W_{1r} \\ \dots & \dots & \dots \\ W_{q1} & \dots & W_{qr} \end{bmatrix} + [b_{ZY_1} \dots b_{ZY_r}] \quad (2.7.3)$$

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

$$\partial[Y_{out1} \dots Y_{outr}] = f([Y_{in1} \dots Y_{inr}]) = \left[\frac{1}{1 + e^{-Y_{in1}}} \dots \frac{1}{1 + e^{-Y_{inr}}} \right] \quad (2.7.4)$$

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....

Fase 2: Menghitung Error

Langkah 6 : Menghitung nilai error dengan menggunakan persamaan *mean square error*.

Persamaan *Mean Square Error* (MSE) diberikan seperti berikut ini :

$$\text{loss} = \frac{1}{2} (\text{aktual} - Y_{outr})^2 \quad (2.7.5)$$

dengan :

aktual = nilai aktual

Y_{outr} = nilai prediksi

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

(Situmorang, 2022)

Fase 3: Backpropagation

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 dengan *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 :

$$\delta_r = (\text{aktual} - Y_{outr}).f'([Y_{in1} \dots Y_{inr}]) \quad (2.7.6)$$

dengan :

aktual	= nilai aktual
Y_{outr}	= nilai prediksi
$f'([Y_{in1} \dots Y_{inr}])$	= turunan aktivasi keluaran untuk unit Y_r
$aktual - Y_{outr}$	= 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:

$$\Delta W_{qr} = \eta \delta_r \partial [Z_{out1} \dots Z_{outq}] \quad (2.7.7)$$

dengan :

ΔW_{qr}	= menghitung korelasi bobot
η	= <i>learning rate</i>
$\delta_r \partial [Z_{out1} \dots Z_{outq}]$	= <i>local gradient</i> dikalikan dengan nilai dari unit pada <i>hidden layer</i> .

Untuk menghitung nilai dari delta bias ditunjukkan dengan persamaan :

$$\Delta b_r = \eta \delta_r \quad (2.7.8)$$

dengan :

Δb_r	= menghitung delta bias
$\eta \delta_r$	= <i>learning rate</i> dengan <i>local gradient</i> pada <i>output</i>

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 :

$$\delta_q = f'([Z_{in1} \dots Z_{inq}]) \cdot \sum_{q=1}^r w_{qr} \cdot \delta_r \quad (2.7.9)$$

dengan :

W_{qr}	= bobot <i>neuron hidden</i> ke <i>neuron output</i>
δ_r	= menghitung <i>local gradient</i> pada <i>output layer</i>

$f'([Z_{in1} \dots Z_{inq}])$ = 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 :

$$\Delta W_{pq} = \eta \delta_q [X_i \dots X_p] \quad (2.7.7)$$

dengan :

ΔW_{pq} = delta bobot pada *hidden layer*

η = *learning rate*

$\delta_q [X_i \dots X_p]$ = *local gradient* dikalikan dengan *input data*

Untuk menghitung nilai dari delta bias ditunjukkan dengan persamaan :

$$\Delta b_p = \eta \delta_q \quad (2.7.8)$$

dengan :

Δb_p = menghitung delta bias

η = *learning rate*

δ_q = *local gradient* pada *hidden layer*

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:

$$\Delta W_{qr \text{ baru}} = W_{qr \text{ lama}} + \Delta W_{qr} \quad (2.7.9)$$

dengan :

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

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

ΔW_{qr} = delta bobot pada *output layer*

Untuk menghitung perubahan nilai bias ditunjukkan dengan persamaan :

$$\Delta b_{r \text{ baru}} = b_{r \text{ lama}} + \Delta b_r \quad (2.7.10)$$

dengan :

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

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

Δb_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:

$$\Delta W_{pq \text{ baru}} = W_{pq \text{ lama}} + \Delta W_{pq} \quad (2.7.11)$$

dengan :

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

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

ΔW_{pq} = delta bobot pada *hidden layer*

Untuk menghitung perubahan nilai bias ditunjukkan dengan persamaan :

$$\Delta b_p \text{ baru} = b_p \text{ lama} + \Delta b_p \quad (2.7.12)$$

dengan :

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

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

Δb_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.

(Nainggolan & Lumbanraja, 2018)

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)

2.10 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 pemilahan 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. Data akan dinormalisasikan, dimana normalisasi ini adalah

proses memberikan rentang nilai data tertentu untuk memastikan bahwa data tidak ada yang berlebihan. Tahapan ini perlu dilakukan dikarenakan banyak sumber data yang memiliki format yang berbeda.

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

2.11 Evaluasi Hasil

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.

2.11.1 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.

$$MSE = \frac{1}{n} \sum_{t=1}^n (aktual - peramalan)^2$$

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.

2.11.2 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.

$$RMSE = \sqrt{MSE}$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{t=1}^n (aktual - peramalan)^2}$$

Semakin rendah RMSE, semakin baik model dan peramalannya. RMSE yang lebih tinggi menunjukkan bahwa ada penyimpangan yang besar dari residual ke kebenaran dasar.

2.12 Normalisasi

Normalisasi data merupakan bagian dari data transformasi yang digunakan untuk mengubah data menjadi nilai yang mudah dipahami. Adapun tujuan dari normalisasi ini dilakukan yaitu mendapatkan bobot yang sama dari semua atribut data dan tidak bervariasi atau hasil dari pembobotan tersebut tidak terdapat atribut yang lebih dominan atau dianggap lebih utama dari pada yang lain. Teknik ini dibutuhkan ketika data yang digunakan untuk prediksi memiliki rentang yang drastis yang mana data dengan ciri seperti ini dapat memberikan hasil yang kurang akurat (Trivusi, 2022).

Metode normalisasi yang digunakan dalam penelitian ini adalah *min-max normalization*. *min-max normalization* merupakan metode normalisasi dengan melakukan transformasi linier terhadap data asli sehingga menghasilkan keseimbangan nilai perbandingan antar data saat sebelum dan sesudah proses, adapun persamaannya adalah sebagai berikut:

$$\text{normalized}(x) = \frac{\text{minRange} + (x - \text{minValue})(\text{maxRange} - \text{minRange})}{\text{maxvalue} - \text{minValue}} \quad (2.10.1)$$

(Nasution et al., 2019)

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

3.1 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.

3.1.1 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

Variabel	Type data	Deskripsi
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

3.1.2 Seleksi Data

Dalam penelitian ini akan dilakukan pemilihan atribut yang akan digunakan dalam prediksi. Pemilihan atribut ini disesuaikan dengan bentuk data yang akan diolah dalam pembelajaran mesin dan tujuannya. Untuk penelitian ini, peneliti akan mengolah data ke bentuk data time series, sehingga peneliti hanya memerlukan jumlah pendaftar calon mahasiswa baru dari tahun 2016 sampai dengan 2022. Selain dari jumlah pendaftar, peneliti juga memerlukan data nama sekolah asal, dikarenakan tujuan penelitian ini adalah peneliti ingin melakukan prediksi jumlah pendaftar calon mahasiswa baru berdasarkan asal sekolahnya.

3.2 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.

3.2.1 Data Integration

Pada tahapan ini peneliti melakukan penggabungan data jumlah pendaftar yang diperoleh dari jalur seluruh jalur yang dibuka oleh institusi. Penggabungan data dilakukan peneliti dengan melakukan perhitungan jumlah peserta dari sekolah-sekolah SMA yang mendaftar sebagai peserta calon mahasiswa. Dari penggabungan data, peneliti menghasilkan data berupa jumlah peserta dari setiap sekolah yang mendaftar ke Institut Teknologi Del untuk setiap tahunnya dimulai dari tahun 2016-2022.

Berdasarkan proses integration yang telah dilakukan maka diperoleh dataset dengan tiga atribut sebagai berikut ini :

Tabel 3. 2 Atribut pada dataset

Nomor Atribut	Atribut	Tipe Data	Deksripsi
Atribut 1	Nama Sekolah	Character	Nama asal sekolah yang mendaftar
Atribut 2	Jumlah Pendaftar pada tahun sebelumnya	Numerik	Jumlah pendaftar pada tahun sebelumnya

Atribut 3	Prediksi Jumlah Pendaftar	Numerik	Prediksi jumlah pendaftar
-----------	---------------------------	---------	---------------------------

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

3.2.2 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

3.2.3 Pengubahan Variabel Teks ke Bentuk Numerik dengan Teknik 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.

3.2.4 Normalisasi Dataset

Setelah dilakukan encoding tahapan *preprocessing* selanjutnya adalah melakukan normalisasi data agar nilai data memiliki range yang sama sehingga *output* yang didapatkan model dapat mengklasifikasikan data secara tepat (**Wisnu, n.d.**).

Berikut adalah data yang akan dilakukan normalisasi.

Tabel 3. 6 Sampel Dataset sebelum normalisasi

Nama Sekolah	Jumlah Pendaftar 2016	Jumlah Pendaftar 2017	Jumlah Pendaftar 2018	Jumlah Pendaftar 2019	Jumlah Pendaftar 2020	Jumlah Pendaftar 2021	Jumlah Pendaftar 2022
SMA Kampus FKIP HKBP Nomensen	18	15	6	5	12	16	1
SMA Kristen Kalam Kudus Pematang Siantar	16	40	39	19	17	22	11
SMA Budi Mulia Pematang Siantar	32	38	18	23	28	21	7
SMA Yadika 6	0	0	0	2	4	0	0
SMA Bintang Timur Pematang Siantar	28	24	63	38	45	44	32

Dari sampel dataset di atas terlihat bahwa nilai dari setiap data memiliki rentang nilai yang tidak jelas, sehingga sebelum melakukan pemodelan perlu dilakukan normalisasi terlebih dahulu.

Perhitungan dilakukan dengan tahapan berikut.

- (1) Menentukan nilai max dan min dari setiap kolom.

Jumlah Pendaftar 2016 : max =32, min=0

Jumlah Pendaftar 2017 : max =40, min=0

Jumlah Pendaftar 2018 : max = 63, min=0

Jumlah Pendaftar 2019 : max = 38, min = 2

Jumlah Pendaftar 2020 : max = 45, min = 4

Jumlah Pendaftar 2021 : max = 44, min = 0

Jumlah Pendaftar 2022 : max = 32, min = 0

- (2) Melakukan perhitungan setiap kolom dengan menggunakan persamaan *min-max normalization* pada persamaan 2.10.1.

$$\chi^* = \frac{\chi_i - \min(\chi_i)}{\max(\chi_i) - \min(\chi_i)}$$

$$\chi^* = \frac{18 - 0}{32 - 0}$$

$$\chi^* = \frac{18}{32}$$

$$\chi^* = 0,56$$

Dari hasil perhitungan normalisasi yang dilakukan maka akan didapatkan nilai seperti pada tabel berikut ini.

Tabel 3. 7 Sampel Dataset setelah normalisasi

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 Kampus FKIP HKBP Nomensen	0,29	0,00	0,00	0,08	0,20	0,18	0,00
SMA	0,19	1,00	0,58	0,47	0,32	0,35	0,32

Kristen Kalam Kudus Pematang Siantar							
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

3.3 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.

Adapun langkah - langkah proses Analisis Algoritma SVR adalah sebagai berikut:

1. Inisialisasi parameter SVR

Parameter SVR terdiri dari parameter ϵ (nilai epsilon untuk zona yang masih dapat diterima), C (cost), γ untuk masukan fungsi kernel yang digunakan dan jumlah iterasi maksimum. Nilai parameter ini harus diatur untuk mendapatkan nilai parameter yang optimal.

2. Melakukan proses normalisasi data, adapun metode yang digunakan yaitu min-max normalization yang bertujuan untuk menstandarisasi semua data yang digunakan dalam perhitungan sehingga berada pada jarak tertentu.

3. Menentukan range nilai parameter c , ϵ , γ .

4. Melakukan pemodelan dengan algoritma SVR

3.3.1 Perhitungan SVR

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

1. Menginisialisasi awal untuk nilai α , C, epsilon, gamma, lambda:

Tabel 3. 8 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 telah dilakukan normalisasi dapat dilihat pada tabel berikut:

Tabel 3. 9 Data Uji yang sudah dinormalisasikan

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 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. 10 Data Uji

	X1	X2	X3	X4	X5	X6	Y
	A1	A2	A3	A4	A5	A6	Y
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. 11 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:

$$k(x, x') = \exp(-\gamma ||x - y||^2)$$

Maka untuk perhitungan dengan kernel rbf yaitu:

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

$$\mathbf{K(A1,A2)} = \exp(-0,167\|0,29-0\|^2) + \exp(-0,167\|0,19-1\|^2) + \exp(-0,167\|1,00-0,92\|^2) = \\ 3,881 \\ 139032$$

maka untuk hasil perhitungan dengan menggunakan kernel rbf yaitu:

Tabel 3. 12 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
A3		2,855984901	2,885417662	3	2,96829960 9	2,95227962

4. Menghitung matriks dengan rumus:

$$D_{ij} = y_i y_j (K(x_i x_j) + \lambda^2)$$

Dimana:

D_{ij} = elemen matriks ke-ij

y_i = kelas data ke-i

y_j = kelas data ke-j

λ = 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. 13 Hasil Perhitungan Matriks Hessian

	A1	A2	A3	A4	A5	A6
A1	0	0	0	0	0	0

A2	0,4230286645	0,4352	0,4234667686	0,4284331323	0,4250561511	0,421689620
A3	0,1482260549	0,1492885776	0,153425	0,1522806159	0,1517022943	0,152622829

5. Mencari nilai error menggunakan rumus:

$$E_i = \sum_{j=i}^i \alpha_j D_{ij}$$

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. 14 Hasil Perhitungan Nilai Error

	Error
A1	0
A1	1,278437168
A3	0,4537726859

6. Menghitung nilai delta alpha

$$\delta\alpha_i = \min\{\max[\gamma(1 - E_i), -\alpha_i], C - \alpha_i\}$$

$$\text{Min}\{\max[0,167(1 - 0), -0,5], 1 - 0,5\}$$

$$\text{Min}\{\max(0,167, -0,5), 0,5\}$$

$$\text{Min}\{0,167, 0,5\}$$

$$= 0,167$$

Hasil perhitungan delta alpha:

Tabel 3. 15 Hasil Perhitungan Delta Alpha

	Delta Alpha
A1	0,167
A2	-0,04649900706
A3	0,09121996145
A4	0,1195388175

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

7. Menghitung nilai α baru dengan menggunakan rumus:

$$\alpha_i = \alpha_i + \delta_i \alpha$$

Contoh perhitungan:

$$\begin{aligned}\alpha_1 &= 0,5 + 0,167 = 0,667 \\ \alpha_2 &= 0,5 + (-0,4649900706) = 0,035\end{aligned}$$

Tabel 3. 16 Hasil Perhitungan Nilai Alpha Baru

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

8. Melakukan pencarian nilai w dengan rumus:

$$w = \sum_{i=l}^l (\alpha_i - \alpha_i^*)$$

Contoh perhitungan:

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

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

Maka hasil perhitungan w adalah sebagai berikut:

Tabel 3. 17 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:

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

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. 18 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. 19 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.

3.4 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.

3.4.1 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. 20 Nilai bias dari input layer ke hidden layer

(bias, hidden layer)	(,1)	(,2)	(,3)
(1,)	0,48	0,50	0,65

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

(input, hidden layer)	(,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. 22 Nilai bias dari hidden layer ke output layer

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

Tabel 3. 23 Nilai bobot dari hidden layer ke output layer

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

3.4.2 Inisialisasi Hyperparameter

Setelah dilakukan inisialisasi parameter bobot dan bias, maka selanjutnya dilakukan inisialisasi hyperparameter seperti learning rate, epoch, dan target error. Nilai hyperparameter yang akan digunakan pada analisis ini adalah *learning rate*=0,5 , *epoch*=1, dan *target error* = 0,3.

3.4.3 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:

1. 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 2.7.1 .

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

$$[Z_{in1} \ Z_{in2} \ Z_{in3}] = [0.19 \ 1.00 \ 0.58 \ 0.47 \ 0.32 \ 0.35] \cdot \begin{bmatrix} 0.20 & 0.50 & 0.50 \\ 0.15 & 0.20 & 0.40 \\ 0.30 & 0.10 & 0.35 \\ 0.40 & 0.25 & 0.15 \\ 0.18 & 0.30 & 0.25 \\ 0.22 & 0.45 & 0.30 \end{bmatrix}$$

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

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

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

Setelah dilakukan penghitungan sinyal-sinyal inputan terbobot, selanjutnya akan dilakukan penghitungan fungsi aktivasi dari sinyal inputan terbobot. Untuk melakukan perhitungan fungsi aktivasi digunakan persamaan 2.7.2 .

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

$$\partial([Z_{in1} \ Z_{in2} \ Z_{in3}]) = \left[\frac{1}{1 + e^{-(1.165)}} \quad \frac{1}{1 + e^{-(1.224)}} \quad \frac{1}{1 + e^{-(1.604)}} \right]$$

$$\partial([Z_{in1} \ Z_{in2} \ Z_{in3}]) = [0.762 \ 0.773 \ 0.832]$$

$$[Z_{in1} \ Z_{in2} \ Z_{in3}] = [0.762 \ 0.773 \ 0.832]$$

2. 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 dikirimkan ke *output layer*. Persamaan yang digunakan untuk perhitungan ini adalah persamaan 2.7.3 .

$$[Y_{in}] = [Z_{in1} \ Z_{in2} \ Z_{in3}] \cdot \begin{bmatrix} W_{11} \\ W_{21} \\ W_{31} \end{bmatrix} + [b_{21}]$$

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

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

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

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 2.7.4 seperti berikut :

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

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

3. Menghitung Nilai Error

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 2.7. 5 , nilai *loss* dapat ditentukan seperti berikut ini.

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

3.4.4 Backpropagation Propagation

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.

Setelah dilakukan prediksi perlu dilakukan denormalisasi kembali dengan tujuan mengembalikan nilai prediksi yang merupakan nilai normalisasi ke bentuk nilai prediksi yang merupakan nilai sebenarnya. Perhitungan denormalisasi dilakukan dengan menggunakan persamaan 2.11.1 berikut ini :

$$d = d'(max - min) + min$$

$$d = 0,838 (32 - 0) + 0$$

$$d = 26,816$$

$$d = 26$$

Dari hasil denormalisasi diatas, maka prediksi jumlah pendaftar calon mahasiswa baru tahun 2022 dari sekolah SMA Kalam Kudus Pematang Siantar adalah sebanyak 26 orang.

3.5 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. Semakin kecil nilai RMSE yang dihasilkan model, maka nilai prediksi semakin akurat (Andriyani, 2021). Evaluasi model RMSE memiliki beberapa kelebihan diantaranya adalah :

1. RMSE memberikan hasil dalam satuan yang sama dengan target, sehingga mudah dipahami dan diinterpretasikan

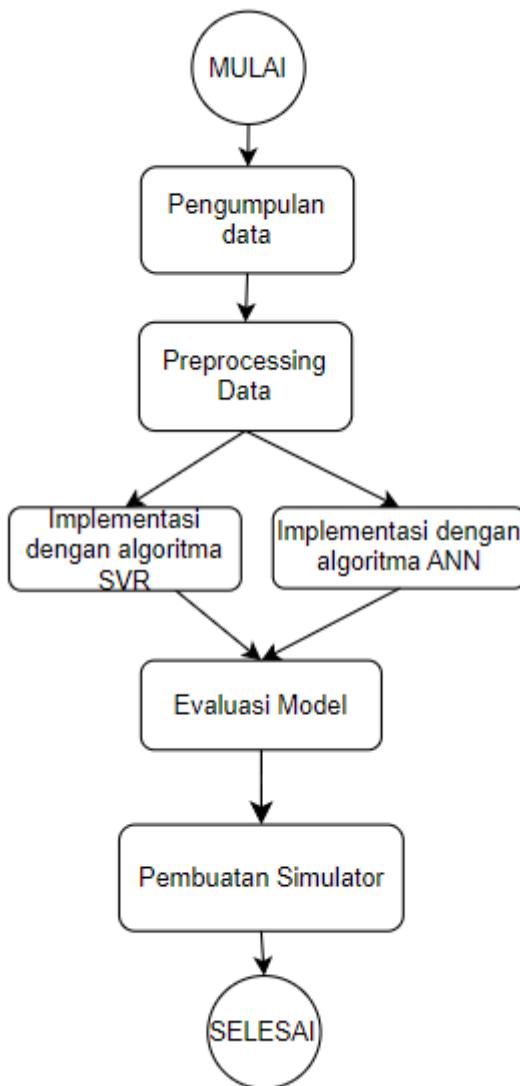
2. RMSE memperhitungkan semua nilai residual (perbedaan nilai antara nilai aktual dan nilai prediksi), termasuk outlier, sehingga lebih sensitif terhadap perbedaan besar antara nilai aktual dan nilai prediksi.
3. RMSE menunjukkan performa model secara keseluruhan, bukan hanya performa pada titik tertentu.

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 akan 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 pendaftarnya. 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.

3. Implementasi Algoritma

Setelah data yang akan diolah telah disiapkan, maka selanjutnya data tersebut akan digunakan pada tahapan implementasi algoritma. Pada implementasi algoritma ada 2 algoritma yang akan digunakan yaitu algoritma SVR dan algoritma ANN.

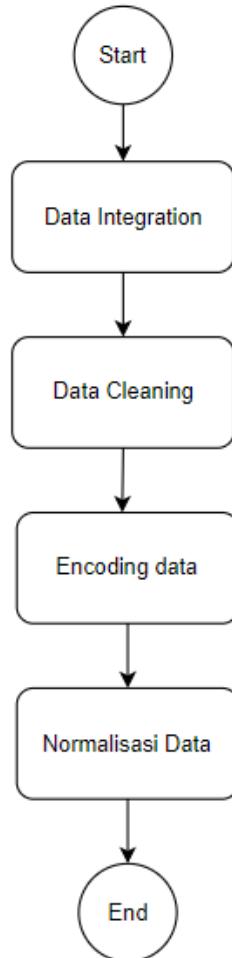
4. Evaluasi Model

Setelah melakukan implementasi algoritma maka akan dilakukan evaluasi model dari kedua algoritma yang telah dibuat. Dimana pada tahapan ini akan mencari nilai akurasi dari hasil setiap pemodelan data yang telah dilakukan. Kemudian akan dilakukan pemilihan algoritma terbaik berdasarkan nilai evaluasi model terbaik.

5. Pembuatan Simulator

Pada tahapan ini akan dilakukan pembangunan aplikasi sederhana sebagai simulator untuk menampilkan hasil prediksi jumlah pendaftar calon mahasiswa baru. Pada aplikasi simulator ini algoritma yang akan digunakan adalah algoritma terbaik yang dilihat berdasarkan hasil evaluasi model.

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 peneliti melakukan penggabungan data pendaftar yang diperoleh dari seluruh jalur pendaftaran untuk setiap tahunnya. Sehingga hasil akhir dari tahapan ini adalah dataset dengan bentuk time series yang terdiri dari variabel nama sekolah, jumlah pendaftar tahun 2016 sampai jumlah pendaftar tahun 2022.

2. Data Cleaning

Pada tahapan data cleaning ini, peneliti melakukan pembersihan data dengan melakukan *missing value*. *Missing value* dilakukan dikarenakan adanya kekosongan jumlah pendaftar dari salah satu sekolah untuk tahun tertentu.

3. Encoding Data

Pada tahapan encoding data ini, peneliti melakukan pengubahan bentuk data nama sekolah dari bentuk teks ke bentuk numerik dengan tujuan agar nama sekolah dapat digunakan dalam pemodelan.

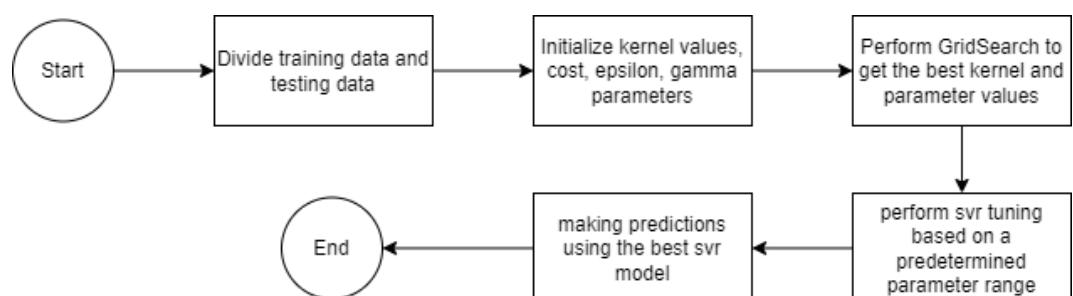
4. Normalisasi Data

Pada tahapan ini akan dilakukan penyamaan skala nilai data ke dalam rentang 0-1.

4.3 Rancangan Implementasi Algoritma

Berikut ini adalah rancangan implementasi algoritma yang digunakan dalam penelitian.

4.3.1 Implementasi Algoritma SVR



Gambar 4. 3 Flow algoritma *Support Vector Regression*

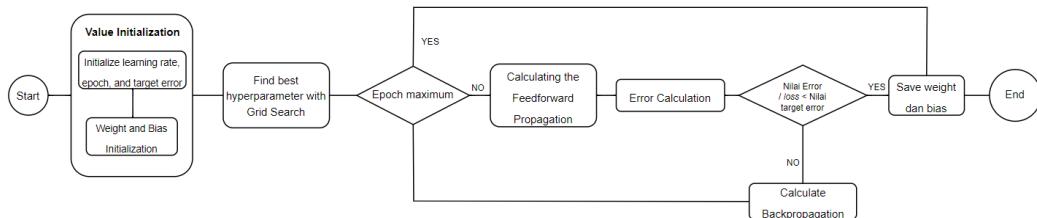
Berdasarkan Gambar di atas, langkah pelaksanaan algoritma Support Vector Regression dilakukan dengan beberapa tahapan dengan rincian dari masing – masing tahapan adalah sebagai berikut:

1. Melakukan inisialisasi parameter.

Parameter SVR terdiri dari beberapa parameter ϵ (nilai epsilon untuk zona yang masih dapat diterima), C (cost), γ untuk masukan fungsi kernel yang digunakan dan jumlah iterasi maksimum. Nilai parameter harus diatur agar mendapatkan nilai parameter yang optimal.

2. Melakukan proses normalisasi data menggunakan metode min-max normalization yang berfungsi untuk melakukan standarisasi semua data yang digunakan dalam perhitungan sehingga berada pada jarak tertentu.
3. Menentukan range nilai parameter c , ϵ , γ
4. Melakukan optimasi kernel dan parameter terbaik menggunakan Grid Search
5. Melakukan pemodelan dengan algoritma SVR, melakukan prediksi serta menghitung RMSE yang didapatkan.

4.3.2 Implementasi Algoritma ANN Backpropagation



Gambar 4. 4 Flow algoritma ANN *Backpropagation*

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

1. Inisialisasi Nilai

Pada tahapan ini, perlu dilakukan inisialisasi nilai parameter yang diperlukan untuk pemodelan. Nilai parameter yang perlu diinisialisasikan adalah nilai bobot, nilai bias, nilai *learning rate*, dan nilai target *error*.

2. Cek Kondisi *Epoch* atau Iterasi

Pada tahapan ini dilakukan pengecekan iterasi yang telah dilakukan, ketika iterasi yang dilakukan lebih besar sama dengan (\geq) nilai *epoch* yang telah ditentukan, maka nilai bobot dan bias tersebut akan disimpan. Nilai bobot dan bias ini yang akan digunakan pada algoritma. Jika nilainya lebih kecil, maka akan dilanjutkan pada perhitungan feedforward.

3. Menghitung Feedforward Propagation

Pada tahapan ini akan dilakukan pemodelan dengan feedforward propagation terlebih dahulu dengan menghitung sinyal-sinyal yang dihasilkan oleh setiap input mulai dari sinyal ke *hidden layer* hingga sinyal *output layer*.

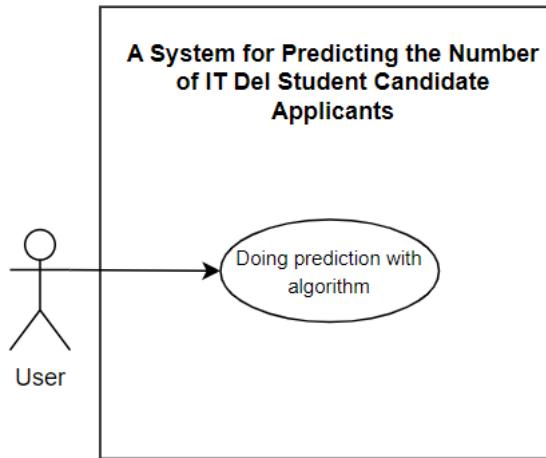
4. Menghitung Error

Pada tahapan ini akan dilakukan penghitungan *error* dari hasil pemodelan data training. Penghitungan *error* dilakukan dengan model MSE yang mana akan dilihat nilai error hasil prediksi terhadap hasil aktual. Berdasarkan hasil MSE ini akan ditentukan apakah perlu dilakukan perubahan bobot dan bias atau tidak. Penghitungan bobot dan bias baru dapat ditentukan dengan memperhatikan nilai error lebih besar dari nilai aktual atau tidak, jika lebih besar maka perlu dilakukan perhitungan dan jika tidak berarti nilai bobot dan bias sudah tidak perlu diperbarui.

5. Menghitung Backpropagation

Pada tahapan ini, akan dilakukan perubahan nilai bobot dan bias bergantung pada kondisi dari nilai error yang diperoleh dari tahapan sebelumnya. Jika nilai *error (loss)* pada MSE lebih kecil ($<$) dari nilai target *error*, maka tidak perlu dilakukan pembaharuan bobot bias. Namun jika sebaliknya, maka perlu dilakukan pembaharuan bobot bias pada tahapan ini.

4.4 Use Case Diagram



Gambar 4. 5 Rancangan use case aplikasi web

Gambar usecase di atas merupakan gambaran aktivitas yang dilakukan oleh user dalam aplikasi web yang menjadi simulator dari prediksi jumlah pendaftar dengan menggunakan pemodelan dengan akurasi terbaik dari kedua algoritma yang telah diimplementasikan. Pada *use case* tersebut digambarkan bahwa user akan mengisi inputan variabel pada halaman UI kemudian sistem akan melakukan prediksi dengan algoritma yang telah dipilih setelah dilakukan evaluasi model. Kemudian sistem juga akan menampilkan hasil prediksi yang telah dilakukan sistem pada halaman UI user.

4.5 Desain Prototype

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

Halaman Home

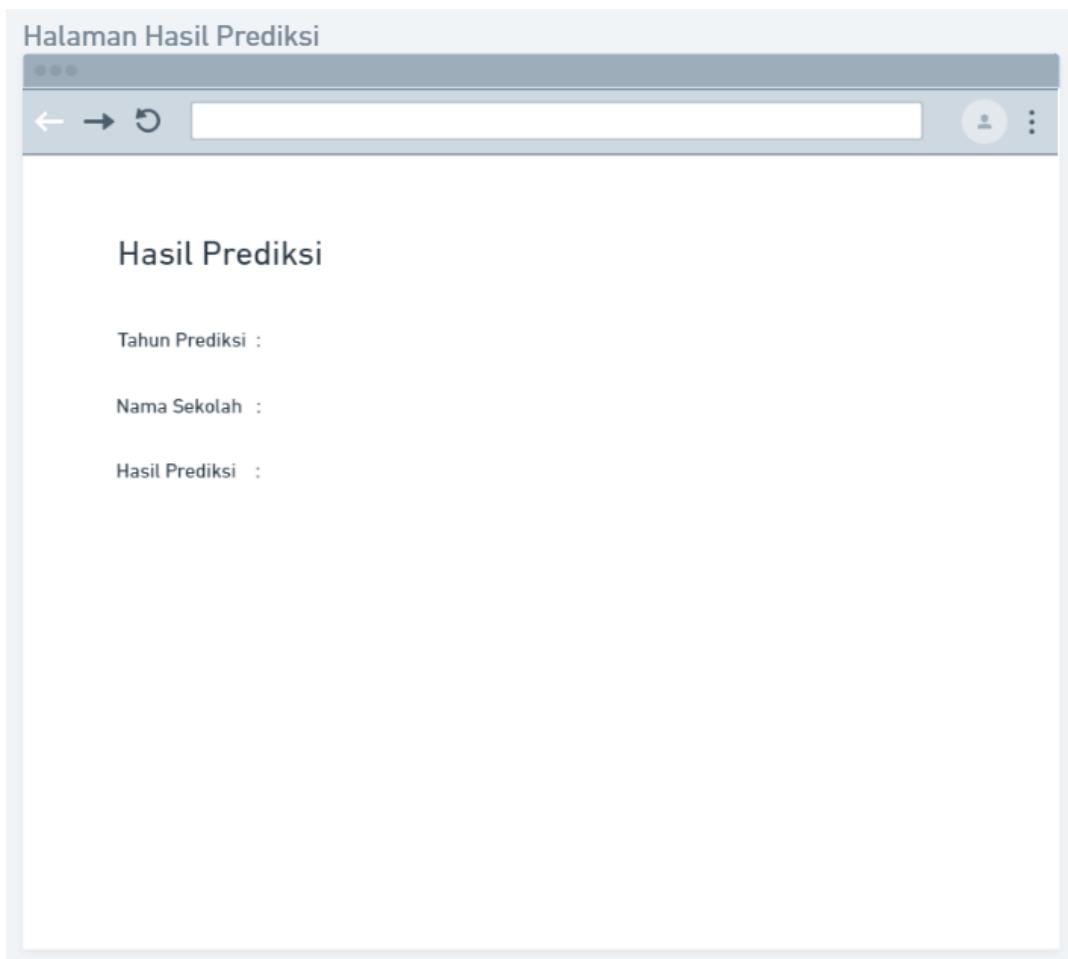
Prediksi Pendaftar Calon Mahasiswa Baru
IT DEL

Tahun Prediksi

Nama Sekolah

Lakukan Prediksi

Gambar 4. 6 Rancangan halaman prediksi



Gambar 4. 7 Rancangan halaman hasil prediksi

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 :

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 :

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

5.3 Implementasi Pengumpulan Data

Seperti yang telah disampaikan pada **sub-bab 3.1.1** , pengumpulan data dilakukan dengan meminta data peserta yang mendaftar sebagai calon mahasiswa IT Del dari BAAK IT Del. Hasil dari pengumpulan data yang diterima adalah data informasi peserta yang diperoleh mulai tahun 2016 sampai tahun 2022.

Data yang diperoleh masih dalam bentuk data mentah yang belum dapat digunakan oleh algoritma untuk melakukan prediksi. Agar data dapat digunakan algoritma untuk prediksi, maka perlu dilakukan preprocessing data terlebih dahulu.

5.4 Implementasi Preprocessing

Tahapan preprocessing data dilakukan dengan beberapa tahapan seperti yang dijelaskan pada **sub-bab 3.2**. Pada tahapan preprocessing, beberapa tahapan tidak dilakukan dengan menggunakan implementasi kode. Tahapan preprocessing data

yang menggunakan implementasi kode adalah tahapan encoding dan tahapan normalisasi data.

5.4.1. Data Integration

Pada tahapan data integration ini data peserta pendaftar calon mahasiswa IT Del dari seluruh jalur penerimaan institusi setiap tahunnya digabungkan dan dikelompokkan berdasarkan asal sekolah sehingga hasil yang diperoleh membentuk data time series. Dari hasil penggabungan data maka akan diperoleh data yang berisi variabel nama sekolah, jumlah pendaftar 2016, jumlah pendaftar 2017, jumlah pendaftar 2018, jumlah pendaftar 2019, jumlah pendaftar 2020, jumlah pendaftar 2021, dan jumlah pendaftar 2022 (seperti pada gambar 5.1).

LIST PENDAFTARAN USM 3 201				LIST PENDAFTARAN USM 1 2016 Peserta USM 1 Del (manual)				
NO.	NOMOR PENDAFTARAN	NAMA SISWA	JENIS KELAMIN	ASAL SEKOLAH	NO.	NOMOR PENDAFTARAN	NAMA SISWA	ASAL SEKOLAH
1	USM316010001	Irinus Agnes Larasati	P	SMA Negeri 2 Mandau	1	USM1.16-01001	Firda Siranga	SMA Negeri 1 Siantar Narumonda
2	USM316010002	Gabriel Argado Hutapea	L	SMK Swasta St. Nahanson	2	USM1.16-01002	Gely Ayunda	SMA Negeri 1 Siantar Narumonda
3	USM316010003	Elinfrida E. Marpaung	P		3	USM1.16-01003	Roy Christian N.T. Sitorus	SMA Negeri 1 Siantar Narumonda
4	USM316010004	Cindy C. Silaken	P	SMA Negeri 1 Aek Songgonan	4	USM1.16-01004	Herewanti Iedha Rajengukuk	SMA Negeri 1 Siantar Narumonda
5	USM316010005	Berkat Leisa	L	SMA Negeri 1 Lumbanjuh	5	USM1.16-01005	Santius Moutaha Mamurang	SMA Negeri 1 Siantar Narumonda
6	USM316010006	Lestari R. Sitorus	P	SMK Negeri 1 Balige	6	USM1.16-01006	Irawati Simaga	SMA Negeri 1 Siantar Narumonda
7	USM316010007	Juliana Tambunan	P	SMA Bintang Timur Balige	7	USM1.16-01007	Ivan Fransiskus Simatupang	SMA Negeri 1 Siantar Narumonda
8	USM316010008	Willy V. Sadaan	L	SMK Yayanra Soposurung	8	USM1.16-01008	Xenia Caterine Subahan	SMA Negeri 1 Siantar Narumonda
9	USM316010009	Pratini N. S. Sitompul	P	SMA Negeri 1 Kolang	9	USM1.16-01009	Maria Y. Lumban Toruan	SMA Negeri 1 Siantar Narumonda
10	USM316010010	Febri R. Simatupang	L	Taman Siswa	10	USM1.16-01010	Syaziah Artha Siagian	SMA Negeri 1 Balige
11	USM316010011	Restina W. Hutaileaun	P	SMK Swasta Asisi Pematangsiantar	11	USM1.16-01011	Cevin C.P. Napitupulu	SMA Negeri 1 Balige
12	USM316010012	Gina Naomi Tambunan	P	SMA Negeri 1 Balige	12	USM1.16-01012	Fameraida Y. A. Tambunan	SMA Negeri 1 Balige

Gambar 5. 1 Data mentah yang diperoleh dari BAAK

A	B	C	D	E	F	G	H
1 Nama Sekolah	2016	2017	2018	2019	2020	2021	2022
2 SMA ABDI SEJATI PERDAGANGAN	1	2	0	1	0	0	2
3 SMA ADVENT PEMATANG SIANTR	1	0	1	1	1	0	0
4 SMA ASISI PEMATANG SIANTR	1	0	0	3	5	1	3
5 SMA BINTANG TIMUR BALIGE	70	54	53	110	46	48	53
6 SMA BINTANG TIMUR PEMATANG SIANTR	30	19	6	7	18	8	2
7 SMA BRIGIJEN KATAMSO 1 MEDAN	1	1	4	1	0	0	0
8 SMA BUDI MULIA PEMATANG SIANTR	44	47	39	36	16	30	10
9 SMA BUDI MURNI 1 MEDAN	8	26	12	2	5	11	3
10 SMA BUDI MURNI 2 MEDAN	2	2	1	3	3	3	1
11 SMA CAHAYA MEDAN	14	11	6	1	2	1	0
12 SMA CINTA RAKYAT PEMATANG SIANTR	5	3	0	1	2	1	1
13 SMA FRANSISKUS SIBOLGA	2	1	9	1	1	0	0
14 SMA GLOBAL PERSADA MANDIRI	1	0	0	1	0	0	0
15 SMA KALAM KUDUS MEDAN	1	3	11	1	1	6	0
16 SMA KAMPUS FKIP HKBP NOMENSEN	1	1	1	0	0	0	0
17 SMA KARTIKA JAYA 1-4	1	1	0	0	0	0	0
18 SMA KATOLIK KABANJAHE	1	0	0	3	2	4	2
19 SMA KATOLIK SIBOLGA	10	10	2	4	17	4	4
20 SMA KATOLIK TRISAKTI MEDAN	9	7	0	3	4	4	0
21 SMA KRISTEN IMMANUEL	1	1	2	3	0	0	0
22 SMA KRISTEN KALAM KUDUS PEMATANG SIANTR	7	10	4	3	4	4	0
23 SMA MANDIRI TANGERANG	1	0	0	1	2	1	1

Gambar 5. 2 Data hasil integration

5.4.2. Data Cleaning

Pada tahapan ini, peneliti melakukan pengisian data yang mengalami *missing value* dengan memasukkan nilai default 0. Pelaksanaan data cleaning ini telah dijelaskan pada **sub-bab 3.2.2**.

5.4.3. Encoding

Encoding adalah proses pengubahan data yang non-numerik menjadi bentuk numerik, sehingga dapat diolah oleh model *machine learning*. Teknik encoding ini dilakukan pada tahapan implementasi ini agar model dapat ikut mengolah nilai nama sekolah yang bertipe non-numerik.

Library :

labelEncoder, merupakan library yang digunakan untuk melakukan encoding pada variabel ‘nama sekolah’

Input :

Dataset jumlah pendaftar mahasiswa yang telah disiapkan.

Output :

Dataset jumlah pendaftar mahasiswa dengan nilai encoding variabel ‘nama sekolah’

Tahapan :

1. Import library labelEncoder
2. Memuat dataset
3. Melakukan encoding pada variabel ‘nama sekolah’

5.4.2. Normalisasi

Normalisasi adalah suatu teknik dalam pengolahan data, dimana dilakukan pengubahan nilai pada variabel ke dalam rentang yang kecil dan sama. Tujuan dilakukan normalisasi adalah untuk mempermudah pemrosesan data dan meningkatkan performa dari model yang dibuat.

Library :

MinMaxScaler, library yang digunakan untuk melakukan normalisasi.

Input :

Dataset jumlah pendaftar mahasiswa yang telah disiapkan.

Output :

Dataset jumlah pendaftar yang sudah dilakukan normalisasi

Tahapan :

1. Import library yang akan digunakan untuk normalisasi
2. Memuat dataset yang akan digunakan
3. Melakukan pembagian data menjadi variabel inputan dan variabel target.
4. Melakukan normalisasi pada variabel inputan.

5.5 Implementasi Algoritma

5.5.1. Support Vector Regression

Pada bagian ini akan dilakukan pembuatan model dengan menggunakan algoritma Support Vector Regression yang akan digunakan untuk melakukan prediksi jumlah pendaftar calon mahasiswa IT Del untuk tahun depan. Pada pemodelan ini, data yang diolah adalah data yang sudah melewati tahapan *encoding* data. Berikut adalah langkah-langkah yang dilakukan.

Library :

Pandas : library untuk memuat dataset dan membuat dataframe.

Numpy : library yang digunakan untuk melakukan perhitungan aritmatika.

KFold : library yang digunakan dalam pembagian data.

Inputan :

Dataset yang telah dilakukan *encoding* data.

Output :

Model prediksi jumlah pendaftar menggunakan algoritma support vector regression

Tahapan :

1. Import library yang digunakan
2. Memuat dataset yang digunakan
3. Menentukan variabel target dan variabel inputan
4. Melakukan normalisasi
5. Mendefinisikan nilai-nilai parameter yang akan digunakan
6. Mendefinisikan nilai k-fold yang akan digunakan

7. Melakukan pembagian data sebanyak k yang didefinisikan sebelumnya
8. Melakukan pemodelan dengan algoritma SVR
9. Menghitung nilai rata-rata dari RMSE
10. Melakukan final train model pada seluruh data
11. Mendefinisikan data sekolah yang ingin diprediksi
12. Melakukan prediksi jumlah pendaftar calon mahasiswa dari sekolah yang telah didefinisikan.

5.5.2. Artificial Neural Network

Pada bagian ini akan dilakukan pembuatan model dengan menggunakan algoritma Artificial Neural Network yang akan digunakan untuk melakukan prediksi jumlah pendaftar calon mahasiswa IT Del untuk tahun depan. Pada pemodelan ini, data yang diolah adalah data yang sudah melewati tahapan *encoding* data. Berikut adalah langkah-langkah implementasi yang dilakukan.

Library :

Pandas : library untuk memuat dataset dan membuat dataframe.
 Numpy : library yang digunakan untuk melakukan perhitungan aritmatika.
 KFold : library yang digunakan dalam pembagian data.
 Tensorflow : library yang digunakan untuk membantu membuat model deep learning.
 Keras : library yang digunakan untuk membantu penerapan model deep learning.
 GridSearch: library yang digunakan untuk mencari nilai parameter terbaik yang menghasilkan performa model yang optimal.
 model_from_json : library yang digunakan untuk menyimpan model dengan library keras ke dalam bentuk .h5.

Inputan :

Dataset yang telah dilakukan *encoding* data.

Output :

Model prediksi jumlah pendaftar menggunakan algoritma artificial neural network.

Tahapan :

1. Import library yang digunakan
2. Memuat dataset yang digunakan
3. Menentukan variabel target dan variabel inputan

4. Melakukan normalisasi
5. Mendefinisikan hyperparameter yang akan digunakan
6. Mencari hyperparameter terbaik dengan menggunakan Grid Search
7. Memasukkan nilai hyperparameter yang telah ditemukan ke dalam model
8. Mendefinisikan nilai k-fold yang akan digunakan
9. Melakukan pembagian data sebanyak k yang didefinisikan sebelumnya
10. Melakukan pemodelan dengan algoritma ANN
11. Menghitung nilai rata-rata dari R\ MSE dan akurasi dari k-fold
12. Melakukan final train model pada seluruh data
13. Mendefinisikan sekolah yang ingin diprediksi
14. Melakukan prediksi jumlah pendaftar calon mahasiswa dari sekolah yang telah didefinisikan.
15. Menyimpan model ke dalam format .h5

BAB 6

HASIL DAN PEMBAHASAN

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

6.1. Pembahasan Preprocessing Data

Pada sub-bab ini, peneliti akan membahas dua tahapan preprocessing data yaitu tahapan encoding dan normalisasi.

6.1.1. Pembahasan Encoding

Encoding perlu dilakukan karena mesin hanya dapat memahami angka, sehingga mesin dapat mengolah data dan mempelajarinya menggunakan proses aritmatika. Berikut bentuk data sebelum dilakukan encoding.

index	Nama Sekolah	2016	2017	2018	2019	2020	2021	2022
0	SMAABDI SEJATI PERDAGGAN	1	2	0	1	0	0	2
1	SMAADVENT PEMATANG SIANTAR	1	0	1	1	1	0	0
2	SMAASISI PEMATANG SIANTAR	1	0	0	3	5	1	3
3	SMA BINTANG TIMUR BALIGE	70	54	53	110	46	48	53
4	SMA BINTANG TIMUR PEMATANG SIANTAR	30	19	6	7	18	8	2
5	SMA BRIGIJEN KATAMISO 1 MEDAN	1	1	4	1	0	0	0
6	SMA BUDI MULIA PEMATANG SIANTAR	44	47	39	36	16	30	10
7	SMA BUDI MURNI 1 MEDAN	8	26	12	2	5	11	3
8	SMA BUDI MURNI 2 MEDAN	2	2	1	3	3	3	1
9	SMA CAHAYA MEDAN	14	11	6	1	2	1	0
10	SMA CINTA RAKYAT PEMATANG SIANTAR	5	3	0	1	2	1	1
11	SMA FRANSISKUS SIBOLGA	2	1	9	1	1	0	0
12	SMA GLOBAL PERSADA MANDIRI	1	0	0	1	0	0	0
13	SMA KALAM KUDUS MEDAN	1	3	11	1	1	6	0
14	SMA KAMPUS FKIP HKBP NOMENSEN	1	1	1	0	0	0	0
15	SMA KARTIKA JAYA 1-4	1	1	0	0	0	0	0
16	SMA KATOLIK KABANAJAHE	1	0	0	3	2	4	2
17	SMA KATOLIK SIBOLGA	10	10	2	4	17	4	4
18	SMA KATOLIK TRISAKTI MEDAN	9	7	0	3	4	4	0
19	SMA KRISTEN IMMANUEL	1	1	2	3	0	0	0
20	SMA KRISTEN KALAM KUDUS PEMATANG SIANTAR	7	10	4	3	4	4	0
21	SMA MANDIRI TANGERANG	1	0	0	1	2	1	1
22	SMA MARKUS MEDAN	3	2	1	4	2	6	3
23	SMA MARS PEMATANG SIANTAR	3	0	0	0	0	0	0
24	SMA METHODIST 1 MEDAN	3	11	6	1	0	0	0

Gambar 6. 1 Data sebelum encoding

Dari bentuk data diatas dilihat bahwa variabel ‘nama sekolah’ masih berbentuk karakter /teks, sehingga variabel tersebut tidak dapat digunakan dalam pemodelan algoritma. Agar variabel ‘nama sekolah’ dapat digunakan dalam pemodelan maka perlu dilakukan encoding, sehingga dari tahapan encoding akan menghasilkan perubahan tipe data variabel ‘nama sekolah’ yaitu menjadi seperti berikut.

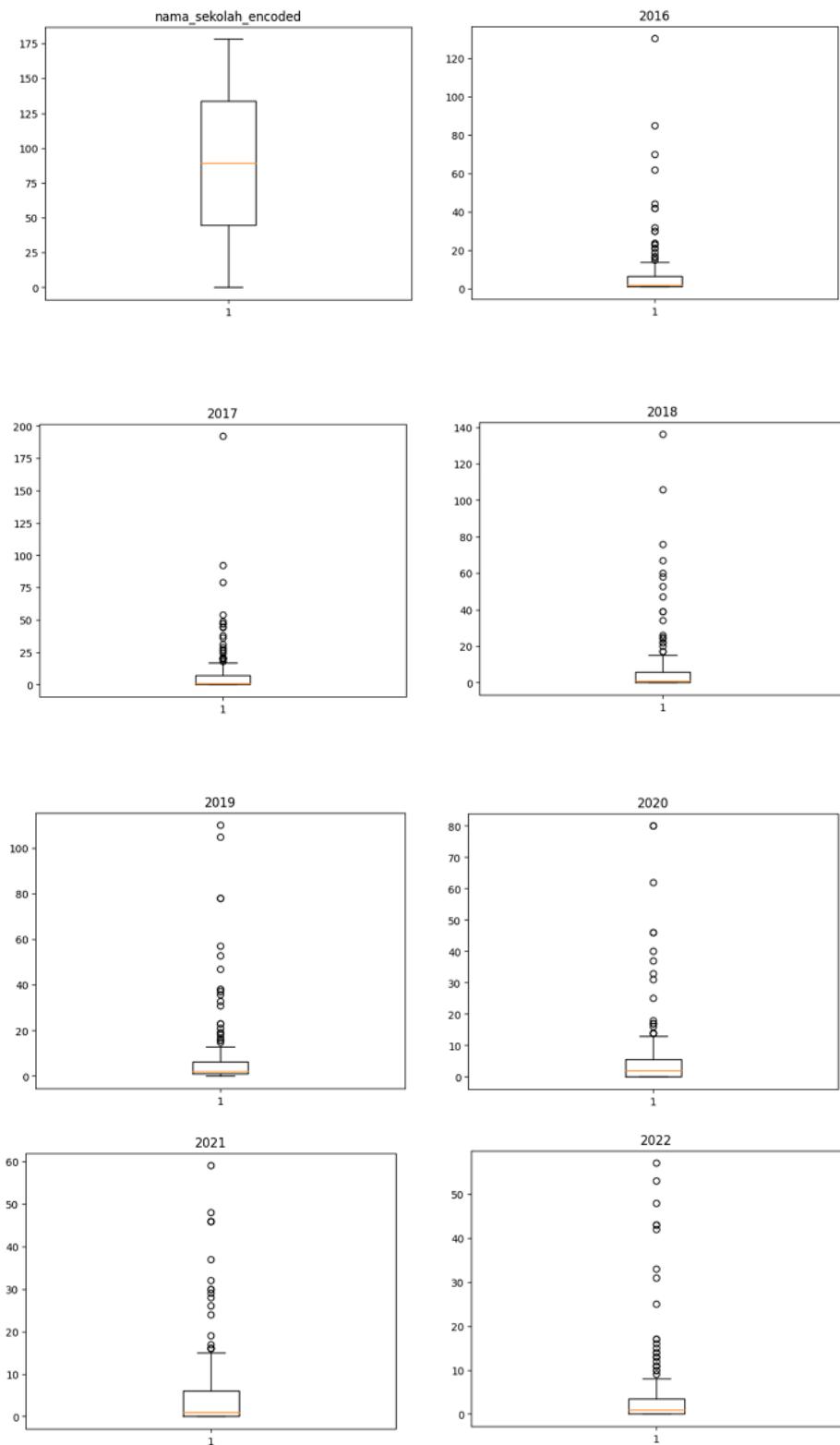
index	Nama Sekolah	nama_sekolah_encoded	2016	2017	2018	2019	2020	2021	2022
0	SMA ABDI SEJATI PERDAGANGAN	0	1	2	0	1	0	0	2
1	SMA ADVENT PEMATANG SIANTR	1	1	0	1	1	1	0	0
2	SMAASISI PEMATANG SIANTR	2	1	0	0	3	5	1	3
3	SMA BINTANG TIMUR BALIGE	3	70	54	53	110	46	48	53
4	SMA BINTANG TIMUR PEMATANG SIANTR	4	30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

Gambar 6. 2 Data setelah encoding

Pada gambar di atas, variabel ‘nama sekolah’ sudah dapat diolah karena sudah memiliki nilai numerik dari hasil encoding.

6.1.2. Pembahasan Normalisasi

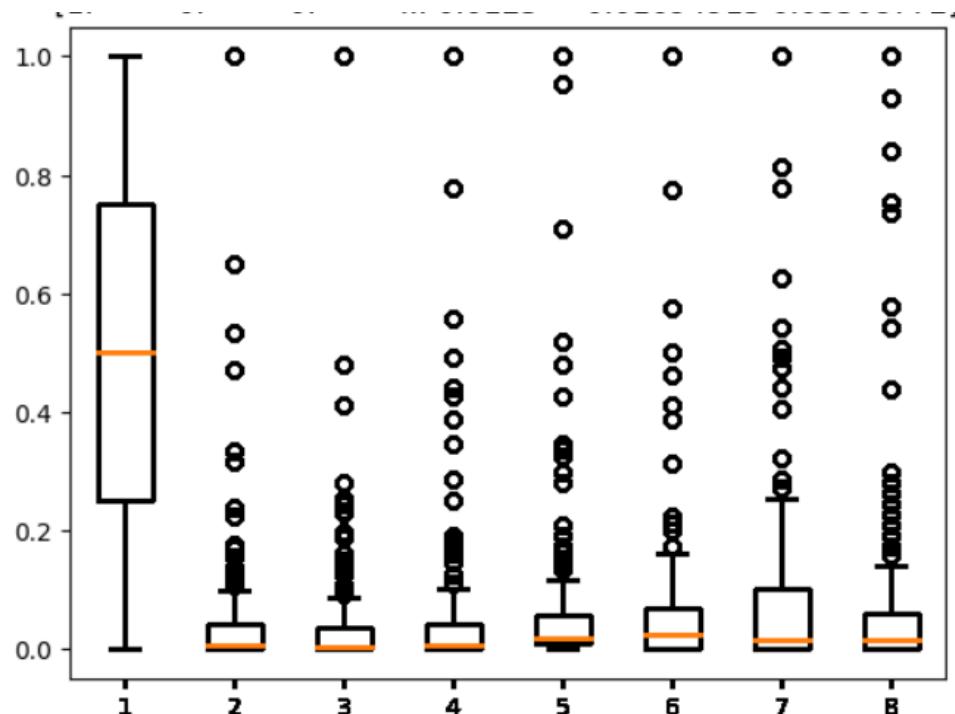
Normalisasi dilakukan untuk mengatasi terjadinya outlier pada data yang akan diolah. Salah satu penyebab dari outlier adalah adanya nilai ekstrim pada data yang mana terdapat perbedaan nilai yang signifikan dengan nilai-nilai variabel lainnya. Sebelum melakukan normalisasi, peneliti melakukan pengecekan data terlebih dahulu apakah dataset yang akan diolah mengalami outlier atau tidak. Berikut adalah gambaran nilai data pada dataset.



Berdasarkan gambaran boxplot di atas terlihat bahwa variabel-variabel data memiliki nilai skala yang berbeda yang mana. Karena seluruh variabel data memiliki skala yang berbeda, maka peneliti mencoba untuk mencari jumlah data yang mengalami outlier dengan menggunakan Z-Score. Berikut ini adalah hasil dari z-score.

Jumlah outliers: 179

Dari hasil z-score ditemukan bahwa terdapat 179 data yang mengalami outlier. Oleh karena itu peneliti melakukan normalisasi terhadap dataset yang akan digunakan. Pada saat melakukan normalisasi, nilai data diskalakan ke dalam rentang 0-1. Berikut gambaran skala nilai data setelah dilakukan normalisasi.



6.2. Pembahasan Algoritma

6.2.1. Pembahasan dan Hasil Pemodelan Support Vector Regression

Pada Algoritma SVR tahapan yang dilakukan yaitu inisialisasi nilai kernel dan parameter yang digunakan. Adapun nilai kernel dan parameter dibuat dengan cara berikut:

```
parameters = {
    "kernel": ["linear", "rbf", "sigmoid", "poly"],
    "C": [0.1, 1, 10, 100],
    "epsilon": [0.01, 0.1, 1],
    "gamma": [0.01, 0.1, 1]
}
```

Pada kode program diatas dilakukan inisialisasi nilai kernel, adapun kernel yang digunakan yaitu kernel linear, rbf, sigmoid, dan polynomial. Selanjutnya inisialisasi nilai parameter Cost dengan rentang nilai yaitu 0,1, 1, 10, 100, kemudian nilai parameter epsilon dengan rentang nilai 0,01, 0,1 dan 1, kemudian inisialisasi parameter gamma dengan rentang nilai 0,01, 0,1, 1. Pada setiap nilai parameter dan juga kernel yang telah diinisialisasikan dilakukan pemilihan parameter dan kernel terbaik dengan menggunakan metode Grid Search sebagaimana dilakukan menggunakan kode program berikut.

```
model = SVR()
cv = KFold(n_splits=5, shuffle=True, random_state=42)
grid_search = GridSearchCV(model, parameters, cv=cv,
                           scoring="neg_root_mean_squared_error")
grid_search.fit(X, y)
print("Parameter terbaik pada model SVR:",
      grid_search.best_params_)
```

Melalui kode program diatas, hal pertama yang dilakukan yaitu melakukan pemodelan menggunakan SVR. Kemudian, dilakukan pembuatan objek KFold sebagai pembagian data untuk cross-validation dengan 5 split, shuffle=True, dan random state=42. Selanjutnya, dibuat sebuah objek GridSearchCV yang akan melakukan penelusuran pada berbagai kombinasi parameter yang diberikan pada model SVR, dengan menggunakan objek K Fold sebagai metode cross-validation. Parameter yang ditelusuri ditentukan pada variabel parameters

Pada penelitian ini, dalam penentuan nilai k, peneliti membuat pengamatan pada beberapa nilai k. Nilai k yang digunakan pada pemodelan ini adalah nilai 2 sampai

dengan 10. Dari beberapa nilai k yang akan dicoba, peneliti akan mengamati nilai k yang akan menghasilkan nilai akurasi terbaik.

Setelah dilakukan optimasi dengan Grid Search maka didapatkan nilai parameter dan kernel terbaik yaitu dengan nilai parameter C adalah 100, parameter epsilon adalah 0,01, parameter gamma adalah 0,01, dan kernel terbaik yaitu kernel linear.

Setelah mendapatkan nilai parameter dan kernel yang terbaik, kemudian dilakukan proses cross-validation untuk melakukan evaluasi model yang menggunakan RMSE (Root Mean Square Error). Adapun hasil yang didapatkan nilai RMSE terbaik.

Berikut ini adalah hasil percobaan yang telah dilakukan pada nilai k dan parameter terbaik.

k	kernel	gamma	cost	epsilon	RMSE
2	Linear	0,01	100	0,01	0.00839
3	Linear	0,01	100	0,01	0.00561
4	Linear	0,01	100	0,01	0.00580
5	Linear	0,01	100	0,01	0.00455
6	Linear	0,01	100	0,01	0.00507
7	Linear	0,01	100	0,01	0.00535
8	Linear	0,01	100	0,01	0.00480
9	Linear	0,01	100	0,01	0.00478
10	Linear	0,01	100	0,01	0.00457

Berdasarkan hasil percobaan dengan menggunakan melakukan perubahan pada nilai k mulai dari rentang 2 sampai dengan 10. Didapatkan bahwa nilai RMSE terendah yang didapatkan yaitu 0.00455 yang mana dengan nilai k yaitu 5 digunakan sebagai model terbaik.

6.2.2. Pembahasan dan Hasil Pemodelan Artificial Neural Network

Arsitektur algoritma dibuat dengan menggunakan arsitektur multilayer, dimana ada terdapat 4 layer yang digunakan yaitu input layer, 2 hidden layer, dan output layer. Pembuatan bentuk arsitektur multilayer ini dibuat dengan cara berikut.

```
# Membangun bentuk arsitektur ANN
model = Sequential()
model.add(Dense(X.shape[1],kernel_initializer='uniform',
input_dim=X.shape[1]))
model.add(Dense(num_neurons_hidden_layer1,kernel_initializer=
'uniform', activation='linear'))
model.add(Dense(num_neurons_hidden_layer2,kernel_initializer=
'uniform', activation='sigmoid'))
model.add(Dense(num_neurons_output ,activation='linear'))
```

Setiap layer pada model memiliki parameter yang didefinisikan saat pembuatan. Pada kode diatas parameter yang didefinisikan adalah bobot yang diinisialisasi pada fungsi *kernel_initializer*. Bobot ini digunakan untuk menentukan seberapa pengaruhnya input neuron sebelumnya ke neuron berikutnya.

Selain parameter, hal lain yang perlu didefinisikan adalah *hyperparameter*. Ada beberapa parameter yang perlu didefinisikan yaitu jumlah neuron pada setiap layer, fungsi aktivasi, *batch_size*, *learning rate*, dan *epoch*. Untuk penentuan *epoch* dilakukan dengan mencoba beberapa nilai *epoch* kemudian epoch dengan nilai RMSE terendah akan menjadi nilai *epoch* yang akan digunakan dalam model. Sedangkan untuk *hyperparameter* lain ditentukan dengan menggunakan teknik *gridsearch parameter* seperti yang dilampirkan pada **Source Code 2. Grid Search ANN**. Berdasarkan hasil *gridsearch parameter* didapatkan hasil nilai *hyperparameter* terbaik antara lain :

- Learning rate : 0.01
- Batch size : 2
- Banyak neuron hidden layer 1 : 24
- Banyak neuron hidden layer 2 : 32
- Fungsi aktivasi hidden layer 1 : Linear

- Fungsi aktivasi hidden layer 2 : Sigmoid

Untuk nilai epoch yang diuji pada pelaksanaan training data adalah 50, 100, 250, 500, dan 1000.

Pada pemodelan, peneliti juga melakukan pembagian data menjadi data train dan data set dengan menggunakan teknik k-fold. Dalam penentuan nilai k, peneliti membuat pengamatan pada beberapa nilai k. Nilai k yang digunakan pada pemodelan ini adalah nilai 2 sampai 10. Dari beberapa nilai k yang akan dicoba, peneliti akan mengamati nilai k yang akan menghasilkan nilai akurasi terbaik.

Setelah dilakukan pendefinisian nilai hyperparameter dan nilai k, selanjutnya peneliti melakukan pemodelan dengan melakukan percobaan pada beberapa nilai epoch dan nilai k yang digunakan pada saat pembentukan model. Berdasarkan data hasil percobaan pada **lampiran hasil implementasi artificial neural network** yang telah dilakukan pada model didapatkan hasil bahwa model dengan nilai RMSE terendah terdapat pada percobaan dengan nilai k = 8 dan *epoch* = 250. Pada hasil training data dengan nilai k dan nilai epoch tersebut didapatkan hasil RMSE sebesar 0.6785 dan nilai akurasi sebesar 48.10%.

6.3. Pembahasan Hasil Evaluasi Model

Pada bagian ini akan menjelaskan tentang bagaimana nilai hasil evaluasi dari hasil pemodelan algoritma yang telah diimplementasikan. Berdasarkan hasil implementasi model dan evaluasi model yang dilakukan, kedua model menghasilkan nilai RMSE yang berbeda. Pada model algoritma SVR, setelah dilakukan pengujian pada nilai k=2 sampai dengan k=10 diperoleh nilai RMSE terendah pada nilai k=5 dengan nilai RMSE = 0.00455. Pada model algoritma ANN, setelah dilakukan pengujian nilai k=2 sampai dengan k=10 dan pada epoch 50, 100, 250, 500, dan 1000 yang diuji pada setiap nilai k diperoleh nilai RMSE paling rendah adalah 0.6785. Berdasarkan hasil evaluasi model yang telah diperoleh maka pemodelan algoritma terbaik adalah algoritma SVR.

BAB 7

KESIMPULAN DAN SARAN

7.1. Kesimpulan

Berdasarkan hasil penelitian dan implementasi dapat diberi kesimpulan sebagai berikut:

1. Dari hasil yang didapatkan dapat disimpulkan bahwa nilai error terendah pada pemodelan menggunakan SVR berada pada nilai 0.00455 sedangkan pemodelan dengan menggunakan ANN memiliki nilai error terendah pada nilai 0.6785. Sehingga dapat disimpulkan bahwa kinerja (performansi) dari metode *Support Vector Regression* lebih baik dibandingkan dengan metode *Artificial Neural Network*
2. Website sistem prediksi jumlah pendaftar di Institut Teknologi Del sudah berjalan dengan fungsionalitas dan penggunaan fungsi website sudah dapat berjalan dengan normal

7.2. Saran

LAMPIRAN

Lampiran A Source Code

Source Code 1. Encoding Data

```
#import library yang akan digunakan
import pandas as pd
import numpy as np
from sklearn.preprocessing import LabelEncoder
import matplotlib.pyplot as plt

# Load dataset
data = pd.read_excel('data_breakdown.xlsx')

# Membuat objek label encoder
encoder = LabelEncoder()

# Melakukan encoding pada variabel 'School Name'
data['nama_sekolah_encoded'] =
encoder.fit_transform(data['Nama Sekolah'])

#Mengurutkan posisi kolom agar lebih mudah digunakan
coloum_order=['Nama Sekolah',
'nama_sekolah_encoded',2016,2017,2018,2019,2020,2021,2022]

#import hasil dataset yang telah di encoding
data_prep.to_csv('data-prep.csv', index=0)
```

Source Code 2. Grid Search ANN

```
#import library
import numpy as np
import pandas as pd
import tensorflow as tf
from keras.layers import Dense
from keras.models import Sequential
from sklearn.preprocessing import StandardScaler
```

```

from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import KFold
from sklearn.model_selection import GridSearchCV
from keras.wrappers.scikit_learn import KerasClassifier

# Memuat dataset yang telah dilakukan preprocessing encoding
from google.colab import drive

drive.mount('/content/gdrive')
%cd /content/gdrive/MyDrive/ANN/Code/Data
data = pd.read_csv('data-prep.csv')

# Penentuan Variabel Input(X) dan Variabel Output(y)
X = data.iloc[:, 1: ].values
y = data.iloc[:, -1 ].values.reshape(-1, 1)

# Menormalisasikan variabel input dengan menggunakan
# normalisasi min-max
scaler = StandardScaler()
X= scaler.fit_transform(X)
# Fungsi untuk membuat model
def create_model(num_neurons_hidden_layer1,
                 num_neurons_hidden_layer2, activation1, activation2,
                 learning_rate):
    model = Sequential()
    model.add(Dense(X.shape[1],kernel_initializer='uniform',
                    input_dim=X.shape[1]))

    model.add(Dense(num_neurons_hidden_layer1,kernel_initializer='uniform',
                    activation=activation1))

    model.add(Dense(num_neurons_hidden_layer2,kernel_initializer='uniform',
                    activation=activation2))
    model.add(Dense(1, activation='linear'))

    model.compile(loss='mean_squared_error',
                  optimizer=tf.keras.optimizers.Adam(learning_rate),

```

```

metrics=['accuracy'])

return model

# Membungkus objek model ke dalam KerasClassifier
model = KerasClassifier(build_fn=create_model)

# Definisikan hyperparameter yang akan dioptimasi
param_grid = {'num_neurons_hidden_layer1': [12,16,24, 64,
                                             128],
              'num_neurons_hidden_layer2': [12,16,24, 32,
                                             64,128],
              'batch_size':[2,16,32,42],
              'activation1':['relu', 'sigmoid','linear'],
              'activation2':['relu','sigmoid','linear'],
              'learning_rate': [0.01, 0.001]}

# Inisialisasi objek GridSearchCV
grid_search = GridSearchCV(estimator=model,
                           param_grid=param_grid, cv=2)

# Fit the grid search object to the data
grid_search.fit(X, y)

# Print the best parameters and the corresponding MSE score
print("Best Parameters: ", 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.preprocessing import LabelEncoder,MinMaxScaler
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split,

```

```

GridSearchCV, KFold
from sklearn.metrics import mean_squared_error,
mean_absolute_error, r2_score
import matplotlib.pyplot as plt
# Melakukan Load data
data = pd.read_csv("/content/drive/My
Drive/Tugas_Aakhir/data-prep.csv")
# Penentuan Variabel Input(X) dan Variabel Output(y)
X = data.iloc[:, 1: ].values
y = data.iloc[:, -1].values.reshape(-1, 1)

# Menormalisasikan variabel input dengan menggunakan
# normalisasi min-max
scaler = StandardScaler()
X= scaler.fit_transform(X)
# Inisialisasi Parameter
parameters = {
    "kernel": ["linear", "rbf", "sigmoid", "poly"],
    "C": [0.1, 1, 10, 100, 500],
    "epsilon": [0.01, 0.1, 1],
    "gamma": [0.01, 0.1, 1]
}
# Melakukan pemodelan menggunakan SVR
model = SVR()
cv = KFold(n_splits=5, shuffle=True, random_state=42)
grid_search = GridSearchCV(model, parameters, cv=cv,
                           scoring="neg_root_mean_squared_error")
grid_search.fit(X, y)
print("Parameter terbaik pada model SVR:",
      grid_search.best_params_)

```

Source Code 4. Pemodelan Algoritma ANN

```

#import library
from keras.models import model_from_json
import numpy as np
import pandas as pd
import tensorflow as tf

```

```

from keras.layers import Dense
import matplotlib.pyplot as plt
from keras.models import Sequential
from sklearn.preprocessing import MinMaxScaler
from keras.wrappers.scikit_learn import KerasRegressor
from sklearn.model_selection import KFold

# Memuat dataset yang telah dilakukan preprocessing encoding
from google.colab import drive

drive.mount('/content/gdrive')
%cd /content/gdrive/MyDrive/ANN/Code/Data
data = pd.read_csv('data-prep.csv')

# Penentuan Variabel Input(X) dan Variabel Output(y)
X = data.iloc[:, 1:].values
y = data.iloc[:, -1].values.reshape(-1, 1)

# Menormalisasikan variabel input dengan menggunakan
normalisasi min-max
scaler = MinMaxScaler()
X= scaler.fit_transform(X)

```

PEMODELAN

```

# Define the hyperparameters
learning_rate = 0.01
num_epochs = 250
batch_size = 2
num_neurons_hidden_layer_1 = 32
num_neurons_hidden_layer_2 = 24
num_neurons_output = 1

# Define k-fold cross validation
n_folds = 8
kf = KFold(n_splits=n_folds, shuffle=True, random_state=16)

#Membuat variabel untuk menyimpan score MSE dan acc
rmse_scores = []
acc_scores = []

```

```

#TRAIN

for train_idx, test_idx in kf.split(X, y):
    # Split the data into training set and testing set
    X_train, X_test = X[train_idx], X[test_idx]
    y_train, y_test = y[train_idx], y[test_idx]

    # Membangun bentuk arsitektur ANN
    model = Sequential()
    model.add(Dense(X.shape[1], kernel_initializer='uniform',
    input_dim=X.shape[1]))

    model.add(Dense(num_neurons_hidden_layer1,kernel_initializer=
    'uniform', activation='linear'))

    model.add(Dense(num_neurons_hidden_layer2,kernel_initializer=
    'uniform', activation='sigmoid'))
    model.add(Dense(num_neurons_output ,
    activation='linear'))

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

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

    # Evaluate the model on the training set
    train_loss, train_acc = model.evaluate(X_train, y_train,
    verbose=0)
    # Evaluate the model on the test set
    test_loss, test_acc = model.evaluate(X_test, y_test,
    verbose=0)

    # Menghitung nilai RMSE dan akurasi pada data testing
    setiap fold
    rmse_scores.append(np.sqrt(test_loss))
    acc_scores.append(test_acc)

```

```

print("\n")

#Menampilkan akurasi data test dan data training
print("Training Accuracy: {:.2f}%".format(train_acc * 100))
print("Test Accuracy: {:.2f}%".format(test_acc * 100))

#Print rata-rata MSE dan akurasi dari k-fold cross-validation
print(f'Average RMSE score: {np.mean(rmse_scores):.4f} ')
print(f'Average acc score: {np.mean(acc_scores)*100:.2f} %')
#Menampilkan plot untuk melihat grafik akurasi dan loss
# Plot accuracy
plt.plot(history.history['acc'])
plt.plot(history.history['val_acc'])
plt.title('Model Accuracy')
plt.ylabel('Accuracy')
plt.xlabel('Epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

# Plot loss
plt.plot(history.history['loss'])
plt.plot(history.history['val_loss'])
plt.title('Model Loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['train', 'test'], loc='upper left')
plt.show()

#Melakukan prediksi dengan menggunakan data X_test
forecasting = model.predict(X_test)
print(forecasting)

# Evaluasi Hasil Prediksi dengan RMSE
from sklearn.metrics import mean_squared_error
from math import sqrt
mse = mean_squared_error(forecasting, y_test)
rmse = sqrt(mse)
print('RMSE: %f' % rmse)

```

Source Code 5. Pemodelan Algoritma 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.preprocessing import LabelEncoder,MinMaxScaler
from sklearn.svm import SVR
from sklearn.model_selection import train_test_split,
GridSearchCV, KFold
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
# Melakukan Load data
data = pd.read_csv("/content/drive/My Drive/Tugas_Aakhir/data-
prep.csv")
# Penentuan Variabel Input(X) dan Variabel Output(y)
X = data.iloc[:, 1: ].values
y = data.iloc[:, -1].values.reshape(-1, 1)

# Menormalisasikan variabel input dengan menggunakan
normalisasi min-max
scaler = MinMaxScaler()
X= scaler.fit_transform(X)
# Inisialisasi Parameter
parameters = {
    "kernel": ["linear", "rbf", "sigmoid", "poly"],
    "C": [0.1, 1, 10, 100, 500],
    "epsilon": [0.01, 0.1, 1],
    "gamma": [0.01, 0.1, 1]
}
# Melakukan pemodelan menggunakan SVR
model = SVR()
cv = KFold(n_splits=5, shuffle=True, random_state=42)
grid_search = GridSearchCV(model, parameters, cv=cv,
scoring="neg_root_mean_squared_error")
grid_search.fit(X, y)
print("Parameter terbaik pada model SVR:",
grid_search.best_params_)

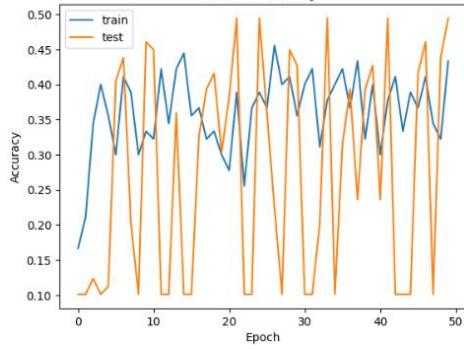
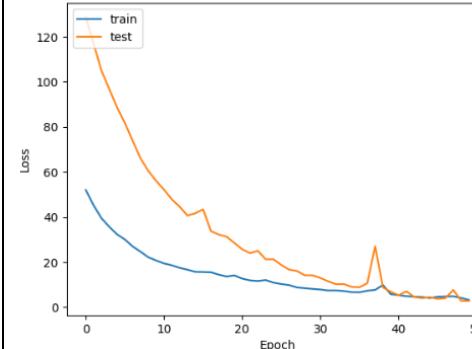
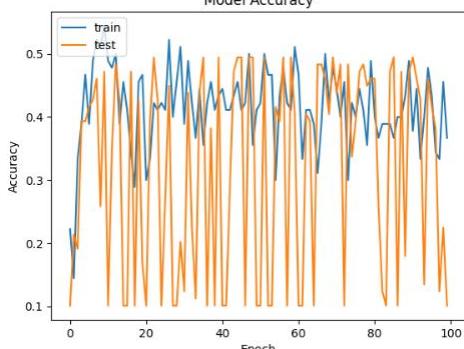
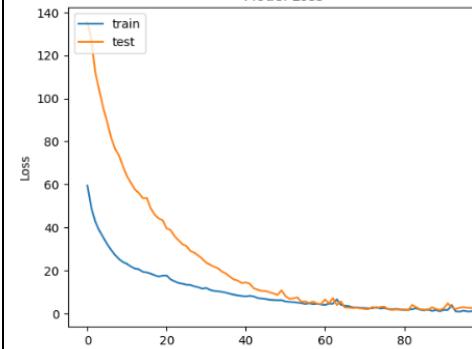
```

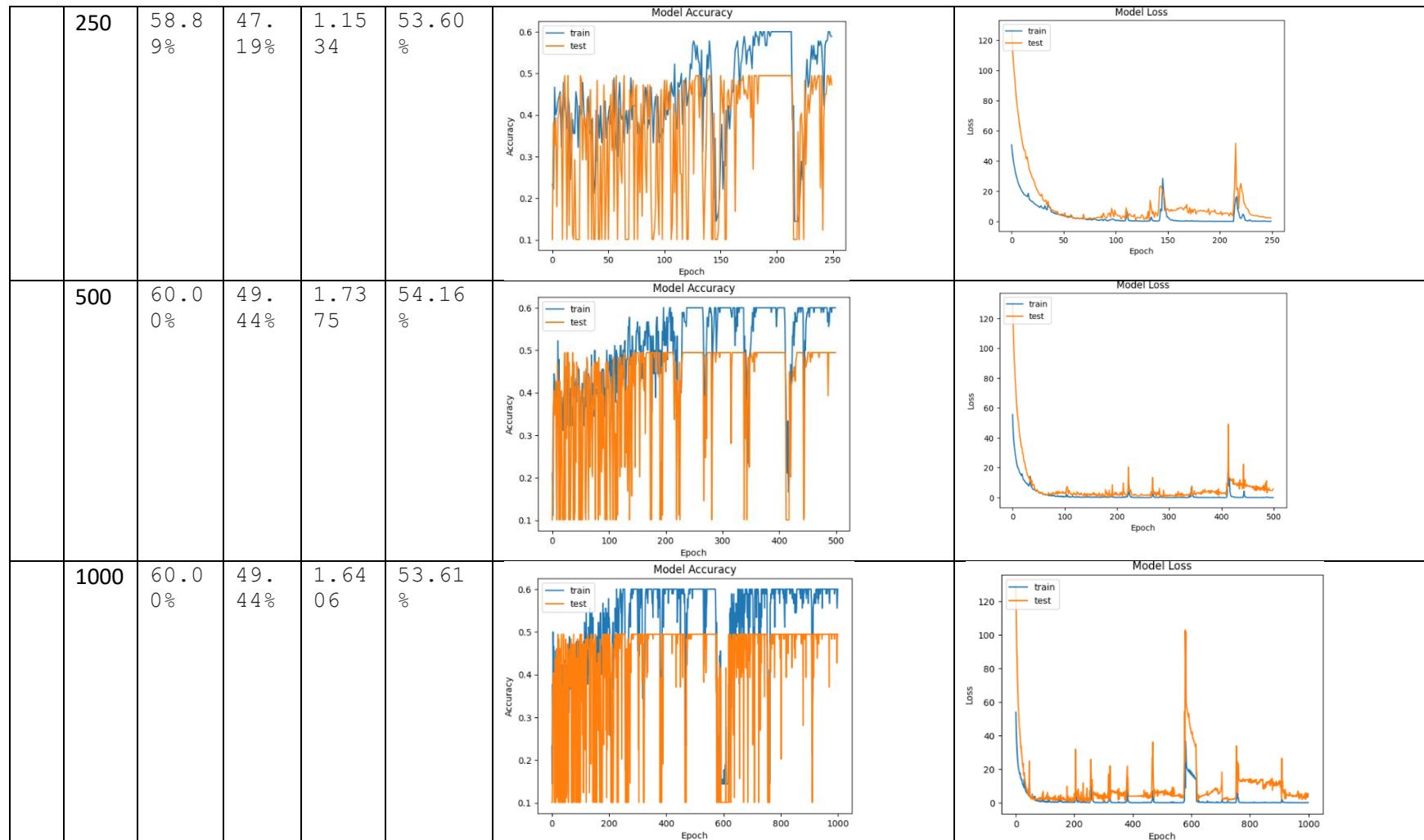
```
# Melakukan pelatihan dengan parameter terbaik
best_model = SVR(**grid_search.best_params_)
best_model.fit(X, y)
# Melakukan evaluasi model dengan menggunakan RMSE
rmse_scores = []
for train_index, test_index in cv.split(X, y):
    X_train, X_test = X[train_index], X[test_index]
    y_train, y_test = y[train_index], y[test_index]
    best_model.fit(X_train, y_train)
    y_pred = best_model.predict(X_test)
    rmse = np.sqrt(mean_squared_error(y_test, y_pred))
    rmse_scores.append(rmse)

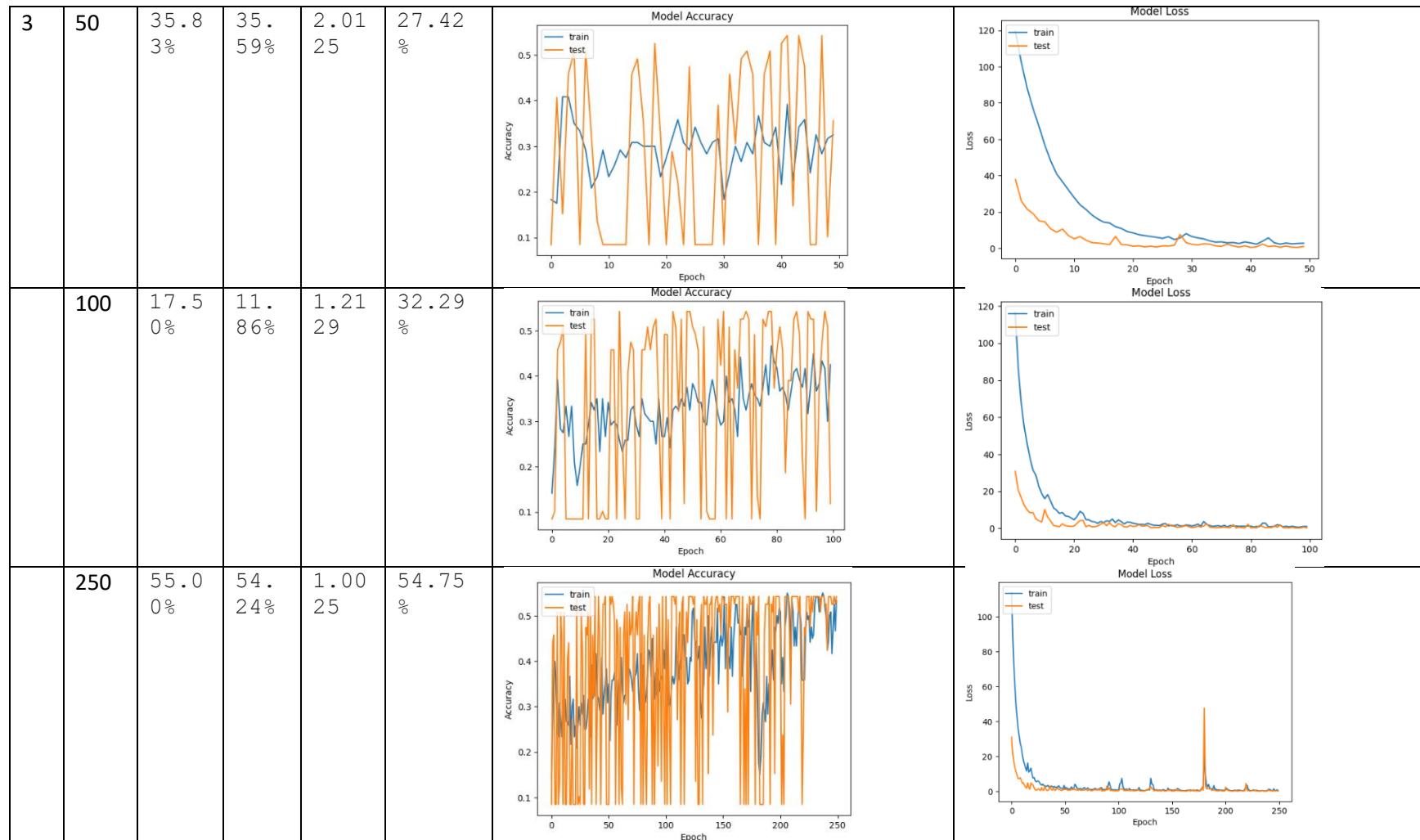
print("RMSE:", np.mean(rmse_scores))
```

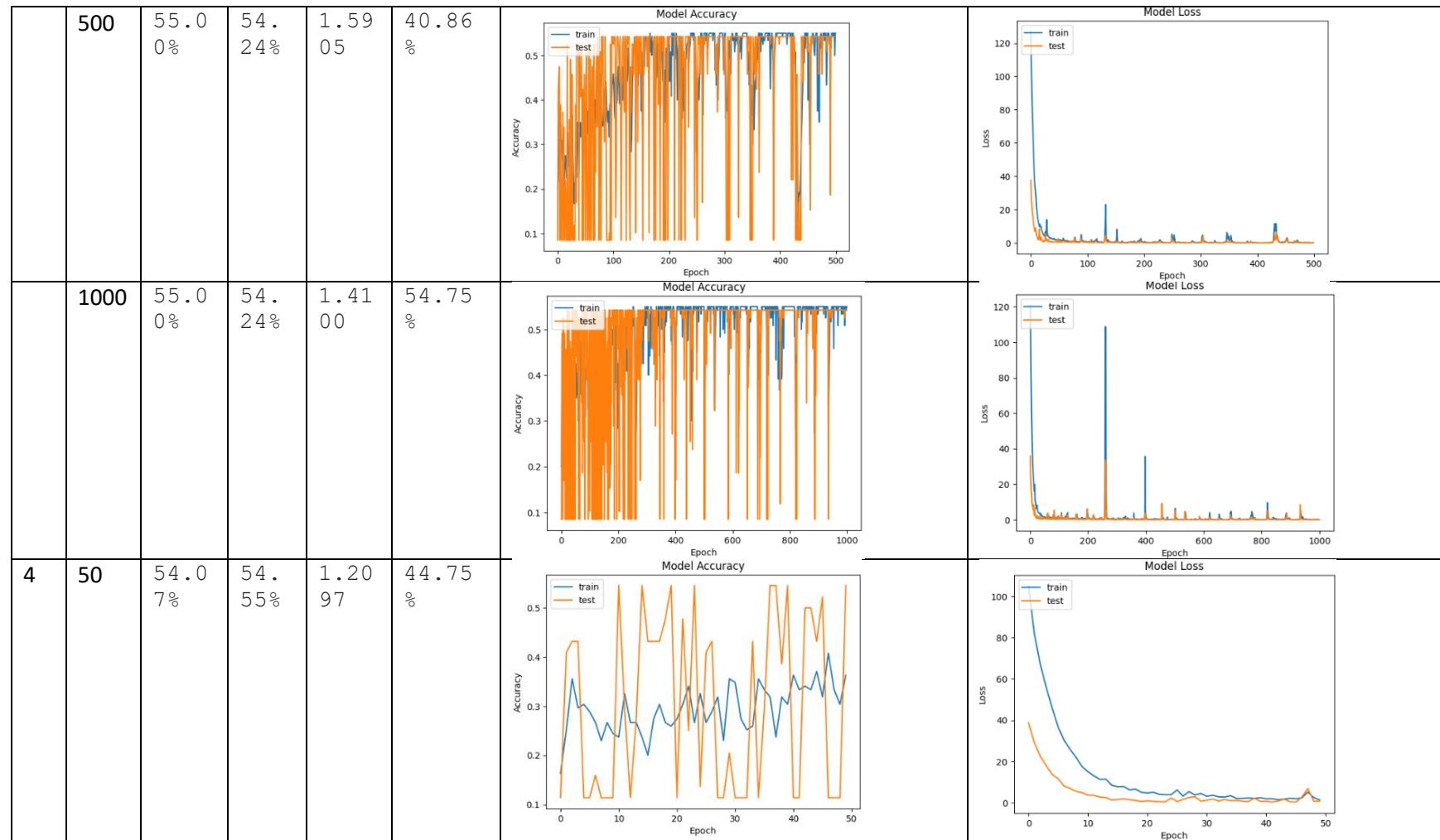
Lampiran B Hasil Implementasi Algoritma

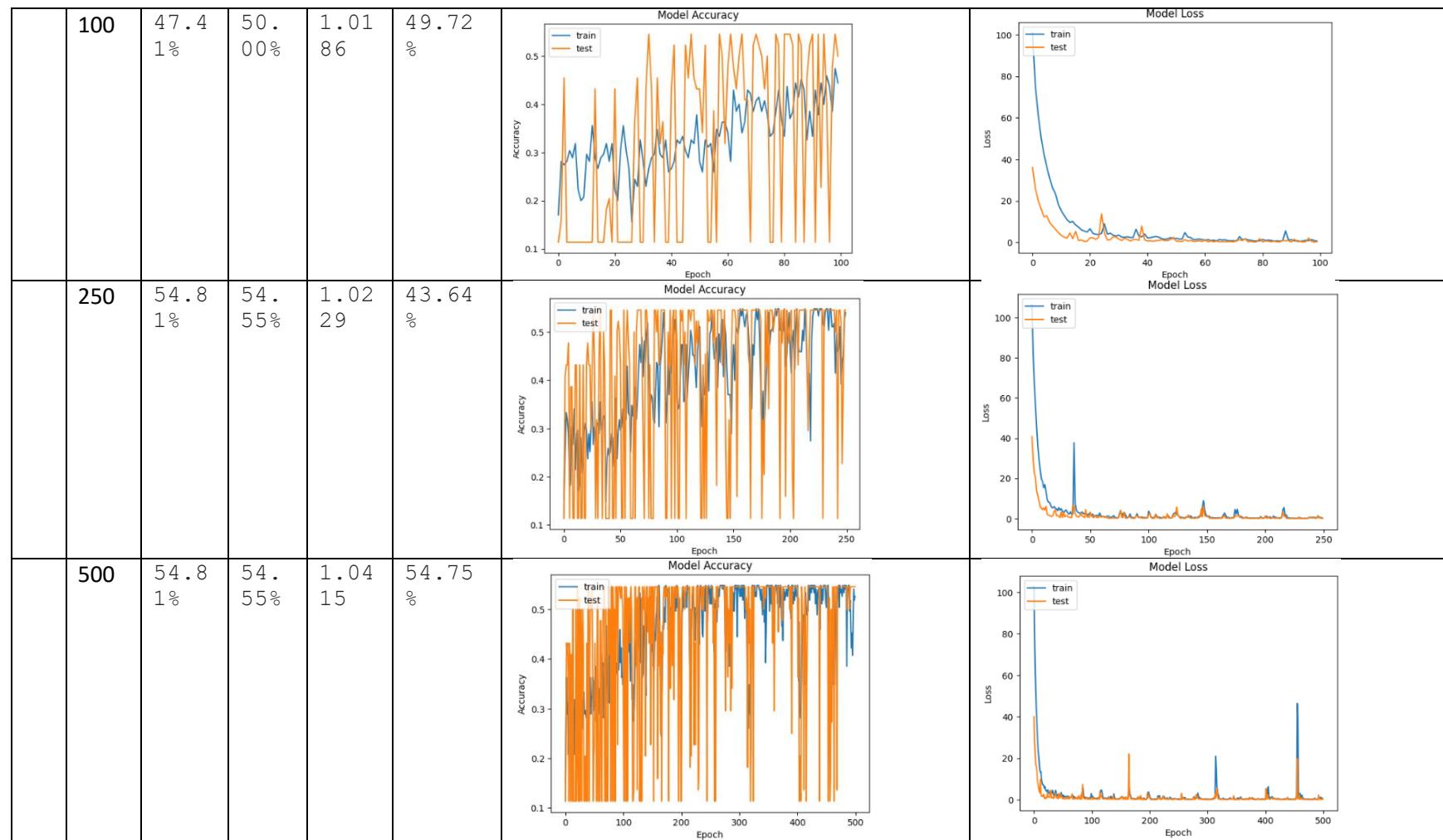
B.1. Hasil Implementasi Artificial Neural Network

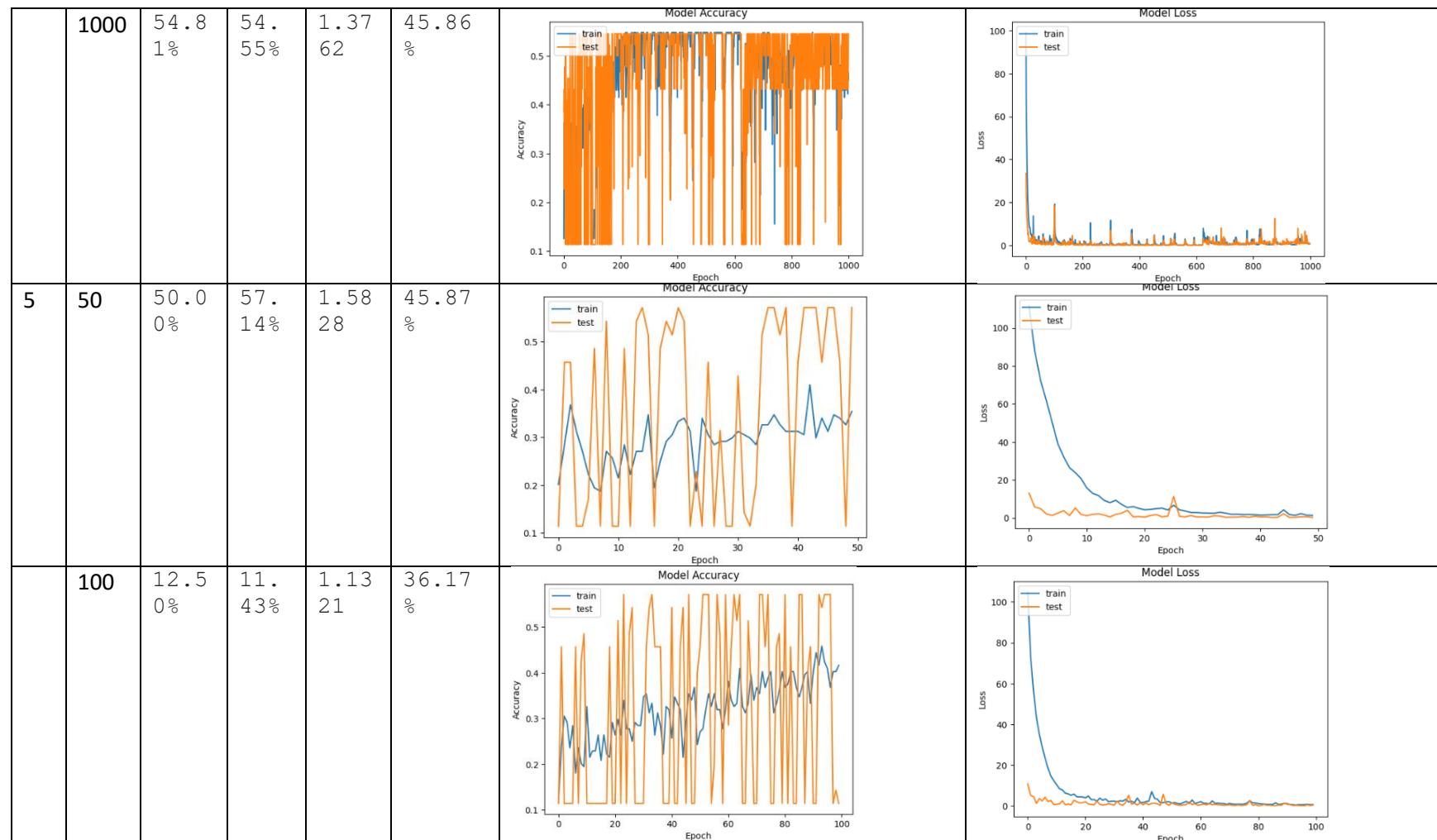
K	Epoch	Training Acc	Test Acc	RMSE	Aver Acc	Grafik Accuracy		Grafik Loss	
						Model Accuracy		Model Loss	
2	50	58.89%	49.44%	1.8558	31.94%				
	100	14.44%	10.11%	1.2564	12.28%				

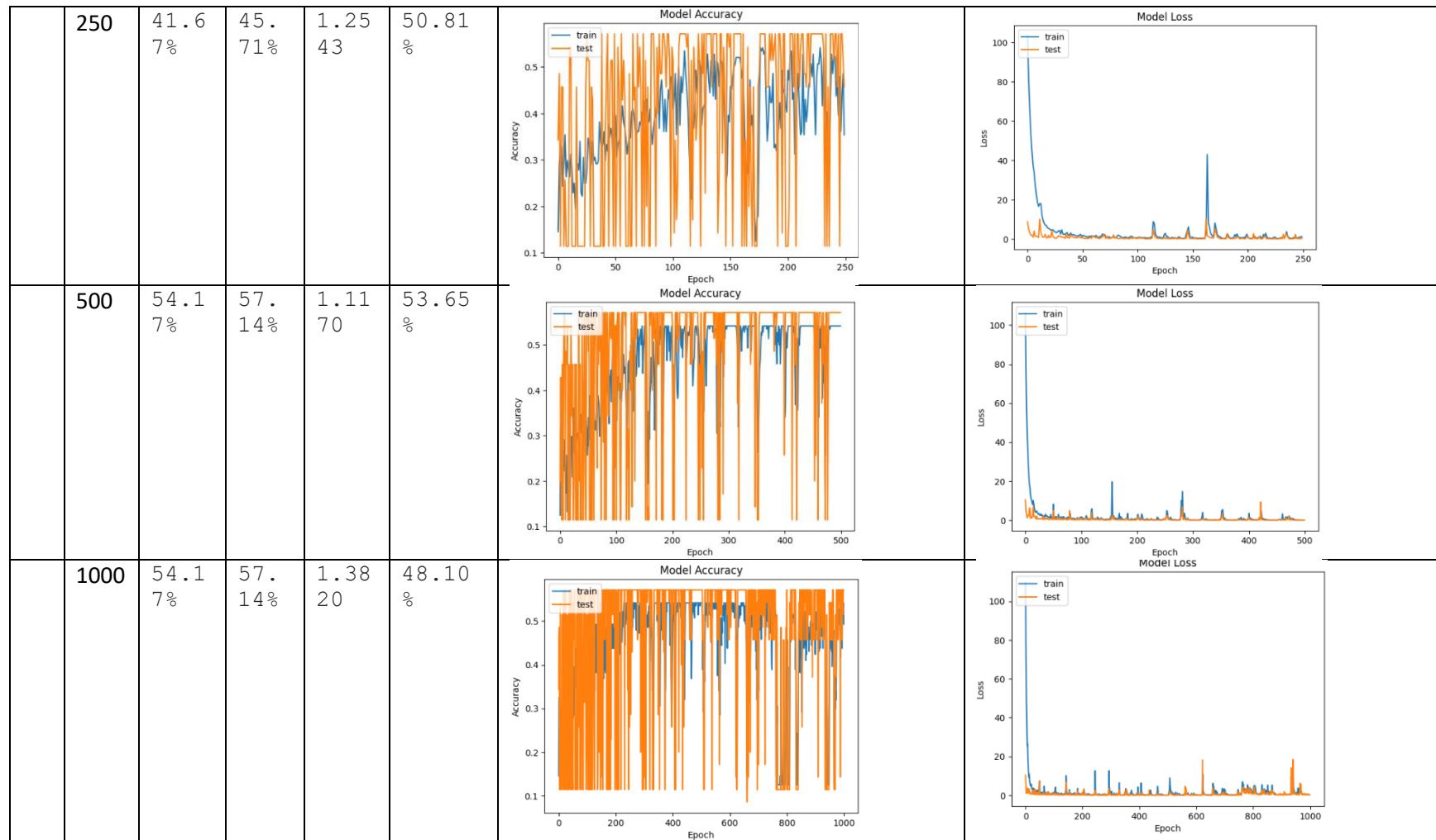


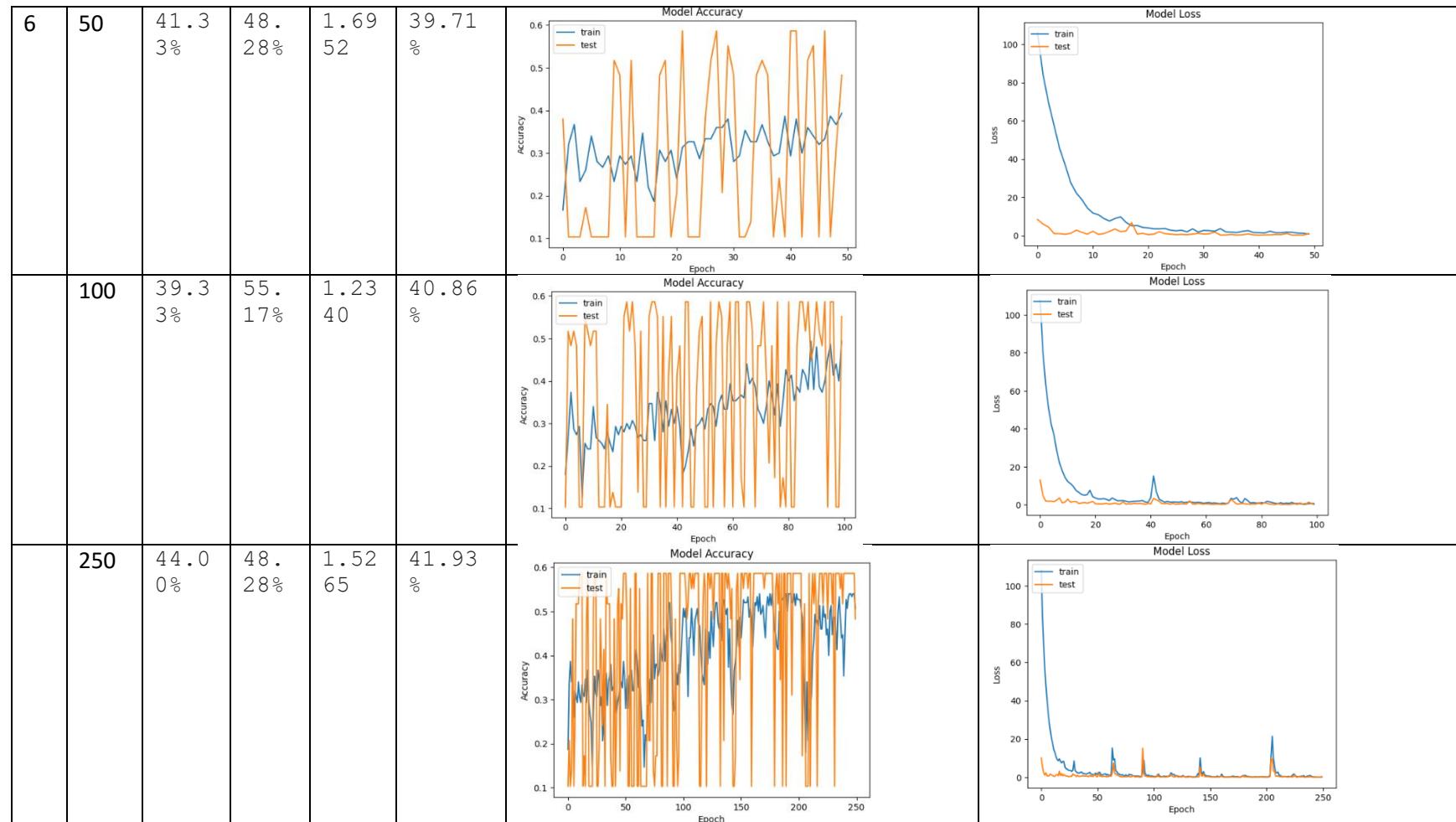


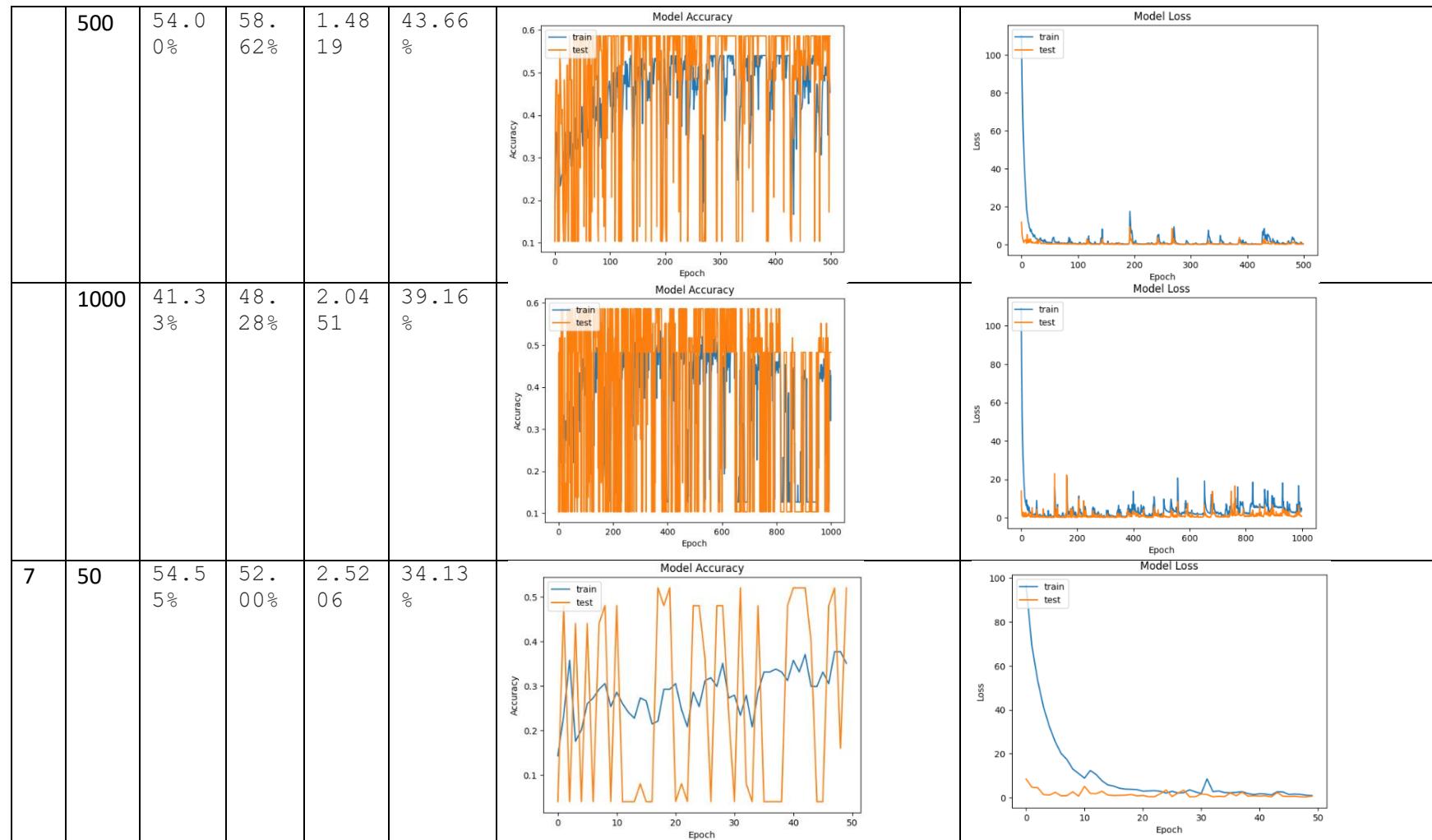


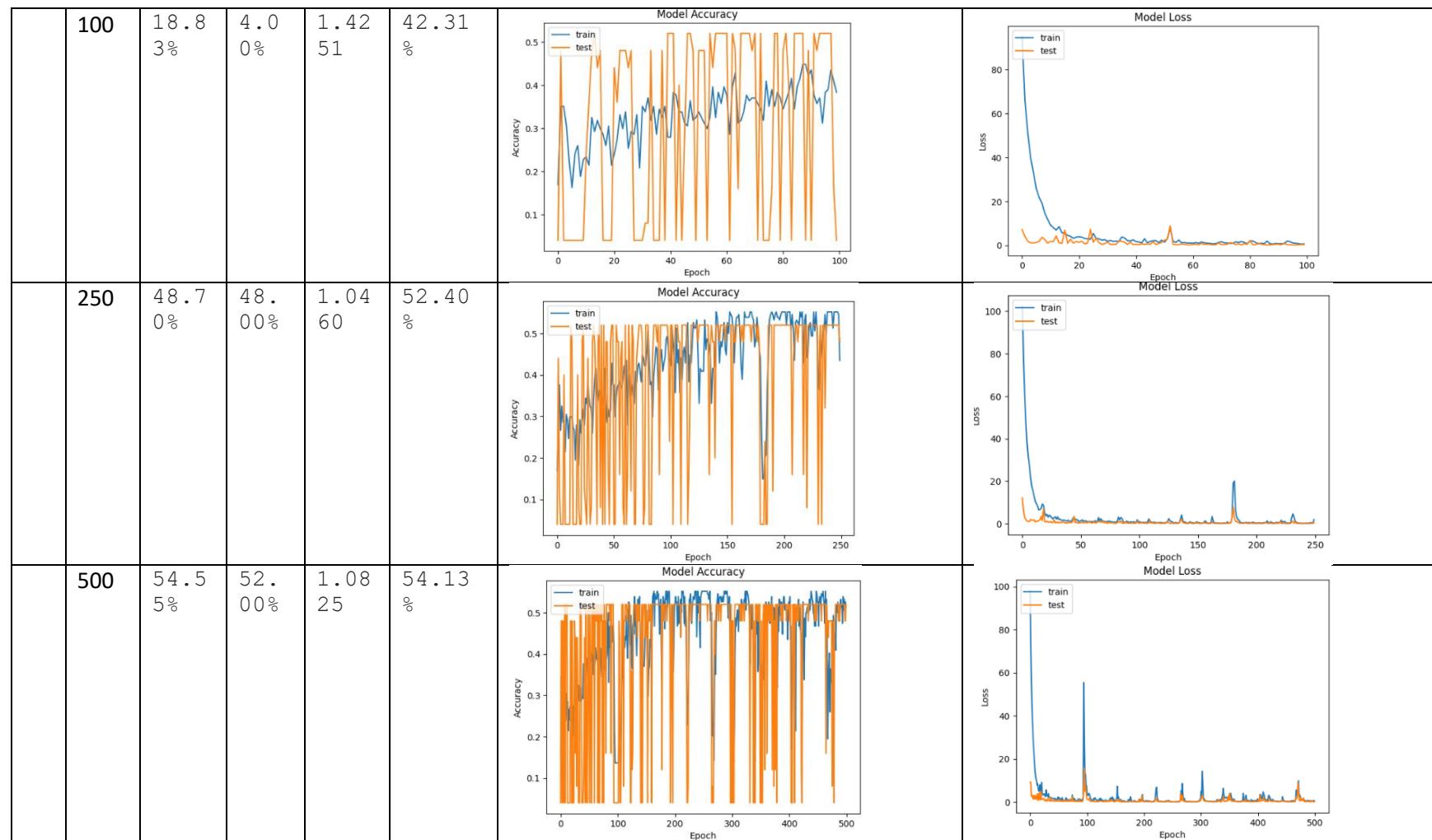


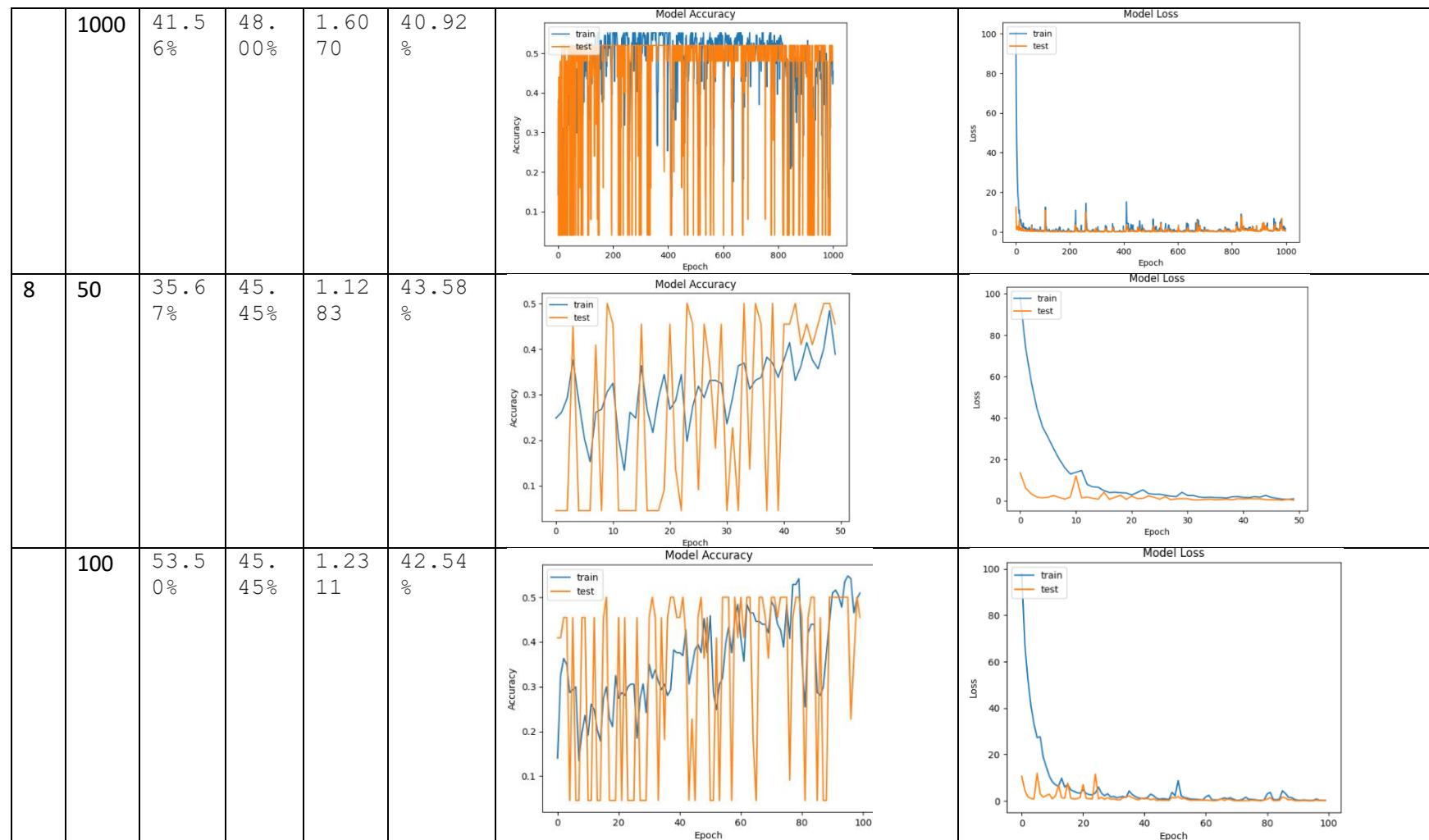


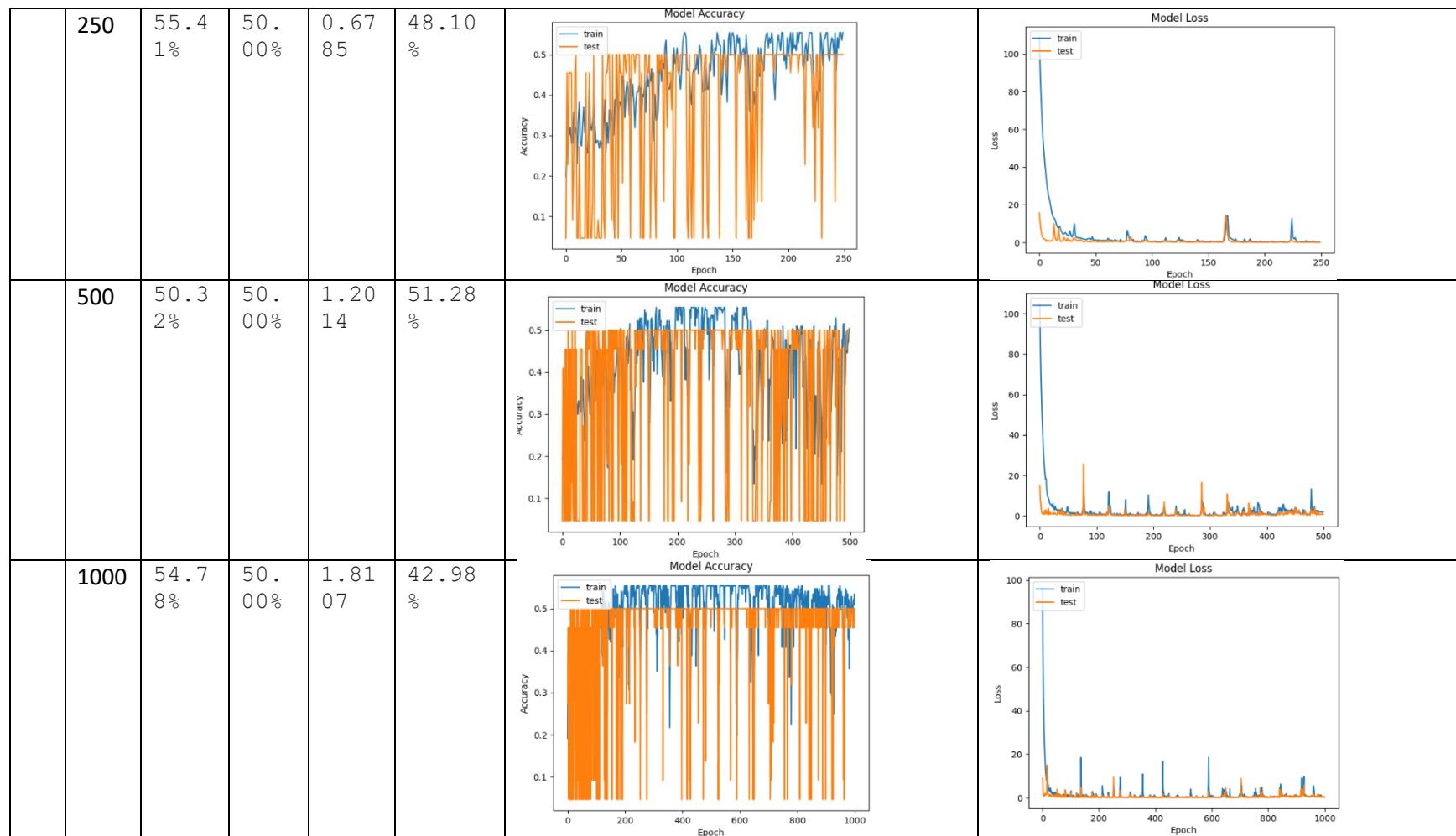


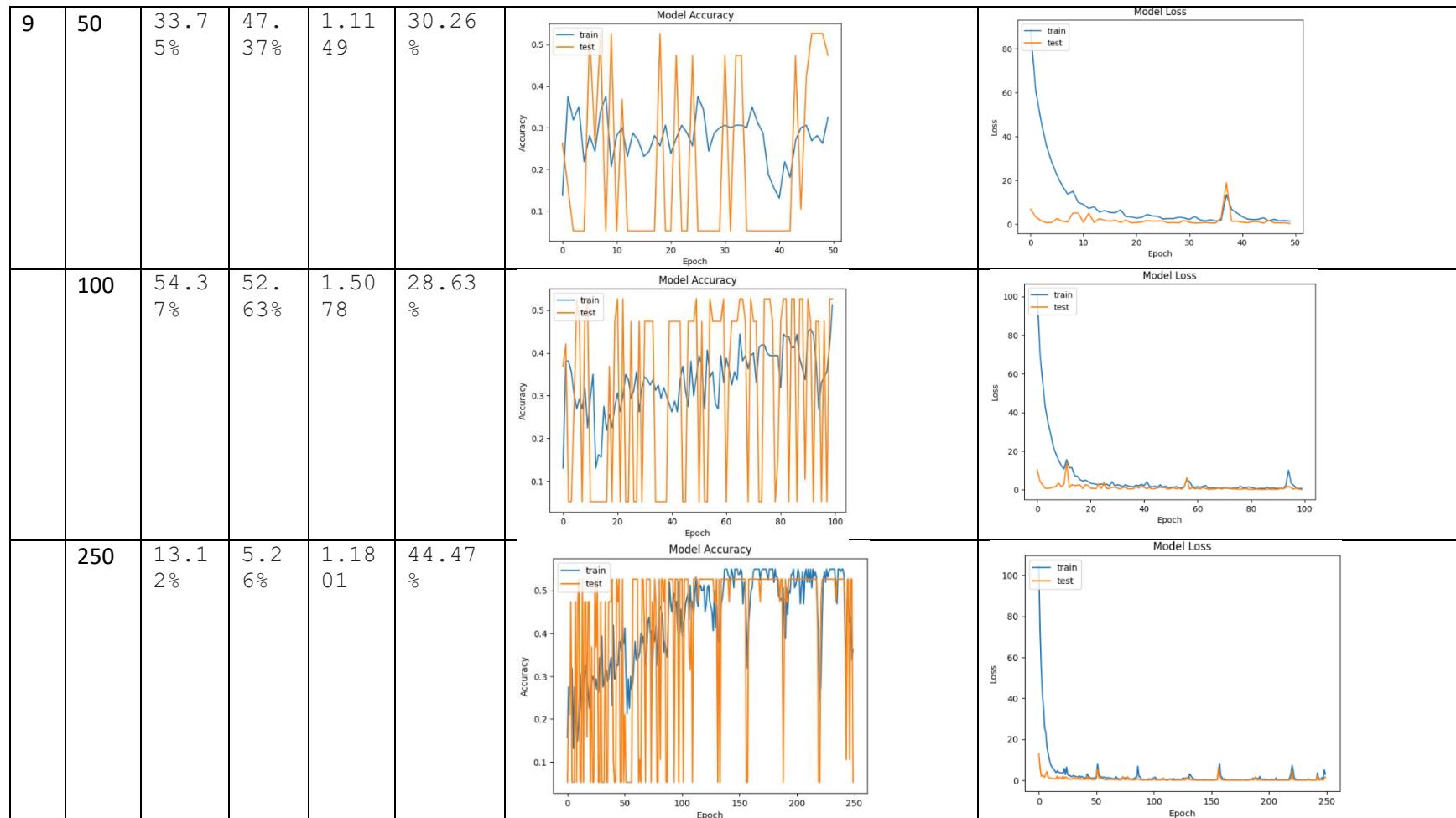


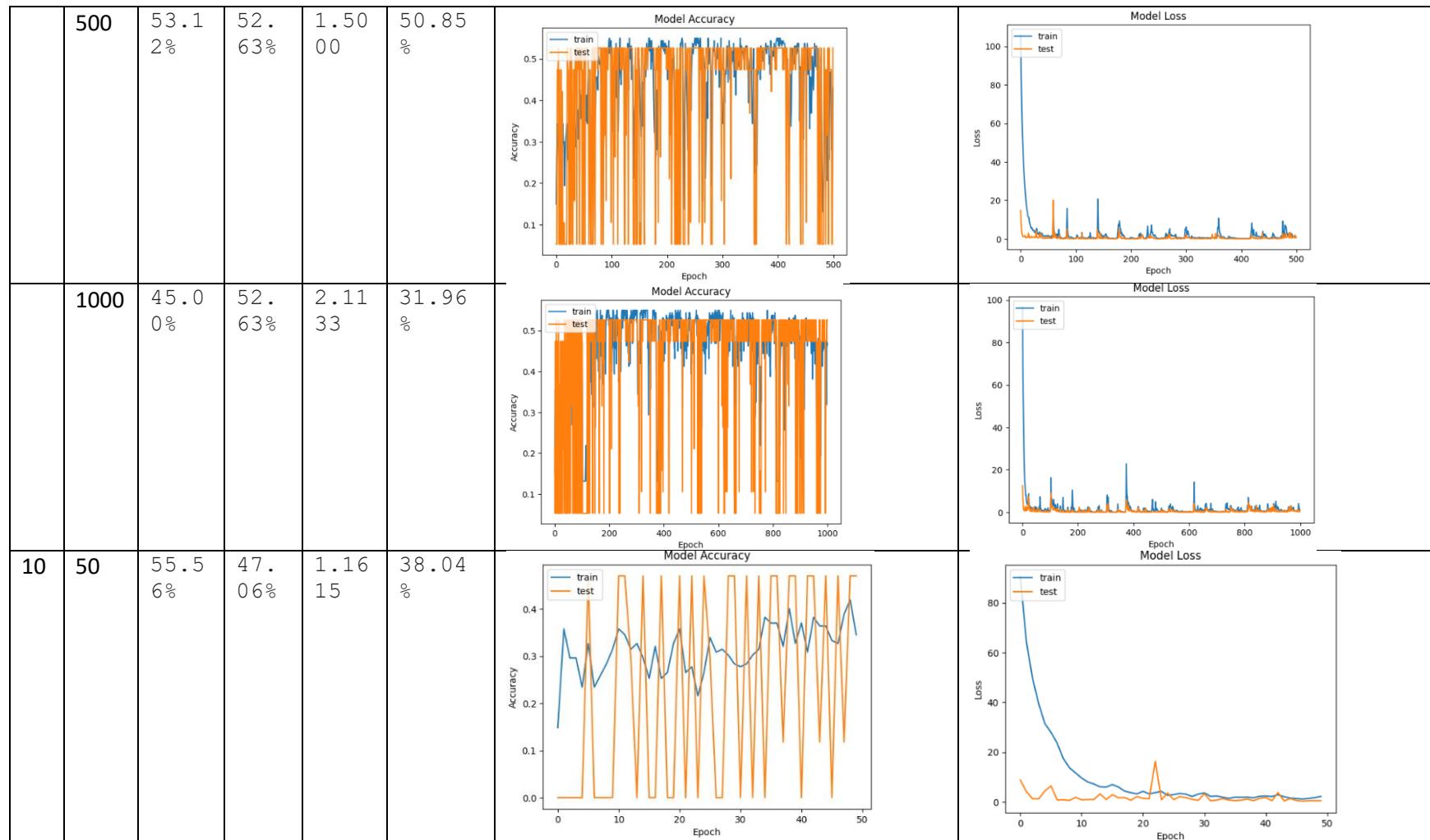


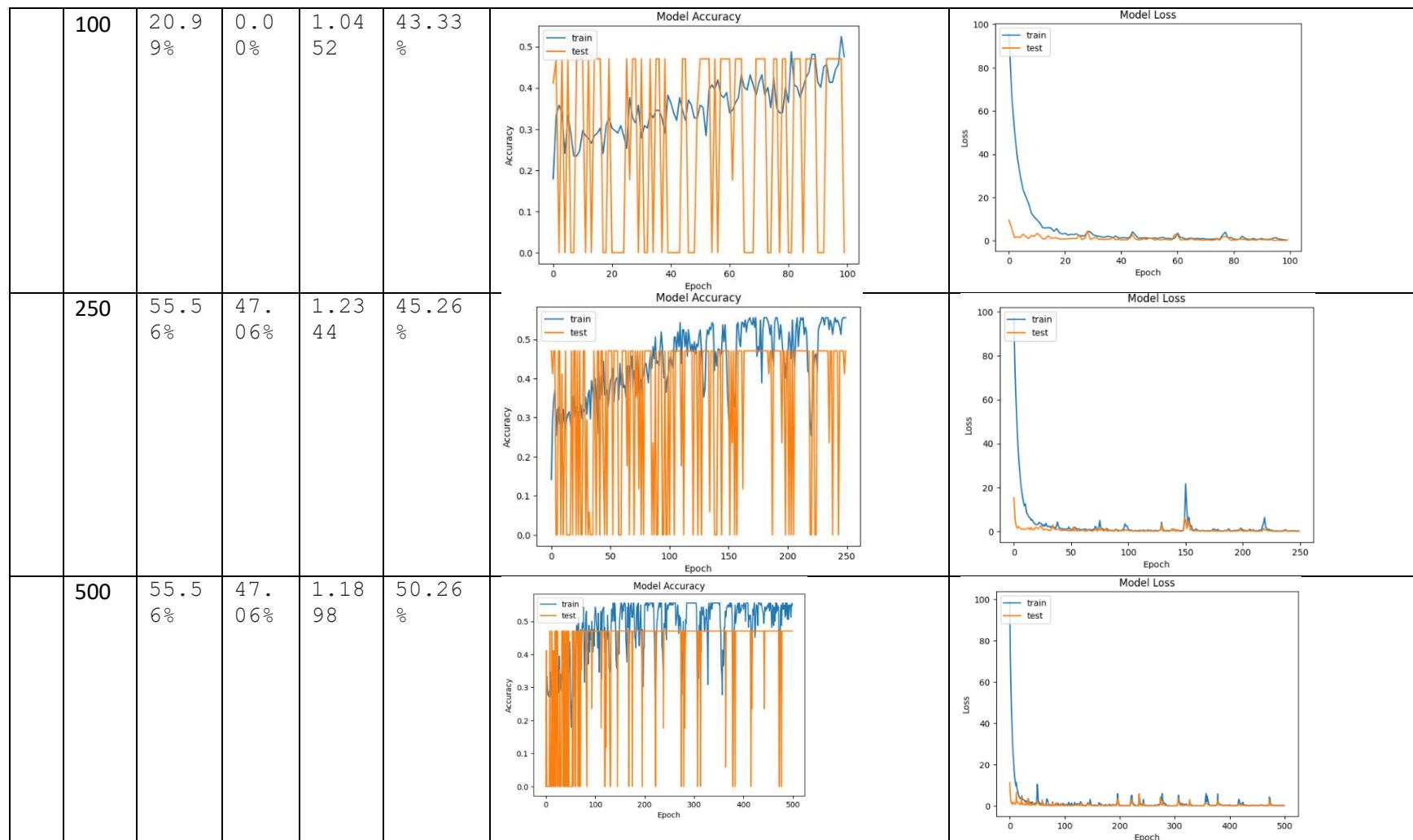


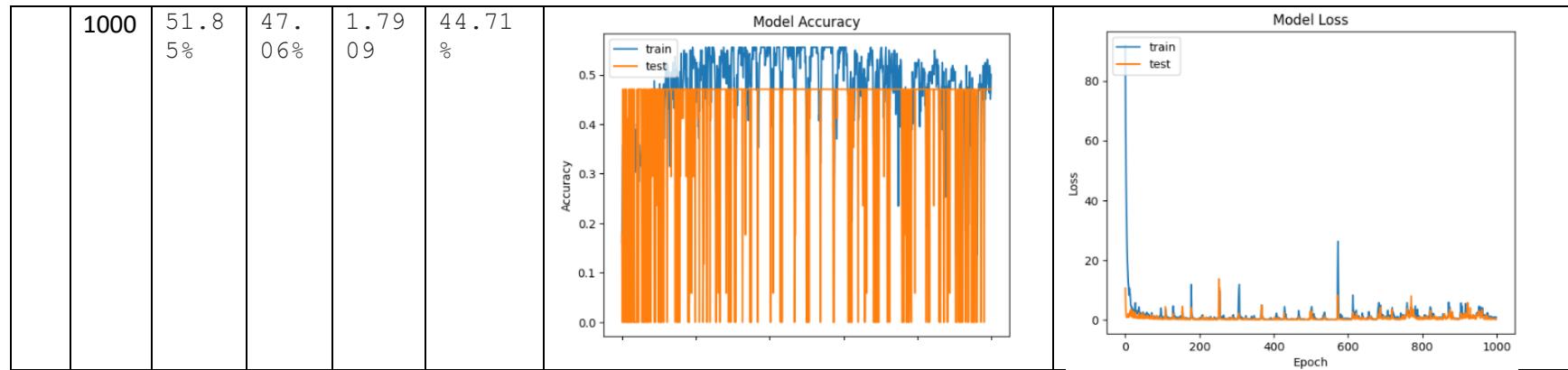




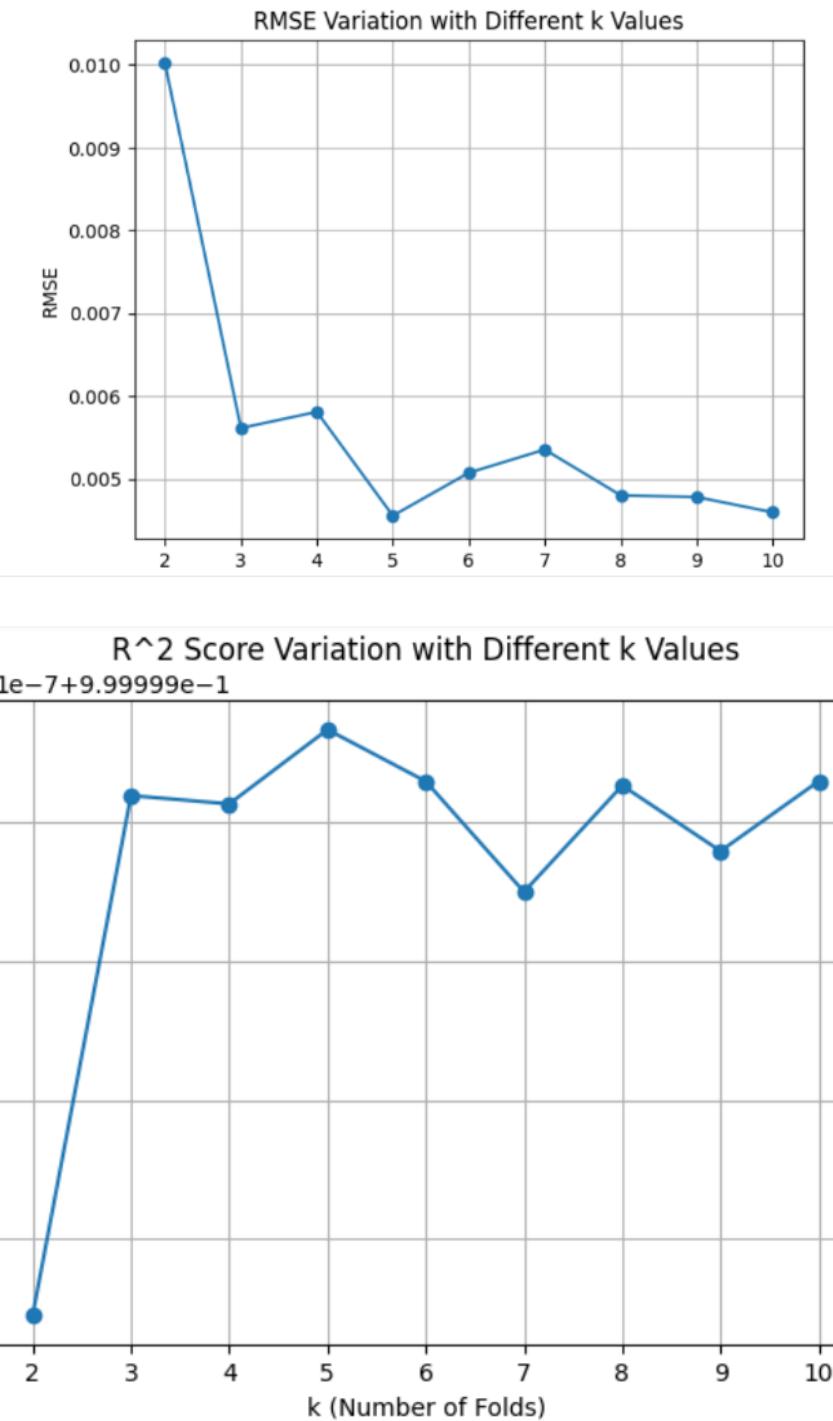








B.2. Hasil Implementasi Support Vector Regression



Lampiran C Aplikasi Web Sistem Prediksi Del Predict

DAFTAR REFERENSI

- 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>
- Ishukatiyar. (2023). *Backpropagation in Data Mining*. GeeksForGeeks. <https://www.geeksforgeeks.org/backpropagation-in-data-mining>
- Itdel. (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).
- Maryana, T., Kusrini, K., & Fatta, H. Al. (2019). Analisis Perbandingan Prediksi Obat Dengan Menggunakan Metode Abc Analisys Dan Svr Pada Aplikasi “Morbis.” *Jurnal Teknologi Informasi*, 3(2), 174. <https://doi.org/10.36294/jurti.v3i2.1016>
- Maulani, H. (2019). *Perbandingan Peramalan Jumlah Penumpang Bus DAMRI menggunakan Metode Artificial Neural Network (ANN) dengan Support Vector Regression*.

- 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>
- 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.
- Nainggolan, K. S., & Lumbanraja, B. (2018). *PREDIKSI CURAH HUJAN DI DELI SERDANG DENGAN BACKPROPAGATION NEURAL NETWORK , DEEP BELIEF NETWORK DAN SUPPORT VECTOR REGRESSION TUGAS AKHIR Diajukan sebagai salah satu syarat untuk memperoleh gelar Sarjana Teknik 11S14008 KWATRI SABATTYAN NAINGGOLAN 11S140.*
- Nasution, D. A., Khotimah, H. H., & Chamidah, N. (2019). Perbandingan Normalisasi Data untuk Klasifikasi Wine Menggunakan Algoritma K-NN. *Computer Engineering, Science and System Journal*, 4(1), 78. <https://doi.org/10.24114/cess.v4i1.11458>
- 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/variansiumm13>
- 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.
- 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>
- Samsudiney. (2019). *Penjelasan Sederhana tentang Apa Itu SVM?* Medium. <https://medium.com/@samsudiney/penjelasan-sederhana-tentang-apa-itu-svm-149fec72bd02>
- 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>
- 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://ejurnal.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>
- Sugara, B., & Subekti, A. (2019). Penerapan Support Vector Machine (Svm) Pada Small Dataset Untuk Deteksi Dini Gangguan Autisme. *Jurnal Pilar Nusa Mandiri*, 15(2), 177–182. <https://doi.org/10.33480/pilar.v15i2.649>
- Syahputra, M. E., Bahri, S., & Rambe, M. F. (2020). Pengaruh Kepemimpinan, Disiplin dan Motivasi Terhadap Kinerja Pegawai Dinas Tarukim Labura. *Pamator Journal*, 13(1), 110–117. <https://doi.org/10.21107/pamator.v13i1.7017>

Trivusi. (2022). *Normalisasi*. Trivusi. <https://www.trivusi.web.id/2022/09/normalisasi-data.html>

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>

