PERBANDINGAN METODE SELEKSI FITUR *FILTER*, *WRAPPER*, DAN *EMBEDDED* PREDIKSI KANDUNGAN VITAMIN C PADA BUAH MANGGA MEGGUNAKAN METODE LINEAR REGRESSION DAN RANDOM FOREST REGRESSION

Disusun Oleh:

N a m a : Dimas Setyawan Ramadhansyah
NIM : 17523152

PROGRAM STUDI INFORMATIKA – PROGRAM SARJANA
FAKULTAS TEKNOLOGI INDUSTRI
UNIVERSITAS ISLAM INDONESIA

2022

HALAMAN PERSEMBAHAN

Alhamdulillahirobbil'alamin, puji syukur ke hadirat Allah SWT atas segala nikmat yang telah diberikan kepada saya dan kita semua, sehingga dapat menyelesaikan Tugas Akhir dengan baik. Terima kasih atas motivasi, dukungan, do'a, perhatian, dan kasih saying kepada semua pihak yang telah ikut serta dalam menyelesaikan laporan Tugas Akhir khususnya kepada kedua orangtua saya yang telah mendidik hingga saat ini.

HALAMAN MOTO

“Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya”

(Q.S Al-Baqarah: 286)

KATA PENGANTAR

Assalamualaikum Warahmatullahi Wabarakatuh

Alhamdulillahirobbil'alamin, puji syukur ke hadirat Allah SWT yang telah melimpahkan rahmat-Nya sehingga penulis mampu untuk menyelesaikan Tugas Akhir dengan baik yang berjudul "Perbandingan Metode Seleksi Fitur *Filter*, *Wrapper*, dan *Embedded* Prediksi Kandungan Vitamin C pada Buah Mangga Menggunakan Metode Linear Regression Dan Random Forest Regression". Laporan ini disusun untuk memenuhi tugas akhir sebagai syarat untuk menyelesaikan Pendidikan pada jenjang Strata (S1) pada Jurusan Informatika Universitas Islam Indonesia. Tidak lupa bahwa banyak sekali pihak yang terlibat dalam penyusunan Tugas Akhir ini, baik berupa dukungan materi, motivasi maupun do'a. Oleh karena itu penulis tidak lupa menyampaikan terima kasih kepada:

1. Allah SWT atas segala nikmat dan rahmat-Nya selama ini
2. Kedua Orang tua yang saya cintai, yang tidak pernah lelah mendoakan, memberikan motivasi, memberi nasehat, dan memberi dukugan penuh
3. Bapak Prof. Fathul Wahid, S.T., M.Sc., Ph.D., sebagai Rektor Universitas Islam Indonesia.
4. Bapak Hendrik, S.T., M.Eng., sebagai Ketua Jurusan Informatika Universitas Islam Indonesia.
5. Bapak Dr. Raden Teguh Dirgahayu, S.T., M.Sc., sebagai Ketua Program Studi Informatika Program Sarjana Universitas Islam Indonesia.
6. Ibu Arrie Kurniawardhani, S.Si, M.Kom., sebagai Dosen Pembimbing Akademik (DPA) sekaligus Dosen Pembimbing Tugas Akhir yang peduli, sabar, serta memberi arahan dalam membantu saya menyelesaikan tugas akhir ini.
7. Bapak dan Ibu dosen Jurusan Informatika yang telah memberikan ilmu bermanfaat kepada penulis
8. Sahabat-sahabat saya, Dimas Ariyoga dan Muhammad Sauqi Khatami yang selalu mendukung saya dari awal kuliah di Universitas Islam Indonesia

Saya menyadari laporan Tugas Akhir ini jauh dari kata sempurna, saya berharap semoga dengan disusunnya laporan Tugas Akhir ini dapat bermanfaat kepada semua pembaca dan semua orang.

Wassalamu'alaikum warahmatullahi wabarakatuh.

Yogyakarta, 15 Juli 2022

(Dimas Setyawan Ramadhansyah)

SARI

Mengolah data yang memiliki dimensi tinggi merupakan suatu tantangan penelitian. Alasannya karena memerlukan waktu komputasi yang lama untuk bisa menyelesaikannya. Data yang memiliki dimensi tinggi juga memiliki kekurangan yang sering disebut *high dimensional data* karena dapat menyebabkan fenomena *Curse of dimensionality*. Fenomena ini menyebabkan pemborosan ruang penyimpanan, kemampuan visualisasi yang buruk, serta terjadi overfitting. Untuk mengatasi masalah itu, penelitian ini akan menggunakan teknik untuk mengurangi fitur yang banyak dengan seleksi fitur. Penelitian ini menggunakan sembilan metode berbeda yang dapat dikategorikan menjadi tiga kategori seleksi fitur, yaitu *Filter, Wrapper, dan Embedded*. Untuk data yang diuji adalah *dataset* NIRS mangga yang terkenal dengan banyaknya fitur didalamnya. Setelah berhasil di seleksi fitur, data kemudian akan dilakukan prediksi menggunakan metode regresi untuk mengetahui kandungan vitamin c pada mangga menggunakan dua metode berbeda, yaitu *Linear Regression* dan *Random Forest Regression*. *Metode Random Forest Regression* akan dilakukan dengan tiga skenario menggunakan tiga *trees* yang berbeda untuk dibandingkan performanya. Hasil yang diperoleh dari prediksi vitamin c pada mangga berbeda tergantung pada model regresi dan seleksi fitur. Untuk prediksi tanpa melakukan seleksi fitur, *Linear Regression* mendapatkan nilai performa yang lebih baik dibandingkan *Random Forest Regression* dengan nilai pengujian 187.48 MSE, 13,42 RMSE, 10,89 MAE, dan R^2 -0.17. Sedangkan untuk metode seleksi fitur *Fisher Score* mendapatkan nilai pengujian performa terbaik di antara delapan metode lainnya setelah di prediksi menggunakan *Linear Regression* dengan nilai pengujian 132.19 MSE, 11.5 RMSE, 9.51 MAE, dan 023 R^2.

Kata kunci: *Linear Regression, Random Forest Regression, Filter, Wrapper, Embedded,* Regresi, Seleksi Fitur, *Wrapper*

GLOSARIUM

Dataset	Kumpulan data yang digunakan untuk melatih model.
Library	Kumpulan kode dengan fungsi tertentu.
Seleksi Fitur	Proses pemilihan fitur terpenting berdasarkan atribut fitur.
Spektrum	Urutan atau rangkaian yang bersinambung.
NIRS	*Near-infrared Spectroscopy*.

DAFTAR ISI

DAFTAR TABEL

DAFTAR GAMBAR

BAB I
PENDAHULUAN

1.1 Latar Belakang

Analisis regresi merupakan sebuah teknik Statistika yang secara matematis dapat digunakan untuk membuat prediksi nilai dari suatu variabel output yang bersifat *continuous* dari sejumlah variabel input yang bersifat independen dengan menggunakan pendekatan dari menganalisis hubungan antar variabel yang kompleks (Yang et al., 2016). Pada pembangunan model regresi tradisional, *dataset* yang digunakan memiliki fitur atau variabel penjelas yang lebih sedikit daripada jumlah observasi. Namun pada masa modern dimana data dapat dengan mudah dikumpulkan dalam jumlah besar, *dataset* yang digunakan dalam banyak masalah regresi seringkali memiliki jumlah variabel penjelas (p) yang melebihi jumlah observasi (n) atau $p > n$, data seperti ini dikenal sebagai *high dimensional data*. Jenis *dataset* baru seperti ini menimbulkan tantangan baru untuk model regresi karena telah diketahui bahwa pemodelan regresi yang dilakukan menggunakan variabel penjelas (fitur) dalam jumlah besar dengan ukuran sampel terbatas atau *high dimensional data* merupakan tugas yang sulit karena dapat menyebabkan terjadinya *Curse of dimensionality* (Filzmoser & Nordhausen, 2021).

Curse of Dimensionality, yang diperkenalkan oleh Richard Bellman pada tahun 1957 ini mengacu pada sifat ledakan spasial dari dimensi data dan efek yang dihasilkannya, seperti peningkatan eksponensial dalam upaya komputasi, pemborosan ruang penyimpanan yang besar, kemampuan visualisasi yang buruk, serta memungkinkan terjadinya *overfitting*. Tingginya jumlah dimensi ini secara teoritis memungkinkan lebih banyak informasi untuk disimpan, tetapi secara praktis jarang membantu karena semakin tinggi dimensi data, kemungkinan noise dan redundansi juga semakin tinggi (Venkat, 2018). Sebagai contoh, (Frénay et al., 2013) dalam penelitiannya memberikan contoh bahwa *Near-infrared Spectroscopy* (NIRS) merupakan jenis *dataset* yang bersifat *high dimensional*. Setiap sampel dari NIRS digambarkan oleh puluhan atau ratusan fitur sesuai dengan komponen spektrumnya. Sebagian besar fitur dari NIRS ini dalam praktiknya bersifat berlebihan atau tidak relevan dengan masalah regresi yang dipertimbangkan sehingga dibutuhkan sebuah solusi dalam mengolah datanya (Yuhua et al., 2013).

Untuk mengatasi masalah *Curse of Dimensionality* pada data, penelitian ini akan menggunakan sebuah teknik yaitu *dimensionality reduction* terhadap *dataset* NIRS yang

digunakan. Salah satu metode dari *dimensionality reduction* yang paling umum digunakan yaitu *feature selection* atau seleksi fitur. Seleksi fitur yang diimplementasikan pada *dataset* dapat mengubah *dataset* asli yang memiliki dimensi tinggi menjadi dataset baru yang memiliki dimensi rendah sambil mempertahankan sebanyak mungkin informasi asli dari data. Saat jumlah dimensi pada data berkurang, maka ruang penyimpanan data dapat dikurangi, waktu komputasi menjadi lebih sedikit serta data yang berlebihan, tidak relevan, dan *noise* pada data dapat dihilangkan. Selain itu, penerapan seleksi fitur pada dataset yang digunakan juga dapat meningkatkan performa prediksi, menghindari *overfitting*, dan mengurangi waktu eksekusi dan pelatihan dari model regresi yang dibangun (Zebari et al., 2020).

Diterapkannya teknik *dimensionality reduction* yaitu seleksi fitur pada penelitian ini diharapkan dapat meningkatkan performa prediksi vitamin C dari model regresi yang dibangun menggunakan data *Near-infrared Spectroscopy* (NIRS).

1.2 Rumusan Masalah

Rumusan masalah pada penelitian ini adalah sebagai berikut:

a. Bagaimana pengaruh seleksi fitur terhadap prediksi vitamin c pada mangga?
b. Bagaimana performa model *Linear Regression* dan *Random Forest Regression* menggunakan beberapa seleksi fitur berbeda?

1.3 Batasan Masalah

a. Data berupa spektrum NIRS mangga dengan Panjang gelombang 1000 – 2500 nm
b. Fitur yang digunakan pada penelitian dibatasi hingga 100 fitur.

1.4 Tujuan Penelitian

Tujuan dari penelitian ini adalah menerapkan metode seleksi fitur dan metode regresi terhadap prediksi vitamin c pada mangga.

1.5 Manfaat Penelitian

Manfaat dari penelitian ini adalah untuk mengetahui proses seleksi fitur yang diterapkan menggunakan metode regresi, mengetahui efektivitas metode seleksi fitur pada data NIRS dan mengetahui seleksi fitur yang memiliki performa terbaik untuk digunakan sebagai metode seleksi fitur data NIRS.

1.6 Sistematika Penulisan

Dalam penyusunan laporan tugas akhir ini, sistematika penulisan dibagi menjadi beberapa bab sebagai berikut:

Bab I Pendahuluan, Bab ini berisi latar belakang mengenai permasalahan aktual yang mendasari penelitian. Bagian ini memuat latar belakang, rumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penyusunan.

Bab II Landasan Teori, Bab ini berisi uraian literatur yang sesuai topik penelitian sebagai dasar melakukan penelitian. Bab ini juga mengkaji penelitian-penelitian sebelumnya yang relevan dengan laporan penelitian ini.

Bab III Metodologi Penelitian, Bab ini berisi tahapan-tahapan dalam melakukan penelitian untuk mencapai solusi dalam permasalahan. Bab ini terdiri dari pengumpulan data, implementasi model, pengujian, dan analisis.

Bab IV Hasil dan Pembahasan, Bab ini menjelaskan tentang hasil implementasi seleksi fitur pada hasil prediksi vitamin c pada mangga. Pada bagian ini juga dilakukan pengujian untuk mendapatkan sebuah hasil yang diharapkan.

Bab V Kesimpulan dan Saran, Bab ini berisi kesimpulan mengenai hasil yang telah didapatkan dari penelitian. Bagian ini juga berisi saran untuk penelitian-penelitian selanjutnya.

BAB II
LANDASAN TEORI

2.1 *Near Infrared Reflectance Spectroscopy* (NIRS)

Near Infrared merupakan sebuah teknik atau metode yang digunakan untuk menganalisis data yang berupa komposisi kimia dari bahan-bahan organik seperti buah-buahan. Informasi mengenai kandungan-kandungan kimia dari bahan organik ini didapatkan berdasarkan reaksi biologi setelah produk organik tersebut diberi radiasi sinar NIR (Devianti et al., 2019). Sementara itu, *Spectroscopy* merupakan sebuah disiplin ilmu yang menjelaskan interaksi dari radiasi elektromagnetik dengan atom dan molekul (Suarsa, 2015). *Spectroscopy* ini umumnya digunakan untuk mengidentifikasi kandungan-kandungan dari suatu objek dengan menganalisis spektrum yang dipancarkan atau yang diserap. Sinar yang digunakan oleh NIRS ini tersedia pada jangkauan panjang gelombang dari 780 nm hingga 2500 nm yang berarti gelombangnya berada di atas sinar yang *visible (Mechram et al., 2021)*. Ketika sebuah benda organik dipaparkan oleh sebuah sinar, maka objek tersebut akan memberi respon berupa pantulan (*reflectance*), serapan (*absorbance*), dan terusan (*transmittance*). Respon inilah yang akan digunakan untuk melihat komposisi beda organik (Munawar et al., 2016).

2.2 Vitamin C

Vitamin C adalah salah satu zat gizi organik yang sangat bagus dan dibutuhkan oleh tubuh manusia untuk memelihara fungsi metabolisme. Selain itu, Vitamin C juga berperan penting sebagai antioksidan pada tubuh, mempercepat penyembuhan bagian tubuh yang sakit atau rusak, pembentukan protein kolagen serta menurunkan kadar kolesterol di dalam darah manusia (Hasanah, 2018). Namun Vitamin C tidak dapat disintesis secara mandiri di dalam tubuh kita, sehingga asupan vitamin C dari luar tubuh sangat dibutuhkan. Vitamin C ini sendiri banyak terkandung bersama zat-zat atau vitamin lainnya di dalam bahan makanan seperti buah dan sayur-sayuran. Salah satu contoh dari buah yang mengandung vitamin C dan sangat disukai oleh masyarakat adalah buah mangga (Mulyati, 2021).

2.3 Buah Mangga

Buah mangga merupakan tanaman yang populer, digemari, dan banyak dibudidayakan di Indonesia. Selain memiliki rasa yang khas, di Indonesia sendiri mangga dapat tumbuh

dengan baik di sebagian besar daerah seperti daerah dataran rendah yang berhawa panas, pohon mangga juga masih ditanam sampai dataran tinggi yang berhawa sedang (Rahman et al., 2015). Buah mangga juga sangat bagus dikonsumsi karena zat-zat baik yang terkandung di dalamnya. Buah mangga ini terdiri dari 80% air, 15% gula, dan sisanya terdiri dari vitamin A, vitamin B, dan vitamin C serta senyawa-senyawa lainnya. Salah satu kandungan vitamin yang paling banyak dalam buah mangga adalah vitamin C dan gula buah (fruktosa) (Yuliati & Kurniawati, 2017).

2.4 Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) merupakan langkah yang akan pertama kali dilakukan dalam sebuah penelitian atau pekerjaan yang berkaitan dengan data, hal ini agar peneliti mampu memahami dengan baik data yang akan digunakan dalam pekerjaannya. EDA adalah sebuah teknik untuk mendapatkan wawasan atau pengetahuan (*insight*) dari sebuah dataset, biasanya EDA dilakukan dengan menggunakan metode-metode *graphical* (secara grafis atau visual) yang mampu memperlihatkan asosiasi kompleks, pola, dan anomali di dalam dataset pada resolusi atau ukuran yang berbeda-beda (Singh, 2020). Walaupun kebanyakan teknik EDA bersifat grafis, terdapat juga beberapa teknik EDA yang bersifat kuantitatif. Alasan ketergantungan yang tinggi pada grafik adalah bahwa pada dasarnya peran utama EDA adalah untuk mengeksplorasi data sehingga grafik mempunyai peran yang tinggi dalam memberikan informasi kepada para analis (Komorowski et al., 2016). Ditambah lagi dengan pemaparan dari Samosir et al. (2021) yang menyatakan bahwa data statistik yang ditampilkan secara numerik saja dapat mengaburkan, menyembunyikan, atau bahkan salah dalam merepresentasikan struktur data sehingga dapat mengakibatkan penarikan kesimpulan yang salah.

EDA sendiri tidak bergantung hanya pada suatu tipe model atau prosedur baku yang sudah didefinisikan sebelumnya. Hal ini dikarenakan EDA memiliki karakteristik fleksibel yang diperlukan untuk melakukan identifikasi dan investigasi suatu fenomena yang muncul pada saat melakukan penelitian empiris (Ariyoga, 2022). Samosir et al. (2021) menjelaskan bahwa memahami kondisi dataset pada EDA dapat merujuk pada setidaknya empat poin berikut:

1. Mengekstrak variabel penting dan membuang variabel yang tidak berguna.
2. Mengidentifikasi outliers, nilai yang kosong atau hilang (missing values), dan kesalahan manusia (*human error*).

3. Memahami hubungan antar variabel.
4. Memaksimalkan pengetahuan yang kita miliki terhadap kondisi data dan meminimalkan potensi kesalahan di kemudian hari.

2.5 Analisis Regresi

Analisis regresi merupakan sebuah teknik Statistika yang secara matematis dapat digunakan untuk menguji hubungan antara variabel yang mempengaruhi (bebas) terhadap suatu variabel yang dipengaruhi (terikat) (Ramadhani, 2022). Dalam *Data Mining* sendiri, analisis regresi adalah sebuah model komputasi yang digunakan untuk membuat prediksi nilai dari suatu variabel output yang bersifat *continuous* dari sejumlah variabel input yang bersifat independen dengan pendekatan dari menganalisis hubungan antar variabel yang kompleks. Beberapa metode regresi telah banyak diusulkan untuk menyelesaikan masalah-masalah regresi berdasarkan keberhasilan metodologinya yaitu seperti Linear Regression, Suppor*t Vector Regression, Neural Network, Piecewise Regression,* dan sebagainya (Yang et al., 2016).

Menurut (Wagschal, 2016), tiga fungsi utama dalam menggunakan analisis regresi yaitu:

1. Menunjukkan apakah variabel independen memiliki hubungan yang signifikan dengan variabel dependen.
2. Menunjukkan pengaruh atau kekuatan relatif dari variabel independen yang berbeda terhadap suatu variabel dependen.
3. Membuat model prediksi.

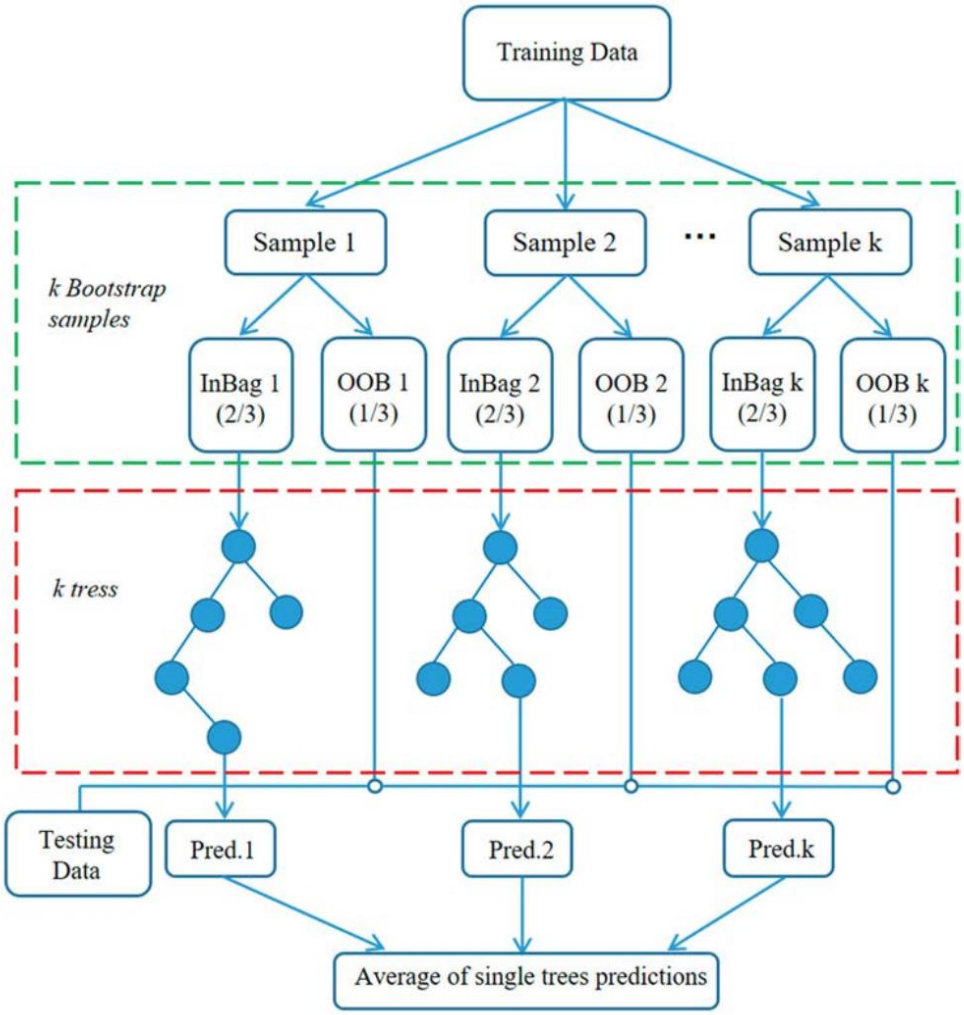
Pada penelitian ini, akan digunakan dua model regresi yaitu *Random Forest Regression* dan *Linear Regression*.

2.5.1 Random Forest Regression

Algoritma *Random Forest Regression* merupakan sebuah metode pembelajaran regresi bersifat *Ensemble* yang diusulkan oleh Leo Breiman (Breiman, 2001). *Ensemble* sendiri merupakan model yang terdiri dari beberapa algoritma, sehingga dalam *Random Forest* ini terkandung beberapa *decision tree* dengan distribusi yang sama dan dengan kondisi tidak berkorelasi satu sama lain untuk membangun *forest* atau "hutan" dengan tujuan melatih dan memprediksi data sampel (Zhang et al., 2021). *Random Forest Regression* dapat mengandung ratusan atau bahkan ribuan *Decision Tree* yang bertindak sebagai fungsi regresi sendiri. Setiap

tree di RFR ditanam dengan *subset* prediktor acak maka dari itu algoritma ini disebut hutan 'acak' (Zhang et al., 2021).

Decision Tree atau yang disebut juga sebagai *Classification and Regression Tree* (CART) adalah sebuah model statistik nonparametrik yang dapat menggambarkan hubungan antara variabel. Variabel di sini merupakan variabel respon (dependen) dengan satu atau lebih variabel prediktor (independen). Setiap pohon keputusan terdiri dari simpul keputusan atau internal node dan simpul daun atau leaf. Setiap node keputusan mengevaluasi setiap sampel variabel atau atribut dan setiap cabangnya merupakan hasil dari pengujian tersebut, sementara itu node terluar yaitu daun menjadi labelnya [DT jelasin]. Berbeda dari hasil akhir *Random Forest Classifier* yang mengambil *Majority Votes* dari berbagai *Decision Tree* sebagai hasilnya, hasil akhir dari Random Forest Regression ini merupakan nilai rata-rata dari keluaran semua *Decision Tree* yang telah dibangun (Li et al., 2018). Gambar 2.1 akan menunjukkan cara kerja dari *Random Forest Regression*.



Gambar 2.1. Diagram alur *Random Forest Regression*

Sumber: (Zhang et al., 2021)

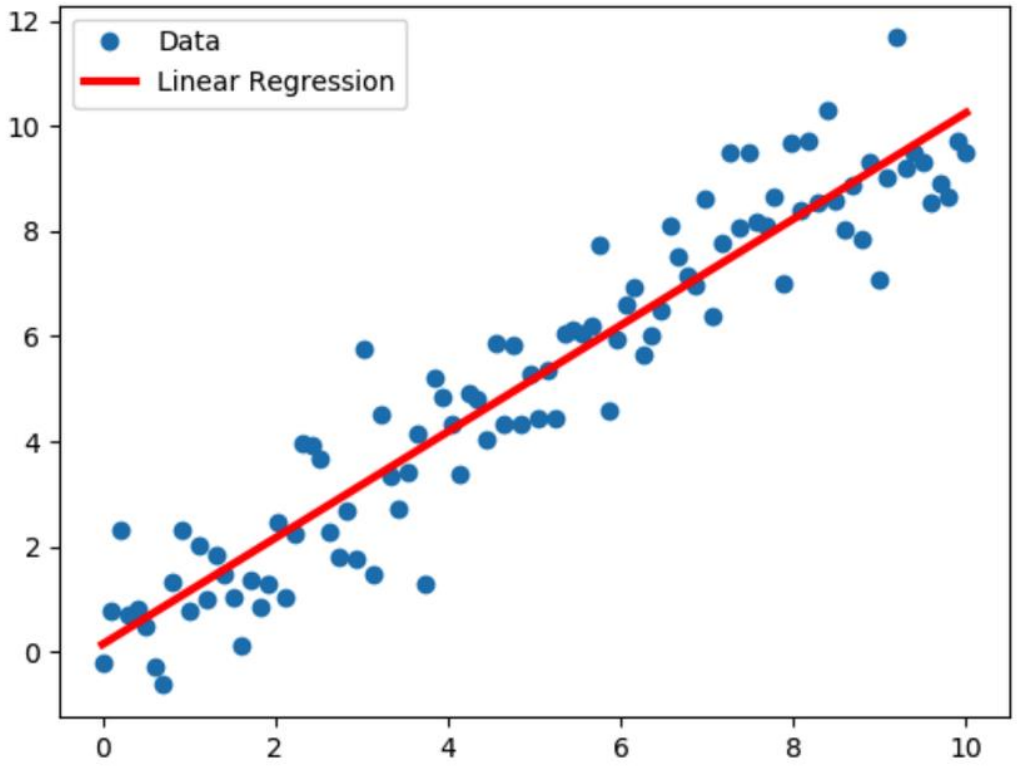
Seperti yang terlihat pada Gambar 2.2, Untuk mendapatkan model *ensemble* dengan kemampuan generalisasi yang kuat, *base learner* atau pohon regresi di dalam model harus dibuat seunik mungkin. Oleh karena itu, digunakan sebuah metode bernama *Bagging* (*bootstrap aggregating*) untuk melakukan *resampling* pada data. Untuk kumpulan data yang berisi *k* sampel, sampel secara acak diambil ke dalam kumpulan sampel. Setelah itu, sampel dikembalikan ke data awal, yang memungkinkan sampel untuk dipilih pada pengambilan sampel berikutnya. Jadi, setelah *k* operasi pengambilan sampel acak, diperoleh satu set yang berisi *k* sampel yang nantinya akan digunakan untuk melatih setiap *Decision Tree* berdasarkan setiap set sampel. Terakhir, akan diambil nilai rata-rata dari setiap regresi *Decision Tree*.

2.5.2 Linear Regression

Linear Regression merupakan salah satu jenis regresi yang melibatkan hubungan antara satu variabel dependen atau variabel tak bebas atau label (Y) dan variabel independen atau variabel bebas atau fitur (X). Besarnya nilai variabel dependen akan dipengaruhi oleh besar kecilnya variabel bebas. Algoritma *Linear Regression* didasarkan pada pola hubungan data terkait masa lalu (Hamdanah & Fitrianah, 2021). Umumnya, algoritma *Linear Regression* dibagi menjadi dua jenis, yaitu *Simple Linear Regression* dan *Multiple Linear Regression*. *Simple Linear Regression* merupakan hubungan antara satu variabel dependen dengan satu variabel independen, sedangkan *Multiple Linear Regression* merupakan hubungan antara satu variabel dependen dengan dua atau lebih variabel independen (Herwanto et al., 2019). Pada penelitian ini, akan digunakan *Multiple Linear Regression* yang diekspresikan melalui Persamaan 2.1.

$$Y = a + a_1X_1 + b_2X_2 + ..+b_nX_n \tag{2.1}$$

Terdapat beberapa komponen pada persamaan 2.1, di antaranya adalah Y yang merupakan variabel dependen atau nilai yang diprediksikan, a adalah konstanta, X_n adalah variabel independen, dan b_n adalah koefisien regresi. Dari persamaan ini, dapat ditarik sebuah garis yang mampu memprediksi variabel dependen berdasarkan variabel independen. Algoritma *Multiple Linear Regression* akan berusaha untuk menemukan garis prediksi terbaik (Gupta et al., 2020). Kualitas garis prediksi dapat ditentukan dari seberapa dekatnya garis prediksi dengan poin-poin data nilai variabel dependen. Berikut Gambar 2.2 yang menggambarkan bagaimana bentuk garis *Linear Regression*.



Gambar 2.2. Contoh garis *Linear Regression*

Sumber: (Tran, 2019)

Pada Gambar 2.2, garis yang berwarna merah merupakan garis *Linear Regression* dan poin-poin berbentuk bulat berwarna biru merupakan poin data dari variabel independen. Sebuah garis *Linear Regression* dapat dikatakan bagus apabila garis mempunyai jarak yang dekat dengan keseluruhan poin-poin data variabel independen. Semakin dekat garis *Linear Regression* dengan poin-poin data variabel independen, maka akan semakin bagus juga prediksi yang dihasilkan.

2.6 Seleksi Fitur

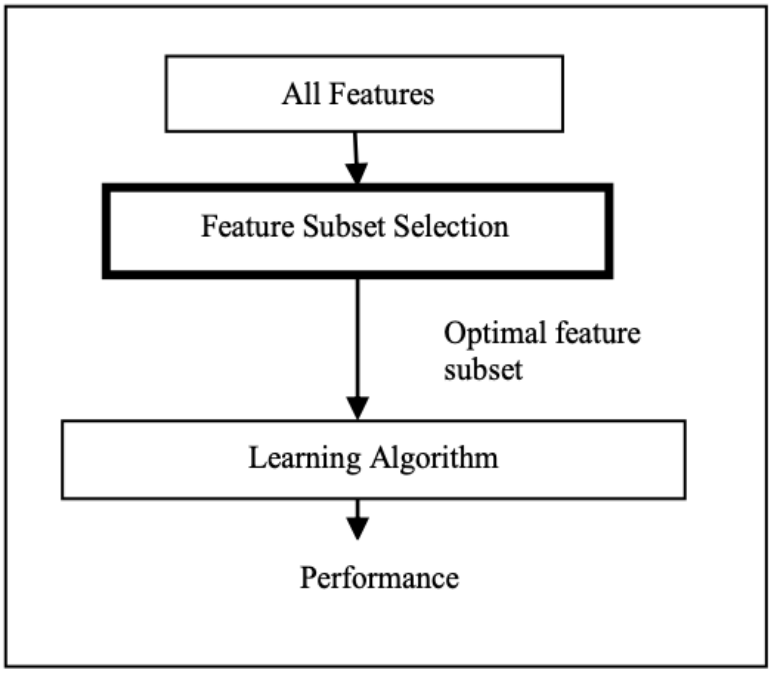
Seleksi fitur merupakan salah satu teknik *dimensionality reduction* (pengurangan dimensi) untuk sebuah *set* fitur. Teknik seleksi fitur ini bekerja dengan cara memilih *subset* yang relevan dari seluruh fitur pada *dataset* dan menghapus fitur-fitur yang tidak relevan, berlebihan, dan bersifat *redundant*. Tujuan digunakannya seleksi fitur ini mengarah pada peningkatan kinerja pembelajaran dari model prediksi yang dibangun, yaitu meningkatkan

performa model, mengurangi *computational cost* atau waktu komputasi, dan interpretabilitas model yang lebih baik (Wang et al., 2020).

Berdasarkan kombinasinya dengan konstruksi model pembelajaran, teknik seleksi fitur dapat diklasifikasikan menjadi tiga kategori yaitu kategori *Filter*, *Wrapper*, dan *Embedded* (Sainin & Alfred, 2011). Diantara tiga kategori seleksi fitur ini, kategori *Wrapper* dan *Embedded* banyak dihindari karena membutuhkan sumber daya komputasi yang besar (Lamba et al., 2018).

2.6.1 Filter Method

Seleksi fitur filter adalah jenis kategori seleksi fitur yang paling umum digunakan Kategori ini melakukan pemilihan fitur tanpa melibatkan algoritma pembelajaran. Pada dasarnya, metode filter menggunakan kriteria Statistika seperti mengevaluasi jarak, informasi yang dikandung, ketergantungan antar fitur, dan konsistensi untuk menghitung dan menetapkan skor terhadap fitur-fitur yang terdapat dalam *training set* dan kemudian melakukan *ranking* atau pengurutan terhadap setiap fitur berdasarkan skor yang dihitung. Fitur-fitur yang bersifat informatif biasanya mendapatkan skor yang lebih tinggi dan fitur yang tidak informatif mendapatkan skor yang lebih rendah (Rajab & Wang, 2020). Gambar 2.3 akan menunjukkan proses kerja dari seleksi fitur kategori *filter* secara umum.



Gambar 2.3. Alur kerja seleksi fitur kategori *Filter*

Sumber: (Sahu et al., 2018)

Berdasarkan Gambar 2.3, ditunjukkan bahwa kategori *Filter* melakukan proses seleksi fitur tanpa melibatkan *learning algorithm*. Pada kategori ini, metode seleksi fitur yang digunakan akan memilih fitur-fitur yang *optimal* dari *dataset*. Selanjutnya, *subset* fitur hasil seleksi yang telah diurutkan berdasarkan peringkat pada skor yang telah dihitung dapat digunakan sebagai data *input* untuk pembangunan model dan kemudian performa modelnya dapat dinilai (Sahu et al., 2018). Pada penelitian ini, akan diterapkan empat metode dari seleksi fitur kategori *Filter*, yaitu *Mutual Information*, *Fisher Score*, *Analysis of Variance* (ANOVA), dan *Pearson Correlation*.

Mutual Information (MI)

Mutual Information merupakan indeks dari ketergantungan statistik antara dua variabel acak. Berbeda dari indeks lainnya seperti koefisien pada korelasi *Pearson*, seleksi fitur MI juga mampu menangkap ketergantungan non-linier dan invarian pada variabel (Beraha et al., 2019). Nilai *Mutual Information* (MI) antara dua variabel acak adalah non-negatif, nilai nol diperoleh jika dan hanya jika dua variabel acak bersifat independen. Semakin tinggi nilai MI yang diperoleh berarti nilai ketergantungan antar variabel lebih tinggi (Zhao et al., 2016). Nilai dari *Mutual Information* dapat ditemukan menggunakan Persamaan 2.2.

$$I(X\,;\,Y) = H(X) - H(X\,|Y) \tag{2.2}$$

Keterangan:

$I(X\,;\,Y)$ = *Mutual Information* yang bersifat *mutual* untuk X dan Y.

$H(X)$ = Entropi untuk X, dan

$H(X\,|Y)$ = Nilai entropi bersyarat untuk X yang diberikan Y.

Fisher Score

Fisher Score merupakan salah satu seleksi fitur yang berbasis pada *ranking* dari ratio pada masing-masing variabel prediktor. *Fisher Score* sendiri bertujuan untuk menghapus fitur-fitur yang tidak relevan dan bersifat *redundan*. Nilai dari *fisher score* pada fitur berdasarkan nilai mean dan varian fitur tiap kelas. Semakin tinggi nilai *fisher score* suatu fitur, semakin baik dalam membedakan objek antar kelas (Hsu et al., 2011). *Fisher score* pada fitur dapat ditemukan menggunakan Persamaan 2.3

$$FScore_r = \frac{\sum_{i=1}^{c} n_i(\mu_r^i - \mu_r)^2}{\sum_{i=1}^{c} n_i(\sigma_r^i)^2} \quad (2.3)$$

Keterangan:

n_i menunjukkan jumlah sampel pada kelas ke i.

μ_r^i merupakan nilai rata-rata fitur ke$-r$ pada kelas ke$-i$.

σ_r^i merupakan nilai varian fitur $ke-r$ pada kelas $ke-i$.

μ_r merupakan nilai rata-rata fitur ke r.

***Analysis of Variance* (ANOVA)**

Seleksi fitur ANOVA atau *Analysis of Variance* adalah metode seleksi fitur yang membandingkan rata-rata dari dua fitur. Perbandingan ini dilakukan untuk mengetahui apakah terdapat perbedaan pada fitur-fitur yang dibandingkan. Seleksi fitur ANOVA akan melakukan perbandingan untuk mengetahui apakah data yang dibandingkan merupakan data yang mirip. Pada seleksi fitur ANOVA, *null hypothesis* (Ho) menduga bahwa varians antar data parameter masukan tidak signifikan. Pengujian untuk ANOVA disebut sebagai uji-F. Uji-F mengindikasikan perbedaan antara dua atau lebih fitur (Mazlan et al., 2020). Persamaan 2.4 menunjukkan perhitungan uji-F.

$$F = \frac{Variance\ between\ the\ sample\ (MSR)}{Variances\ within\ the\ sample\ (MSE)} \quad (2.4)$$

Pada Persamaan 2.4, *Variance between the sample* adalah nilai varians antara populasi atau *Mean Square Regression* (MSR) sedangkan *Variances within the sample* merupakan nilai varians di dalam populasi.

Pearson Correlation

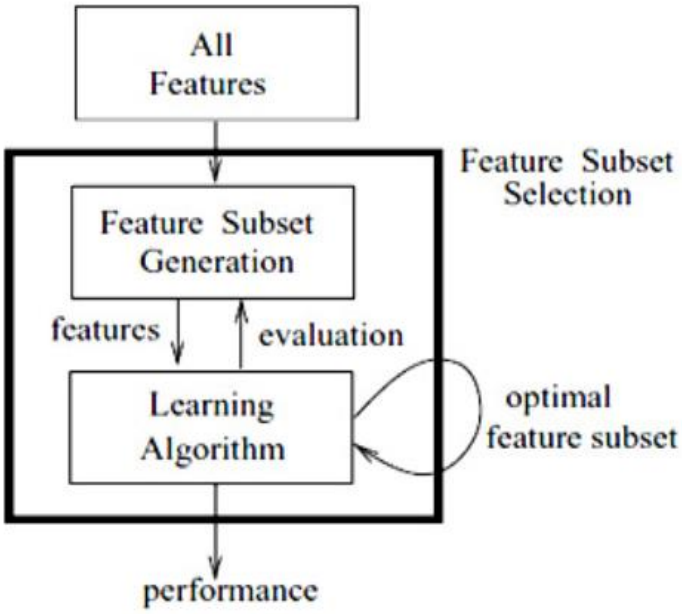
Pearson Correlation adalah seleksi fitur yang mengukur kekuatan hubungan atau hubungan linier antara dua variabel (Samuels, 2014). Nilai yang dihasilkan dari seleksi fitur *Pearson Correlation* terletak pada [-1;1], di mana nilai -1 berarti korelasi negatif sempurna yaitu jika satu variabel meningkat, yang lainnya menurun, nilai +1 berarti korelasi positif sempurna, dan 0 berarti tidak ada korelasi linier antara kedua variabel tersebut (Rozy et al., 2018). Nilai *Pearson Correlation* untuk mendapatkan fitur – fitur yang relevan dilihat berdasarkan nilai *Pearson's* tertinggi (Romadloni & Pardede, 2019). Nilai *Pearson Correlation* dapat dihitung menggunakan Persamaan 2.5.

$$r_{xy} = \frac{n\sum x_i y_i - \sum x_i \sum y_i}{\sqrt{n\sum x_i^2 - (\sum x_i)^2}\sqrt{n\sum y_i^2 - (\sum y_i)^2}} \quad (2.5)$$

Pada Persamaan 2.5, r merupakan nilai *Pearson Correlation*, n adalah *sample size*, dan $x_{i,}y_i$ adalah titik sampel individu yang diindeks dengan i.

2.6.2 *Wrapper Method*

Seleksi fitur dari kategori *Wrapper* menggunakan sebuah algoritma dari *Machine Learning* sebagai *black box evaluator* untuk menemukan *subset* yang berisi fitur-fitur terbaik dari keseluruhan *dataset* yang digunakan. Oleh karena itu, hasil dari kategori *Wrapper* ini sangat bergantung pada algoritma pembelajaran yang digunaka (Effrosynidis & Arampatzis, 2021). Biasanya metode-metode dari seleksi fitur kategori *Wrapper* dapat menghasilkan *subset* fitur yang potensial dengan performa tinggi karena kecocokan dari *subset* tersebut dengan *learning algorithm* yang digunakan (Rahat Hossain et al., 2013). Gambar 2.4 akan menunjukkan proses kerja dari seleksi fitur kategori *Wrapper* secara umum.



Gambar 2.4. Alur kerja seleksi fitur kategori *Wrapper*

Sumber: (Sainin & Alfred, 2011).

Berdasarkan Gambar 2.4, diperlihatkan bahwa dalam pendekatan kategori *Wrapper*, semua fitur akan diambil dan diuji kombinasinya dengan *Learning Algorithm* sebagai fungsi evaluasinya. Proses ini akan terus diulang hingga dihasilkan *subset* fitur yang berisi kombinasi fitur-fitur yang relevan hasil pengujian. Metode-metode dari *Wrapper* akan mencari kombinasi fitur terbaik dengan menghitung perkiraan performa dari *Learning Algorithm* yang digunakan untuk setiap fitur yang akan ditambahkan atau dihapus dari *subset* fitur. Karena *Wrapper* akan mencari secara acak untuk semua kemungkinan dari *subset* terbaik, kategori ini membutuhkan waktu komputasi yang lebih tinggi dari dua kategori lainnya (Wang et al., 2017). Pada penelitian ini, akan diterapkan tiga metode dari seleksi fitur kategori *Wrapper*, yaitu *Sequential Forward Selection* (SFS), *Backward Elimination*, dan *Recursive Feature Selection* (RFE).

***Sequential Forward Selection* (SFS)**

Metode *Sequential Forward Selection* (SFS) merupakan metode seleksi fitur dari kategori *Wrapper* yang berarti membutuhkan sebuah *Learning Algorithm* dalam mencari *subset* terbaik dari keseluruhan fitur yang berisi kombinasi fitur dengan jumlah tertentu. Teknik seleksi fitur SFS ini berjalan dengan strategi pencarian sekuensial untuk menemukan subset

fitur terbaik. Seleksi fitur ini memiliki parameter k yang memungkinkannya berhenti ketika ada k fitur dalam subset fitur yang dipilih (Anukrishna & Paul, 2017). Untuk langkahnya, Metode SFS ini dimulai dengan set fitur kosong kemudian relevansi setiap fitur akan dievaluasi menggunakan *Learning Algorithm* berdasarkan kualitas fitur dengan nilai tertinggi yang dapat menghasilkan kinerja atau performa terbaik ditambahkan ke kumpulan fitur. Prosedur ini akan terus berlanjut dengan penambahan fitur satu demi satu hingga sejumlah fitur yang telah ditentukan sebelumnya terpilih (Chandra, 2015).

Backward Elimination (BE)

Metode seleksi fitur Backward Elimination (BE) dimulai dengan menggunakan keseluruhan set fitur yang lengkap (variabel prediktor) dan kemudian secara bertahap menghapus fitur-fitur yang tidak relevan (Rahat Hossain et al., 2013). Fitur-fitur yang tidak relevan ini akan dieliminasi dengan melihat p-value. Pada langkah pertama, p-value dari untuk semua variabel prediktor akan dihitung, dan variabel dengan p-value terbesar yang melebihi nilai critical p-value atau threshold yang telah ditentukan akan dihapus. Kemudian, p-value akan dihitung kembali untuk variabel-variabel yang tersisa, variabel dengan p-value tertinggi yang melebihi critical p-value akan dihapus. Proses ini akan terus diulang hingga nilai p-value tertinggi dari suatu variabel kurang dari nilai critical p-value, hal ini menunjukkan bahwa variabel yang sesuai tidak berlebihan dengan adanya variabel lain dalam model (Haque et al., 2018).

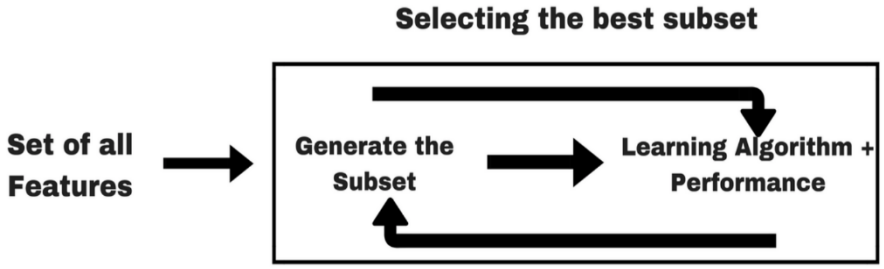
***Recursive Feature Elimination* (RFE)**

Recursive Feature Elimination (RFE) merupakan salah satu metode dari *Wrapper* yang bertujuan untuk mencari jumlah fitur yang optimal dengan menghilangkan fitur yang paling tidak penting. Metode ini melakukan proses seleksi fitur dengan menggunakan algoritma *Machine Learning* sehingga hasilnya tergantung dengan kinerja keberhasilan algoritma yang digunakan. RFE ini menghilangkan atau menghapus fitur yang tidak perlu, lemah, dan paling tidak mempengaruhi keberhasilan model dan pada saat yang sama mempertahankan fitur yang efektif dan kuat (Akkaya, 2021). Dalam penerapannya, RFE menggunakan prosedur iteratif yang bekerja mirip dengan seleksi fitur *Backward Elimination*. Metode ini awalnya membuat model pada seluruh set fitur dan menilai setiap fitur sesuai dengan efek dan kepentingannya pada variabel target. Setelah itu, ia membangun kembali model setelah menghapus fitur yang

paling tidak penting pada setiap langkah dan menghitung ulang pentingnya setiap fitur hingga keberhasilan model tertinggi tercapai (Le Thi et al., 2008).

2.6.3 *Embedded Method*

Ide dari metode-metode dari kategori *Embedded* berdasar pada penggabungan dua kategori seleksi fitur sebelumnya yaitu *Filter* dan *Wrapper*. Kategori ini menggunakan sebuah *Learning Algorithm* sebagai pendekatan *Wrapper* terhadap fitur-fitur yang dipilih dengan pendekatan *Filter*. Atas dasar Ini pula, kompleksitas dari komputasinya tidak akan sebesar kateogri *Wrapper* karena metode *Filter* dapat dengan cepat memilih jumlah fitur yang efektif (Imani et al., 2013). Gambar 2.5 akan menunjukkan proses kerja dari seleksi fitur kategori *Embedded* secara umum.



Gambar 2.5. Alur kerja seleksi fitur kategori *Embedded*

Sumber: Kaushik, 2016

Berdasarkan Gambar 2.5, diperlihatkan bahwa *Embedded* melakukan seleksi fitur dan klasifikasi atau regresi pada saat bersamaan. Dalam implementasinya, algoritma pembelajaran yang digunakan pada kategori *Embedded* ini juga memiliki metode pemilihan fitur bawaan sendiri di dalamnya (Hendrawan et al., 2021). Pada penelitian ini, akan diterapkan dua metode dari seleksi fitur kategori *Embedded*, yaitu *Least Shrinkage and Selection Operator* (LASSO) dan *Elastic Net*.

Least Shrinkage and Selection Operator (LASSO)

Least Shrinkage and Selection Operator (LASSO) merupakan sebuah model berbasis *supervised regression* yang melakukan regularisasi terhadap fitur dan mengidentifikasi fitur yang paling informatif dan paling tidak redundan untuk memprediksi variabel respons. Secara matematis, LASSO melakukan penalti dengan cara meminimalisasi kuadrat terkecil dengan

menggunakan regularisasi L1 terhadap koefisien dari variabel prediktor (Nigon et al., 2020). Estimasi dari LASSO *regression* dapat didefinisikan seperti pada Persamaan 2.6

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda\sum_{j=1}^{p}|\beta_j|\right) \tag{2.6}$$

- N adalah jumlah observasi atau pengamatan.
- yi adalah respon pada pengamatan ke-i.
- xi adalah data, sebuah vektor dari *p-value* pada pengamatan i.
- λ adalah parameter regularisasi positif yang sesuai dengan satu nilai Lambda.
- Parameter β_0 dan β masing-masing merupakan *scaler* dan *p-vector*.

Berdasarkan Persamaan 2.6, Ketika λ meningkat, jumlah komponen yang bukan nol dari β akan berkurang karena LASSO melibatkan L1 dari β. Hal ini berarti fitur dengan koefisien sama dengan nol akan diabaikan, dan fitur dengan koefisien bukan nol akan diambil karena mengandung informasi terbanyak sehingga diizinkan untuk berkontribusi pada model prediksi.

Elastic Net

Regularisasi *Elastic Net* merupakan modifikasi dari pendekatan *Multiple Linear Regression* yang dirancang untuk memecahkan masalah penggunaan fitur pada data berdimensi tinggi (Amini & Hu, 2021). Menggunakan dua istilah penalti (L1 dan L2) yang merupakan gabungan antara LASSO dan *Ridge Regression, Elastic Net* memilih variabel secara otomatis dan melakukan penyusutan terus menerus pada fitur prediktor untuk meningkatkan akurasi prediksi. *Elastic Net* dapat menghapus atau memilih variabel prediktor yang memiliki korelasi tinggi dalam model akhir dan meningkatkan akurasi prediksi. Sama seperti LASSO, *Elastic Net* dapat menghasilkan model yang telah tereduksi fiturnya dengan memberikan penalti atau hukuman kepada variabel yang tidak sesuai hingga koefisiennya bernilai nol dan dapat dihapus (Al-Jawarneh et al., 2021). Rumus dari *Elastic Net* dapat dilihat pada Persamaan 2.7

$$\min_{\beta_0,\beta}\left(\frac{1}{2N}\sum_{i=1}^{N}(y_i-\beta_0-x_i^T\beta)^2+\lambda P_a(\beta)\right) \tag{2.7}$$

Dimana:

$$P_a(\beta)\frac{(1-a)}{2}\|\beta\|_2^2+a\|\beta\|_1=\sum_{j=1}^{p}\left(\frac{(1-a)}{2}\beta_j^2+a\left|\beta_j\right|\right) \tag{2.8}$$

Keterangan:

- N adalah jumlah observasi atau pengamatan.
- yi adalah respon pada pengamatan ke-i.
- xi adalah data, sebuah vektor dari *p-value* pada pengamatan i.
- λ adalah parameter regularisasi positif yang sesuai dengan satu nilai Lambda.
- Parameter β_0 dan β masing-masing merupakan *scaler* dan *p-vector*.

Pada Persamaan 2.7 dan 2.8, dapat dilihat bahwa *Elastic Net* akan melakukan pendekatan sama seperti LASSO (L1) saat α = 1. Namun saat α menyusut menuju 0, *Elastic Net* akan melakukan pendekatan *Ridge Regression* (L2).

2.7 Pengujian Model Regresi

Setelah melakukan prediksi menggunakan model Regresi yang telah dibangun, dibutuhkan beberapa kriteria pengujian untuk mengukur tingkat keberhasilan atau performa dari model. Dalam penelitian ini, digunakan dua kriteria pengujian yaitu *Mean Square Error* (MSE), *Root Mean Square Error* (RMSE), *Mean Absolute Error* (MAE), dan R^2 atau "*R squared"*.

2.7.1 Root Mean Squared Error (RMSE)

Pengujian *Mean Square Error* (MSE) merupakan nilai kuadrat dari perbandingan kualitas kesesuaian antara data aktual dan data hasil model prediksi (Islam, 2021). Namun terkadang jika model regresi pada akhirnya menghasilkan suatu prediksi yang sangat buruk, bagian kuadrat dari fungsi MSE dapat memperbesar kesalahan sehingga *Root Mean Square Error* (RMSE) dapat menjadi solusi. Nilai dari *Root Mean Square Error* (RMSE) adalah hasil dari akar kuadrat hasil *Mean Square Error* (MSE) (Islam, 2021). *Root Mean Square Error* (RMSE) sendiri merupakan salah satu kriteria yang paling umum digunakan untuk model regresi (Rachman, 2018). MSE dan RMSE ini mengukur besarnya tingkat *error* atau kesalahan yang terjadi dari hasil prediksi oleh model, dimana semakin kecil nilai yang didapatkan (mendekati 0), maka hasil prediksi akan semakin akurat (Suprayogi et al., 2014). Nilai dari MSE dan RMSE ini dapat dihitung dengan menggunakan Persamaan 2.9 dan 2.10.

$$MSE = \frac{\sum_{t=1}^{n}(Y_t - {Y'}_t)^2}{n} \tag{2.9}$$

$$RMSE = \sqrt{\frac{\sum_{t=1}^{n}(Y_t - {Y'}_t)^2}{n}} \tag{2.10}$$

Pada Persamaan 2.9 dan 2.10 di atas, Y_t merupakan nilai aktual atau sebenarnya pada periode t, ${Y'}_t$ adalah nilai hasil prediksi pada periode t, dan n menunjukkan jumlah prediksi yang dilakukan.

2.7.2 Mean Absolute Error (MAE)

Mean Absolute Error (MAE) merupakan salah satu pengujian dari model regresi. MAE ini menghitung rata-rata perbedaan mutlak antara nilai yang sebenarnya dan nilai hasil prediksi model yang dibangun. Semua kesalahan pengujian memiliki bobot yang sama pada MAE. Semakin kecil nilai MAE maka semakin akurat hasil prediksinya (Li et al., 2018). Untuk menghitung nilai MAE pada hasil pengujian model regresi, digunakan Persamaan 2.11 berikut.

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{2.11}$$

Pada persamaan 2.11, n merepresentasikan jumlah observasi yang dilakukan, y_i merepresentasikan *experimental values*, dan $\hat{y}_i$ merepresenmtasikan nilai hasil prediksi model.

2.7.3 Koefisien Determinasi (R^2)

Koefisien determinasi atau *R square* yang biasanya disimbolkan dengan R^2 bertujuan untuk mengukur seberapa jauh kemampuan model regresi yang dibangun dalam menerangkan pengaruh dari variasi variabel dependen (tidak bebas) (Manurung, 2015). Uji ini dinamakan koefisien determinasi karena variasi yang terjadi dalam variabel tak bebas (Y) dapat dijelaskan oleh variabel bebas (X) dengan adanya regresi linier Y atas X. Besar nilai dari koefisien determinasi adalah berkisar $0 \leq R2 \leq 1$. Jika R^2 mendekati 1 maka dapat dikatakan pengaruh variabel bebas terhadap variabel terikat adalah besar yang berarti model yang digunakan baik untuk menjelaskan pengaruh variabel tersebut (Harahap et al., 2013). Nilai dari R^2 dapat ditemukan menggunakan Persamaan 2.12

$$R^2 = 1 - \frac{sum\ squared\ regression\ (SSR)}{total\ sum\ of\ squares\ (SST)}, \tag{2.12}$$

atau

$$R^2 = 1 - \frac{\sum(y_i - \hat{y}_i)^2}{\sum(y_i - \overline{y})^2}$$

Pada Persamaan 2.12 di atas, SSR merupakan jumlah variasi yang tidak dapat dijelaskan model sedangkan SST merupakan jumlah total variasi.

2.8 Kajian Pustaka

Selama beberapa tahun terakhir, penelitian mengenai implementasi beragam teknik seleksi fitur sebagai sebuah solusi untuk mengatasi masalah *Curse of Dimensionality* pada *dataset* model regresi telah beberapa kali dilakukan. Penelitian-penelitian tersebut

menggunakan beragam jenis metode dan algoritma untuk mencapai performa regresi yang diinginkan.

Penelitian pertama dilakukan oleh (Srisungsittisunti, 2018) dengan judul "*Forward Feature Selection for Ensembles to Predict Brix Values in Mango Fruits based on NIR Spectroscopy Technique*". Penelitian ini membangun model regresi untuk memprediksi nilai *Brix* dari mangga berdasarkan empat metode regresi yang berbeda. Empat metode ini yaitu *Linear Regression* (LR), *Neural Networks* (NN), *K-Nearest Neighbor* (KNN), dan model *Ensemble* dari tiga model sebelumnya. Terdapat dua tahapan yang dilakukan oleh peneliti untuk setiap model, yaitu seluruh model akan digunakan sebagai *learning algorithm* pada setiap seleksi fitur dan prediksi. Hasil dari penelitian ini menunjukkan bahwa nilai *Standard Deviation* (SD) dan *Root Mean Square Error* (RMSE) yang lebih rendah dihasilkan oleh mangga dengan periode panen yang lebih tinggi. Untuk RMSE, model *Linear Regression* dengan *dataset* periode panen 120 hari dan seleksi fitur dengan ketiga metode menghasilkan nilai RMSE paling sedikit yaitu 1.34. Untuk performa tertinggi dalam memprediksi nilai *Brix*, pelatihan model ensemble LR-NN-KNN dengan dataset periode panen 120 hari dan seleksi fitur dengan metode KNN dilakukan dengan baik dengan memberikan nilai SD 0.16 dan RMSE mendekati minimum yaitu 1.37.

Penelitian kedua dilakukan oleh (Fashoto et al., 2021) dengan judul "*Implementatin of Machine Learning for Predicting Maize Crop Yields Using Multiple Linear Regression and Backward Elimination*". Penelitian ini bertujuan untuk menerapkan model regresi *Multiple Linear Regression* untuk memprediksi hasil panen jagung untuk satu musim di Eswatini serta melihat pengaruh dari implementasi seleksi fitur *Backward Elimination* terhadap performa model. Dataset yang digunakan pada penelitian ini awalnya akan dinormalisasi untuk menentukan apakah normalisasi dapat meningkatkan performa model. Setelah itu, *Backward Elimination* akan diimplementasikan pada dataset untuk menentukan faktor-faktor yang paling mempengaruhi hasil. Hasil performa model prediksi ini akan dianalisis menggunakan nilai R^2 dan *Root Mean Squared Error*. Penerapan normalisasi data dan *Backward Elimination* pada penelitian ini terbukti membantu meningkatkan salah satu skor dari RMSE dari 739.15 menjadi 650.95. Selain itu, dengan penggunaan *Backward Elimination*, diketahui bahwa variabel prediktor yang paling sedikit mempengaruhi prediksi hasil adalah jumlah pupuk yang digunakan, sedangkan faktor yang paling mempengaruhi prediksi adalah jumlah air, kelembaban, dan radiasi matahari.

Penelitian ketiga dilakukan oleh (Prasad et al., 2021) dengan judul "*Crop Yield Prediction in Cotton for Regional Level Using Random Forest Approach* ". Penelitian ini ingin mengevaluasi kemampuan algoritma *Random Forest Regression* dalam memprediksi hasil panen kapas di negara bagian Maharashtra, India. Sebelum pembangunan model, penelitian ini mengimplementasikan salah satu teknik seleksi fitur yaitu *Recursive Feature Elimination* (RFE) yang digabungkan dengan *Cross Validation* untuk memilih variabel independen yang paling tepat untuk prediksi hasil panen. Kemudian, model akan dikalibrasi untuk memperkirakan hasil panen pada akhir bulan september, desember, dan februari. Hasil dari penelitian ini menunjukkan potensi dari kombinasi *Random Forest Regression* dan seleksi fitur RFE yang dapat digunakan sebagai alat yang efektif dalam memprediksi hasil panen secara lebih cepat dan tepat. Hal ini dapat disimpulkan karena hasilnya menunjukkan bahwa model mampu menentukan hasil panen yang diharapkan dengan sangat baik. Hasil tertinggi yang diperoleh oleh model yaitu untuk bulan september yaitu model mampu menjelaskan dengan nilai RMSE 62,7 kg/ha, 0,32 *Mean Absolute Persentase Error* (MAPE) dan 0,85 *Index of Agreement* (IA).

Penelitian keempat dilakukan oleh (Iniyan & Jebakumar, 2021) dengan judul "*Mutual Information Feature Selection (MIFS) Based Crop Yield Prediction on Corn and Soybean Crops Using Multilayer Stacked Ensemble Regression (MSER)*". Penelitian ini menawarkan sebuah model regresi baru yaitu *MIFS based multilayer stacked ensemble* yang berlandaskan seleksi fitur *Mutual Information* dan gabungan dari beberapa algoritma regresi lain seperti *Random Forest Regression, K-Nearest Neighbour, Support Vector Regression, Decision Tree Regression* dan *Multiple linear Regression*. Performa dari model yang ditawarkan ini juga nantinya akan dibandingkan dengan model regresi biasa lainnya. Hasil dari penelitian ini yaitu *MIFS based Multilevel Stacked Ensemble Regression* dapat mengungguli model *Machine Learning* lainnya dan algoritma *advance ensemble* lainnya dengan akurasi prediksi 94,439% serta memiliki nilai MAE dan RMSE terendah dengan masing-masing nilai 6,63 dan 10,545 untuk tanaman jagung serta akurasi prediksi 92,426%, MAE dan RMSE terendah dengan masing-masing nilai 7,431 dan 11,005 untuk tanaman kedelai. Selain itu, penerapan *Mutual Information* pada dataset menghasilkan informasi bahwa berdasarkan skor fiturnya, sekitar 270 fitur jagung dan 290 fitur kedelai berkorelasi dengan hasil panen dari total 432 fitur.

Penelitian kelima dilakukan oleh (Rendall et al., 2019) dengan judul "*Wide Spectrum Feature Selection (WiSe) for Regression Model Building*". Penelitian ini membangun sebuah

metode seleksi fitur baru yang bernama *Wide Spectrum Feature Selection* (WiSe). WiSe terdiri dari dua tahap, tahap pertama adalah seleksi fitur *Filter* dan tahap kedua adalah seleksi fitur *Wrapper*. Penelitian ini menunjukkan bahwa hasil simulasi dari WiSe mampu memilih fitur yang relevan untuk membangun model, terutama *critical feature* yang berkorelasi linier atau nonlinier terhadap respon. Untuk kumpulan data yang disimulasikan, sebagian besar filter efektif dalam memilih fitur yang relevan dan pendekatan terbaik adalah dengan *Filter* berdasarkan *Pearson Correlation*.

Penelitian keenam dilakukan oleh (Yan et al., 2020) dengan judul " *Financial Distress Prediction and Feature Selection in Multiple Periods by Lassoing Unconstrained Distributed Lag Non-linear Models* ". Penelitian ini menawarkan *framework* untuk melakukan prediksi krisis keuangan dengan untuk tujuan meningkatkan kinerja sistem peringatan dini untuk perusahaan yang terdaftar di China. *Framework* yang ditawarkan ini mengimplementasikan LASSO sebagai teknik seleksi fitur yang bertujuan mengecualikan faktor yang berpotensi berlebihan pada data, mengingat daftar panjang rasio akuntansi digunakan dalam konteks prediksi krisis keuangan. Penelitian ini juga membandingkan hasil prediksi yang diperoleh dengan model prediksi lain seperti, *Neural Network, Decision Tree,* SVM, dan *Logistic Regression*. Setelah melewati serangkaian analisis perbandingan untuk menguji kinerja prediksi, hasilnya performa *framework* yang ditawarkan dapat mengungguli seluruh model prediksi lainnya dengan nilai pengujian AUC, G-mean, dan KS seluruhnya berada di atas 0.86.

Penelitian ketujuh dilakukan oleh (Fu et al., 2011) dengan judul " *Elastic Net Grouping Variable Selection Combined with Partial Least Squares Regression (EN-PLSR) for the Analysis of Strongly Multi-collinear Spectroscopic Data* ". Penelitian ini memaparkan bahwa secara umum, kita seringkali hanya menggunakan prediksi untuk menilai apakah model regresi itu bagus atau tidak. Namun, dalam kasus "p besar dan n kecil ", penghematan juga harus dipertimbangkan. Model yang lebih sederhana lebih disukai demi wawasan ilmiah ke dalam struktur data dan pengurangan beban komputasi. Oleh karena itu penelitian ini mengimplementasikan seleksi fitur *Elastic Net* bersama dengan model regresi PSLR untuk melihat pengaruh dari pengurangan fitur. Hasilnya, selain dapat mengurangi komputasi model karena berhasil mengurangi fitur yang digunakan dari 401 menjadi 69 fitur saja, penerapan model PSLR yang digunakan bersama *Elastic Net* juga dapat meningkatkan hasil RMSECV dari model yaitu dari 0.21 menjadi 0.17.

Tabel 2.1 Tabel ringkasan pustaka yang telah dikaji

Judul Penelitian, Peneliti	Metode Seleksi Fitur	Metode Regresi	Dataset	Hasil
Forward Feature Selection for Ensembles to Predict Brix Values in Mango Fruits based on NIR Spectroscopy Technique, (Srisungsittisunti, 2018)	*Forward Selection*	*Linear Regression* (LR), *Neural Networks* (NN), dan *K-Nearest Neighbor* (KNN)	Penelitian ini menggunakan tiga *dataset* NIRS dari 300 mangga. Perbedaan dari dataset NIRS mangga ini adalah dari lama panen, dataset pertama hingga ketiga secara berurutan dipanen pada hari ke-100, 110, dan 120.	Hasil dari penelitian ini menunjukkan bahwa nilai *Standard Deviation* (SD) dan *Root Mean Square Error* (RMSE) yang lebih rendah dihasilkan oleh mangga dengan periode panen yang lebih tinggi. Untuk RMSE, model *Linear Regression* dengan *dataset* periode panen 120 hari dan seleksi fitur dengan ketiga metode menghasilkan nilai RMSE paling sedikit yaitu 1.34. Untuk performa tertinggi dalam memprediksi nilai Brix, pelatihan model ensemble LR-NN-KNN dengan dataset periode panen 120 hari dan seleksi fitur dengan metode KNN dilakukan dengan baik dengan memberikan nilai SD 0.16 dan RMSE mendekati minimum yaitu 1.37.
Implementatin of Machine Learning for Predicting Maize Crop Yields Using Multiple Linear Regression and	*Backward Elimination*	*Multiple Linear Regression*	Penelitian ini mendapatkan data dari dua sumber yaitu dari Perpustakaan Pertanian	Penerapan normalisasi data dan *Backward Elimination* pada penelitian ini terbukti membantu meningkatkan salah

Backward Elimination, (Fashoto et al., 2021).			Nasional Departemen Pertanian Amerika Serikat dan Data lokal yang berasal dari Kementerian Pertanian di Eswatini	satu skor dari RMSE dari 739.15 menjadi 650.95. Selain itu, dengan penggunaan *Backward Elimination*, diketahui bahwa variabel prediktor yang paling sedikit mempengaruhi prediksi hasil adalah jumlah pupuk yang digunakan, sedangkan faktor yang paling mempengaruhi prediksi adalah jumlah air, kelembaban, dan radiasi matahari.
Crop Yield Prediction in Cotton for Regional Level Using Random Forest Approach, (Prasad et al., 2021)	*Recursive Feature Elimination* (RFE)	*Random Forest Regression*	Data yang digunakan pada penelitian ini merupakan data hasil panen tahunan yang diambil dari database Direktorat Ekonomi dan Statistik India.	Hasil dari penelitian ini menunjukkan potensi dari kombinasi *Random Forest Regression* dan seleksi fitur RFE yang dapat digunakan sebagai alat yang efektif dalam memprediksi hasil panen secara lebih cepat dan tepat. Hal ini dapat disimpulkan karena hasilnya menunjukkan bahwa model mampu menentukan hasil panen yang diharapkan dengan sangat baik. Hasil tertinggi yang diperoleh oleh model yaitu untuk bulan september yaitu model mampu menjelaskan dengan nilai RMSE 62,7 kg/ha, 0,32 *Mean Absolute Persentase Error* (MAPE) dan 0,85

				Index of Agreement (IA).
Mutual Information Feature Selection (MIFS) Based Crop Yield Prediction on Corn and Soybean Crops Using Multilayer Stacked Ensemble Regression (MSER), (Iniyan & Jebakumar, 2021)	*Mutual Information*	*Random Forest Regression, K-Nearest Neighbour, Support Vector Regression, Decision Tree Regression, Multiple linear Regression, Gradient Boosting Regression* dan *MIFS based multilayer stacked ensemble*.	Dataset yang digunakan pada penelitian ini merupakan data kinerja hasil panen jagung dan kedelai yang dikumpulkan antara tahun 1980 dan 2018 di 105 lokasi pertanian mencakup 13 negara bagian Amerika Serikat.	Hasil dari penelitian ini yaitu MIFS based Multilevel Stacked Ensemble Regression dapat mengungguli model *Machine Learning* lainnya dan algoritma *advance ensemble* lainnya dengan akurasi prediksi 94,439% serta memiliki nilai MAE dan RMSE terendah dengan masing-masing nilai 6,63 dan 10,545 untuk tanaman jagung serta akurasi prediksi 92,426%, MAE dan RMSE terendah dengan masing-masing nilai 7,431 dan 11,005 untuk tanaman kedelai. Selain itu, penerapan *Mutual Information* pada dataset menghasilkan informasi bahwa berdasarkan skor fiturnya, sekitar 270 fitur jagung dan 290 fitur kedelai berkorelasi dengan hasil panen dari total 432 fitur.
Wide Spectrum Feature Selection (WiSe) for Regression Model Building, (Rendall et al., 2019)	*Pearson Correlation, Spearman Correlation, Mutual Information,* dan *Forward Stepwise*	*Partial Least Square* (PLS), LASSO, dan *Forward Stepwise*	Dataset yang digunakan pada penelitian ini merupakan dataset simulasi dan industrial.	Penelitian ini menunjukkan bahwa hasil simulasi WiSe mampu memilih fitur yang relevan untuk membangun model, terutama *critical feature* yang berkorelasi linier

		Regression		atau nonlinier terhadap respon. Untuk kumpulan data yang disimulasikan, sebagian besar filter efektif dalam memilih fitur yang relevan dan pendekatan terbaik adalah dengan *Filter* berdasarkan *Pearson Correlation*.
Financial Distress Prediction and Feature Selection in Multiple Periods by Lassoing Unconstrained Distributed Lag Non-linear Models, (Yan et al., 2020)	LASSO	*Neural Network*, *Decition Tree*, *Support Vector Machine*, dan *Logistic Regression*.	*Dataset* yang digunakan pada penelitian ini berisi data finansial termasuk 43 rasio keuangan.	Setelah melewati serangkaian analisis perbandingan untuk menguji kinerja prediksi model yang diusulkan oleh penelitian ini, hasilnya performa *framework* yang ditawarkan dapat mengungguli seluruh model prediksi lainnya dengan nilai pengujian AUC, G-mean, dan KS seluruhnya berada di atas 0.86.
Elastic Net Grouping Variable Selection Combined with Partial Least Squares Regression (EN-PLSR) for the Analysis of Strongly Multi-collinear Spectroscopic Data, (Fu et al., 2011)	*Elastic Net*	*Partial Least Squares Regression* (PLSR)	Penelitian ini menggunakan data Spektroskopi yang bersifat Multi-kolinier	Hasilnya, selain dapat mengurangi komputasi model karena berhasil mengurangi fitur yang digunakan dari 401 menjadi 69 fitur saja, penerapan model PSLR yang digunakan bersama *Elastic Net* juga dapat meningkatkan hasil RMSECV dari model yaitu dari 0.21 menjadi 0.17.

BAB III
METODOLOGI PENELITIAN

3.1 Tahapan Penelitian

Penelitian akan dilakukan melalui beberapa tahap yaitu pengumpulan dan analisis data, seleksi fitur, pembangunan model regresi, pengujian dan perbandingan model, serta penarikan kesimpulan. Gambar 3.1 merupakan diagram yang menunjukkan beberapa tahap penelitian.

Gambar 3.1. Diagram alur penelitian

3.1.1 Pengumpulan dan Analisis Data

Tahap pengumpulan dan analisis data akan mengumpulkan, menganalisis, serta mempersiapkan (preprocessing) data yang akan digunakan untuk membangun model regresi. Data yang akan digunakan adalah dataset yang didapatkan dari penelitian Samadi et al. (2020) yang berjudul "Near Infrared Spectroscopic Data for Rapid and Simultaneous Prediction of Quality Attributes in Intact Mango Fruits" yang dapat didapatkan dari https://data.mendeley.com/datasets/b9d6s7hr33/1. Dataset tersebut berisikan data spektrum NIRS (Near Infrared Spectroscopy) 186 buah mangga utuh dengan empat kultivar yang berbeda. Dataset tersedia dalam ekstensi file .XLS dengan total baris sebanyak 186 baris dan total kolom sebanyak 1563 kolom. Kolom dataset ini terdiri dari label, prediksi vitamin, prediksi kelarutan, prediksi keasaman, serta spektrum NIRS ke-186 buah mangga.

Pada tahap ini juga akan dilakukan EDA (Exploratory Data Analysis) untuk menganalisis karakteristik dataset serta memeriksa apakah ada kecacatan (seperti adanya missing value) pada dataset. Setelah melakukan EDA, apabila ditemukan sebuah kecacatan pada dataset, akan dilakukan preprocessing yang bertujuan untuk memperbaiki dataset.

3.1.2 Seleksi Fitur

Tahap seleksi fitur akan menyeleksi 1557 fitur yang ada pada dataset dengan menggunakan metode-metode seleksi fitur. Metode seleksi fitur yang digunakan pada penelitian ini terbagi menjadi tiga kategori yaitu metode seleksi fitur kategori Filter, Wrapper, dan Embedded. Pada metode seleksi fitur kategori Filter digunakan metode ANOVA (Analysis of Variance), Mutual Information, Fisher Score, dan Pearson Correlation. Pada metode seleksi

fitur kategori Wrapper digunakan metode Forward Feature Selection, Backward Elimination, dan Recursive Feature Elimination. Pada metode seleksi fitur kategori Embedded digunakan metode LASSO (Least Absolute Shrinkage and Selection Operator) dan Elastic Net.

3.1.3 Pengujian dan Penarikan Kesimpulan

Tahap pengujian dan perbandingan model regresi akan menguji dan membandingkan performa model regresi dalam beberapa skenario trial and error. Pengujian model regresi akan dilakukan dengan mengevaluasi nilai Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), dan R-Squared (R2) dari model-model regresi yang telah dibangun. Pengujian ini akan dilakukan beberapa kali dengan total fitur yang digunakan untuk membangun sebuah model regresi berbeda-beda. Hasil dari pengujian-pengujian ini akan dibandingkan untuk menemukan model dengan performa terbaik serta seberapa banyak fitur yang dibutuhkan untuk membangun sebuah model dengan performa terbaik.

BAB IV
HASIL DAN PEMBAHASAN

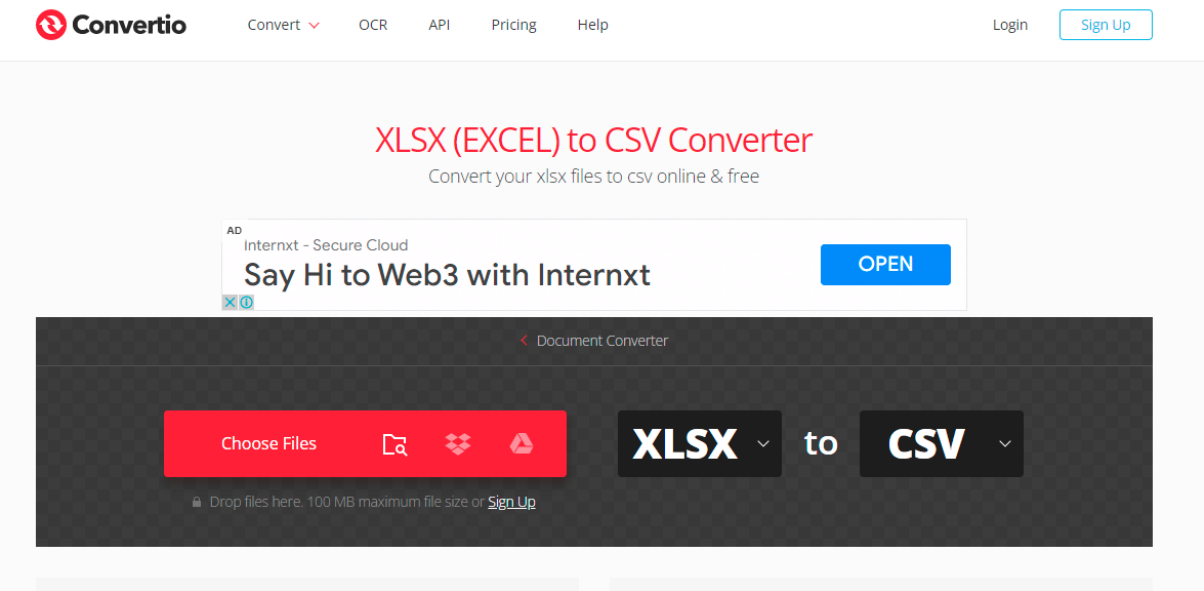
4.1 Pengumpulan Data

Data yang digunakan pada penelitian ini adalah kumpulan data Near-Infrared Spectroscopy (NIRS) mangga berjumlah 186 yang dapat ditemukan di https://data.mendeley.com/datasets/b9d6s7hr33/1. *Dataset* NIRS mangga dengan ekstensi ".xlsx" memiliki 186 baris dan 1563 kolom. *Dataset* ditampilkan seperti pada Gambar 4.1.

No	Mango Cultivars	Vit C (mg/100g)	TA (mg/100g)	SSC (oBrix)	999,9	1000,3	1000,7	1001,1	1001,4	1001,8	1002,2	1002,6	1003	1003,4	1003,8	1004,2
1	Cengkir	62,51267	599,6819	8,695	0,517039	0,516867	0,516921	0,516366	0,516206	0,51566	0,515261	0,51472	0,514282	0,513759	0,513524	0,512883
2	Cengkir	58,55433	488,5819	8,825	0,465913	0,465593	0,465691	0,465959	0,465898	0,464764	0,464363	0,464231	0,464284	0,464047	0,463307	0,462537
3	Cengkir	62,096	549,249	9,225	0,550232	0,549902	0,549755	0,549763	0,54998	0,549185	0,548379	0,547819	0,547142	0,546779	0,546289	0,545881
4	Cengkir	62,30433	464,366	8,965	0,462931	0,462902	0,4627	0,462785	0,462643	0,461748	0,461245	0,461259	0,461403	0,46089	0,460205	0,459927
5	Cengkir	46,241	346,849	9,435	0,449824	0,449643	0,44987	0,450019	0,449672	0,44884	0,4487	0,448091	0,447702	0,447621	0,447355	0,446986
6	Cengkir	35,941	437,093	10,135	0,449824	0,449643	0,44987	0,450019	0,449672	0,44884	0,4487	0,448091	0,447702	0,447621	0,447355	0,446986
7	Cengkir	51,741	167,9413	13,705	0,449824	0,449643	0,44987	0,450019	0,449672	0,44884	0,4487	0,448091	0,447702	0,447621	0,447355	0,446986
8	Cengkir	53,451	156,7747	20,815	0,499188	0,499168	0,499243	0,499276	0,498889	0,498166	0,498043	0,497774	0,497297	0,496689	0,496202	0,496262
9	Cengkir	60,381	376,3858	11,295	0,499188	0,499168	0,499243	0,499276	0,498889	0,498166	0,498043	0,497774	0,497297	0,496689	0,496202	0,496262
10	Cengkir	44,351	342,8858	14,425	0,46456	0,464569	0,46484	0,464797	0,464321	0,463564	0,46354	0,463491	0,46304	0,462485	0,462137	0,461851
11	Cengkir	49,731	309,3858	13,387	0,628127	0,627072	0,62647	0,625762	0,624661	0,623522	0,623366	0,622498	0,621493	0,620496	0,619615	0,618739
12	Cengkir	57,311	319,2947	13,517	0,628127	0,627072	0,62647	0,625762	0,624661	0,623522	0,623366	0,622498	0,621493	0,620496	0,619615	0,618739
13	Cengkir	53,531	399,208	13,917	0,49887	0,498158	0,498572	0,498509	0,497802	0,497247	0,49715	0,497005	0,496722	0,495875	0,494925	0,494425
14	Cengkir	67,131	339,1635	13,657	0,463472	0,462753	0,46289	0,462739	0,462209	0,461645	0,461649	0,461625	0,461443	0,461146	0,460213	0,459463
15	Cengkir	63,841	186,5524	14,127	0,525084	0,524445	0,524628	0,524391	0,523918	0,523329	0,523013	0,522749	0,522371	0,521691	0,520743	0,520401
16	Cengkir	57,7986	191,1991	14,827	0,491996	0,491499	0,491438	0,490788	0,490058	0,489565	0,489667	0,489497	0,489025	0,48854	0,487886	0,486877
17	Cengkir	71,772	399,208	18,397	0,471459	0,471074	0,470934	0,470379	0,47026	0,46988	0,469497	0,469435	0,469454	0,468998	0,468351	0,467619
18	Cengkir	55,7637	346,3951	25,507	0,513505	0,513086	0,513458	0,513126	0,512536	0,51202	0,51142	0,511253	0,511205	0,510674	0,509978	0,509548
19	Kweni	57,029	429,208	15,987	0,363834	0,363489	0,363418	0,36327	0,363237	0,36238	0,362127	0,36199	0,361873	0,361192	0,360576	0,36027
20	Kweni	77,8647	667,092	19,117	0,363834	0,363489	0,363418	0,36327	0,363237	0,36238	0,362127	0,36199	0,361873	0,361192	0,360576	0,36027
21	Kweni	51,9651	339,892	19,32	0,359705	0,359462	0,359603	0,359467	0,359104	0,358377	0,357889	0,357575	0,357461	0,356991	0,356505	0,356167

Gambar 4.1 Dataset Near-infrared Spectroscopy (NIRS) mangga

Dataset dengan ekstensi ".xlsx" perlu dikonversi terlebih dahulu menjadi ekstensi ".csv" untuk bisa digunakan pada penelitian. Salah satu cara konversi ".xlsx" menjadi ".csv" dengan *website online converter* https://convertio.co/xlsx-csv/ seperti yang ditampilkan pada Gambar 4.2.



Gambar 4.2 Halaman *website online converter* convertio

Setelah berhasil dikonversi menjadi file berekstensi ".csv", *dataset* yang diberi nama "Dataset_RawSpectrum_NIRS_for_Intact_Mangoes.csv" memiliki tampilan seperti pada Gambar 4.3.

Gambar 4.3 Data NIRS mangga berekstensi ".csv"

4.2 *Exploratory Data Analysis* (EDA)

Exploratory Data Analysis ((EDA) merupakan langkah pertama kali pada penelitian untuk mengetahui, memahami, dan mengeksplorasi data dengan baik. EDA merupakan Teknik untuk mendapatkan wawasan dari sebuah *dataset*. Semua tahapan pada penelitian ini diteliti menggunakan Bahasa pemrograman Python. Dijalankan di software *jupyter notebook* dan *google colab*. Sebelum memulai EDA, terlebih dahulu untuk *import library* yang diperlukan

untuk kebutuhan penelitian. *Library* yang digunakan pada penelitian dapat dilihat pada Gambar 4.4.

```
import pandas as pd
import numpy as np
from matplotlib import pyplot as plt
import seaborn as sns
import time
from sklearn.model_selection import train_test_split

from sklearn.linear_model import LinearRegression
from sklearn.ensemble import RandomForestRegressor

from sklearn.feature_selection import f_regression,SelectKBest
from sklearn.feature_selection import mutual_info_regression
from skfeature.function.similarity_based import fisher_score
from mlxtend.feature_selection import SequentialFeatureSelector as sfs
from sklearn.feature_selection import RFE
from sklearn.model_selection import GridSearchCV
from itertools import product
from sklearn.linear_model import ElasticNet
from sklearn.pipeline import Pipeline
from sklearn.linear_model import Lasso

from sklearn.model_selection import KFold
from sklearn.model_selection import cross_validate
from sklearn.metrics import mean_squared_error, mean_absolute_error,
r2_score
```

Gambar 4.4 Kode program *import libraries*

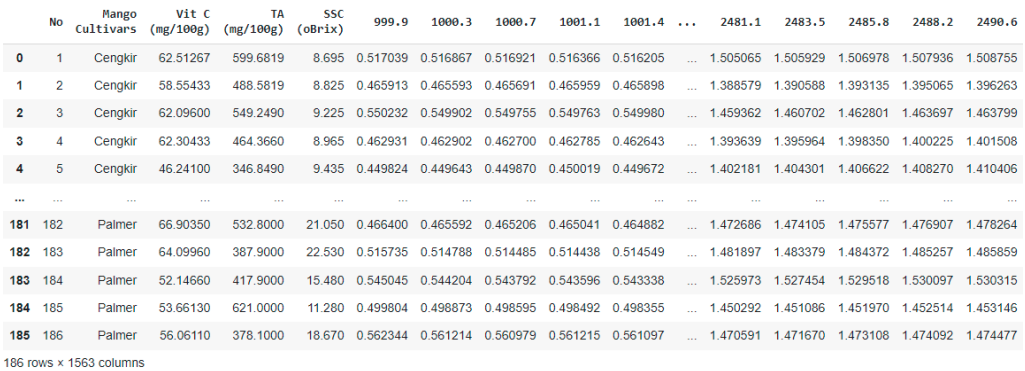
Beberapa *library* dipilih untuk menunjang kebutuhan penelitian, "pandas" menjadi salah satu *library* yang memiliki fungsi untuk analisis dan manipulasi data, "numpy" digunakan untuk membantu proses yang berhubungan dengan operasi matematika, "matplotlib.pyplot" digunakan untuk visualisasi grafik, "seaborn" digunakan untuk visualisasi yang berhubungan dengan korelasi data.

Setelah berhasil *import library*, dilanjutkan dengan *import dataset* NIRS mangga seperti yang ditampilkan pada Gambar 4.5.

```
df = pd.read_csv('Dataset_RawSpectrum_NIRS_for_Intact_Mangoes.csv')
```

Gambar 4.5 Kode program *import dataset*

Dataset dengan ekstensi ".csv" dimasukkan ke dalam *dataframe* "df" dengan bantuan fungsi dari *library* "pandas". Untuk tampilan keseluruhan isi dari dataset "Dataset_RawSpectrum_NIRS_for_Intact_Mangoes.csv" dapat dilihat pada Gambar 4.6.



	No	Mango Cultivars	Vit C (mg/100g)	TA (mg/100g)	SSC (oBrix)	999.9	1000.3	1000.7	1001.1	1001.4	...	2481.1	2483.5	2485.8	2488.2	2490.6
0	1	Cengkir	62.51267	599.6819	8.695	0.517039	0.516867	0.516921	0.516366	0.516205	...	1.505065	1.505929	1.506978	1.507936	1.508755
1	2	Cengkir	58.55433	488.5819	8.825	0.465913	0.465593	0.465691	0.465959	0.465898	...	1.388579	1.390588	1.393135	1.395065	1.396263
2	3	Cengkir	62.09600	549.2490	9.225	0.550232	0.549902	0.549755	0.549763	0.549980	...	1.459362	1.460702	1.462801	1.463697	1.463799
3	4	Cengkir	62.30433	464.3660	8.965	0.462931	0.462902	0.462700	0.462785	0.462643	...	1.393639	1.395964	1.398350	1.400225	1.401508
4	5	Cengkir	46.24100	346.8490	9.435	0.449824	0.449643	0.449870	0.450019	0.449672	...	1.402181	1.404301	1.406622	1.408270	1.410406
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
181	182	Palmer	66.90350	532.8000	21.050	0.466400	0.465592	0.465206	0.465041	0.464882	...	1.472686	1.474105	1.475577	1.476907	1.478264
182	183	Palmer	64.09960	387.9000	22.530	0.515735	0.514788	0.514485	0.514438	0.514549	...	1.481897	1.483379	1.484372	1.485257	1.485859
183	184	Palmer	52.14660	417.9000	15.480	0.545045	0.544204	0.543792	0.543596	0.543338	...	1.525973	1.527454	1.529518	1.530097	1.530315
184	185	Palmer	53.66130	621.0000	11.280	0.499804	0.498873	0.498595	0.498492	0.498355	...	1.450292	1.451086	1.451970	1.452514	1.453146
185	186	Palmer	56.06110	378.1000	18.670	0.562344	0.561214	0.560979	0.561215	0.561097	...	1.470591	1.471670	1.473108	1.474092	1.474477

186 rows × 1563 columns

Gambar 4.6 *Dataset* NIRS mangga

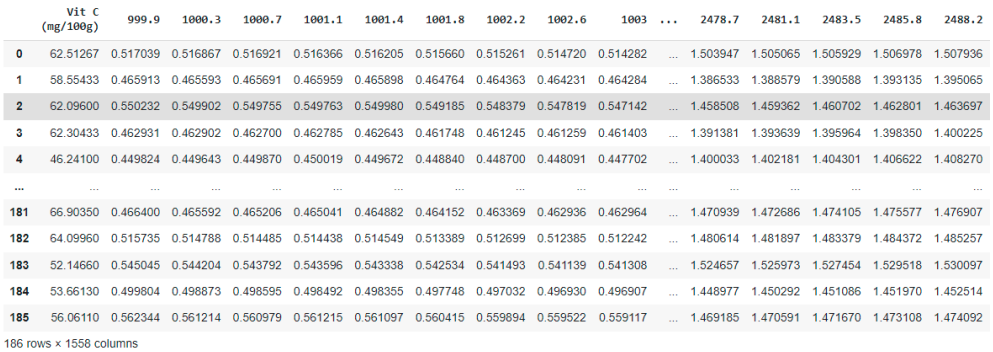
Dataset NIRS mangga memiliki 186 baris dan 1563 kolom, 1557 kolom adalah data spektrum NIRS mangga, sedangkan sisanya adalah “No”, “Mango Cultivars”, Vitc C (mg/100g), “TA (mg/100g)”, “SSC (oBrix)”, dan “label”.

Karena pada penelitian ini berfokus pada data spektrum NIRS mangga dan Vitamin C, maka kolom yang tidak dibutuhkan perlu untuk dihapus. Terlihat pada Gambar 4.7 Kode program akan *drop* atau menghapus kolom yang tidak perlu.

```
df = df.drop(['No', 'Mango Cultivars', 'TA (mg/100g)', 'SSC (oBrix)', 'label'], axis = 1)
```

Gambar 4.7 Kode program hapus kolom yang tidak digunakan

Setelah kolom yang tidak perlukan dihapus, maka *dataframe* yang baru akan terbentuk seperti pada Gambar 4.8 dengan 186 baris dan 1558 kolom.



	Vit C (mg/100g)	999.9	1000.3	1000.7	1001.1	1001.4	1001.8	1002.2	1002.6	1003	...	2478.7	2481.1	2483.5	2485.8	2488.2
0	62.51267	0.517039	0.516867	0.516921	0.516366	0.516205	0.515660	0.515261	0.514720	0.514282	...	1.503947	1.505065	1.505929	1.506978	1.507936
1	58.55433	0.465913	0.465593	0.465691	0.465959	0.465898	0.464764	0.464363	0.464231	0.464284	...	1.386533	1.388579	1.390588	1.393135	1.395065
2	62.09600	0.550232	0.549902	0.549755	0.549763	0.549980	0.549185	0.548379	0.547819	0.547142	...	1.458508	1.459362	1.460702	1.462801	1.463697
3	62.30433	0.462931	0.462902	0.462700	0.462785	0.462643	0.461748	0.461245	0.461259	0.461403	...	1.391381	1.393639	1.395964	1.398350	1.400225
4	46.24100	0.449824	0.449643	0.449870	0.450019	0.449672	0.448840	0.448700	0.448091	0.447702	...	1.400033	1.402181	1.404301	1.406622	1.408270
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
181	66.90350	0.466400	0.465592	0.465206	0.465041	0.464882	0.464152	0.463369	0.462936	0.462964	...	1.470939	1.472686	1.474105	1.475577	1.476907
182	64.09960	0.515735	0.514788	0.514485	0.514438	0.514549	0.513389	0.512699	0.512385	0.512242	...	1.480614	1.481897	1.483379	1.484372	1.485257
183	52.14660	0.545045	0.544204	0.543792	0.543596	0.543338	0.542534	0.541493	0.541139	0.541308	...	1.524657	1.525973	1.527454	1.529518	1.530097
184	53.66130	0.499804	0.498873	0.498595	0.498492	0.498355	0.497748	0.497032	0.496930	0.496907	...	1.448977	1.450292	1.451086	1.451970	1.452514
185	56.06110	0.562344	0.561214	0.560979	0.561215	0.561097	0.560415	0.559894	0.559522	0.559117	...	1.469185	1.470591	1.471670	1.473108	1.474092

186 rows × 1558 columns

Gambar 4.8 Dataset baru setelah hapus beberapa fitur

Kemudian *dataset* diteliti lebih lanjut dengan beberapa cara seperti cek ukuran dimensi, deskripsi, dan cek apakah di dalam *dataset* memiliki data yang bernilai “Not a Number” (NaN) yang dapat dilihat pada Gambar 4.9

```
df.shape
df.describe()
df.isna().sum()
```

Gambar 4.9 Kode program meneliti keadaan *dataset*

Untuk mengecek dimensi suatu dataset, menggunakan “shape”. Untuk hasil yang dikeluarkan dapat dilihat pada Gambar 4.10 yang menunjukkan bahwa *dataset* memiliki 186 baris dan 1558 kolom.

```
(186, 1558)
```

Gambar 4.10 *Output* cek dimensi *dataset*

Selanjutnya fungsi describe() untuk menghitung rangkuman statistik data seperti *mean, min, max* yang ditampilkan pada Gambar 4.11.

	Vit C (mg/100g)	999.9	1000.3	1000.7	1001.1	1001.4	1001.8	1002.2	1002.6	1003	...	2478.7	
count	186.000000	186.000000	186.000000	186.000000	186.000000	186.000000	186.000000	186.000000	186.000000	186.000000	...	186.000000	1
mean	42.782664	0.470990	0.470317	0.470073	0.469959	0.469614	0.468830	0.468439	0.468219	0.467915	...	1.529063	
std	13.686407	0.067623	0.067513	0.067419	0.067371	0.067330	0.067233	0.067149	0.067119	0.067103	...	0.127583	
min	18.327000	0.359705	0.359462	0.359603	0.359467	0.359104	0.358377	0.357889	0.357575	0.357461	...	1.223759	
25%	33.050330	0.438271	0.437551	0.437256	0.437129	0.436796	0.436155	0.435898	0.435692	0.435161	...	1.461599	
50%	40.987850	0.468262	0.467445	0.466986	0.467068	0.466896	0.466074	0.465598	0.465148	0.464806	...	1.524657	
75%	52.130782	0.499109	0.498421	0.498491	0.498465	0.497888	0.497192	0.496888	0.496721	0.496463	...	1.571987	
max	77.864700	0.759821	0.758425	0.757702	0.757295	0.756754	0.755493	0.754821	0.754700	0.754536	...	2.455570	

8 rows × 1558 columns

Gambar 4.11 Rangkuman statistik data

Dilanjutkan dengan mengamati nilai *missing values* dengan fungsi isna.sum(), Menurut *output* yang ditampilkan pada Gambar 4.12, *dataset* NIRS mangga tidak memiliki nilai yang NaN.

```
Vit C (mg/100g)    0
999.9              0
1000.3             0
1000.7             0
1001.1             0
                  ..
2490.6             0
2493               0
2495.4             0
2497.8             0
2500.2             0
```

Gambar 4.12 Cek nilai NaN

4.3 Seleksi Fitur

Seleksi fitur merupakan salah satu teknik *dimensionality reduction* (pengurangan dimensi) sebuah *set* fitur. Teknik ini menghapus fitur yang tidak relevan dan yang bersifat *redundant*. Tujuannya adalah untuk peningkatan kinerja pembelajaran model dan mengurangi waktu komputasi. Teknik seleksi fitur dapat diklasifikasikan menjadi tiga kategori yaitu *Filter*, *Wrapper*, dan *Embedded*. Untuk kategori *Filter* dan *Embedded* akan melakukan seleksi fitur dengan melakukan iterasi dari satu hingga seratus untuk menemukan performa terbaik, sedangkan kategori *Wrapper* akan menggunakan lima skenario berbeda, yaitu 20, 40, 60, 80, dan 100 untuk fitur yang akan dipilih.

4.3.1 ANOVA

Analysis of Variance adalah metode seleksi fitur dengan membandingkan rata-rata dari dua fitur. ANOVA melakukan pengujian dengan uji-F. ANOVA melakukan perbandingan untuk mengetahui apakah data yang dibandingkan merupakan data yang mirip. Untuk pendefinisian seleksi fitur ANOVA ditunjukkan pada Gambar 4.13.

```
anova = SelectKBest(score_func=f_regression,k=100)
```

Gambar 4.13 Kode program pendefinisian seleksi fitur ANOVA

Pendefinisian dilakukan dengan memilih fungsi SelectKBest() yang diisi dengan parameter “score_func” yang diberi nilai “f_regression” dengan k diisi dengan nilai 100.

Parameter tersebut memiliki arti bahwa uji-F melakukan seleksi fitur ANOVA dengan banyaknya fitur yang diambil adalah 100.

```
anova.fit(x_train,y_train)
```

Gambar 4.14 Kode program melatih seleksi fitur ANOVA

.

Langkah selanjutnya adalah melatih seleksi fitur yang ditampilkan pada Gambar 4.14. Fungsi fit() bertujuan untuk melatih dengan parameter di dalamnya “x_train” dan “y_train”.

Setelah berhasil melatih data, dilanjutkan dengan membuat *dataframe* yang diisi dengan fitur, *F-score*, dan *p-value* seperti yang ditunjukkan pada Gambar 4.15.

```
features_score = pd.DataFrame(anova.scores_)
features_pvalue = pd.DataFrame(np.round(anova.pvalues_,4))
features = pd.DataFrame(x_train.columns)

features_anova= pd.concat([features,features_score,features_pvalue],axis=1)

features_anova.columns = ["Input_Features","Score","P_Value"]

print(features_anova.nlargest(100,columns="Score"))
```

Gambar 4.15 Kode program mendapatkan 100 fitur terbaik ANOVA

Dataframe dibuat dan diisi dengan beberapa kolom, pertama adalah *scores* yang didapat dari uji-F, kemudian *p-value* yang didapat dari pvalues, dan *features* adalah seluruh fitur pada data *train*. Selanjutnya digabungkan menggunakan fungsi “concat”. Terakhir adalah mengambil dan mengurutkan 100 fitur terbaik menggunakan nilai “Score” yang didapat dari uji-F. Gambar 4.16 ditampilkan 100 fitur terbaik yang diurutkan dari “Score” tertinggi.

```
     Input_Features     Score  P_Value
117          1047.1  2.452869   0.1198
116          1046.7  2.426808   0.1217
815          1458.2  2.423241   0.1220
819          1461.5  2.423166   0.1220
806          1450.9  2.421868   0.1221
..              ...       ...      ...
148          1060.4  2.364038   0.1266
110          1044.2  2.362536   0.1267
119            1048  2.361998   0.1268
837          1476.5  2.361971   0.1268
102          1040.8  2.360734   0.1269

[100 rows x 3 columns]
```

Gambar 4.16 100 fitur terbaik seleksi fitur ANOVA

4.3.2 *Mutual Information (MI)*

Mutual Information (MI) merupakan indeks dari ketergantungan statistic antara dua variabel acak. Semakin tinggi nilai MI yang diperoleh, berarti nilai ketergantungan antar variabel lebih tinggi. Untuk pendefinisian MI ditunjukkan pada Gambar 4.17.

```
mi = SelectKBest(score_func=mutual_info_regression,k=100)
```

Gambar 4.17 Kode program pendefinisian *Mutual Information*

Pendefinisian dilakukan dengan memilih fungsi SelectKBest() yang diisi dengan parameter "score_func" yang diberi nilai "mutual_info_regression" dengan k diisi dengan nilai 100. Parameter tersebut memiliki arti bahwa seleksi fitur MI dilakukan dengan data *continuous* dengan banyaknya fitur yang diambil adalah 100.

```
mi.fit(x_train,y_train)
```

Gambar 4.18 Kode program melatih seleksi fitur *Mutual Information*

Selanjutnya adalah melatih seleksi fitur yang ditampilkan pada Gambar 4.18. Fungsi fit() bertujuan untuk melatih dengan parameter di dalamnya "x_train" dan "y_train".

Setelah berhasil melatih data, dilanjutkan dengan membuat *dataframe* yang diisi dengan fitur dan *mutual information value* seperti yang ditunjukkan pada Gambar 4.19.

```
features_mi = pd.DataFrame(mi.scores_)
features = pd.DataFrame(X_train.columns)
features_mi = pd.concat([features,features_mi],axis=1)

features_mi.columns = ["Input_Features","Mutual Information Value"]

print(features_mi.nlargest(100,columns="Mutual Information Value"))
```

Gambar 4.19 Kode program mendapatkan 100 fitur terbaik *Mutual Information*

Dataframe dibuat dan diisi dengan beberapa kolom, *mutual information value dan features*. Keduanya digabungkan dalam *dataframe* menggunakan fungsi "concat". Selanjutnya adalah mengambil 100 fitur terbaik dari seleksi fitur MI yang memiliki nilai *mutual information value* tertinggi.

	Input_Features	Mutual Information Value
318	1139.7	0.709039
319	1140.2	0.703579
322	1141.7	0.702486
315	1138.2	0.700575
316	1138.7	0.699476
..	...	...
697	1367.5	0.633102
536	1260.4	0.632364
0	999.9	0.631941
569	1281	0.631650
565	1278.5	0.631549

[100 rows x 2 columns]

Gambar 4.20 100 fitur terbaik seleksi fitur *Mutual Information*

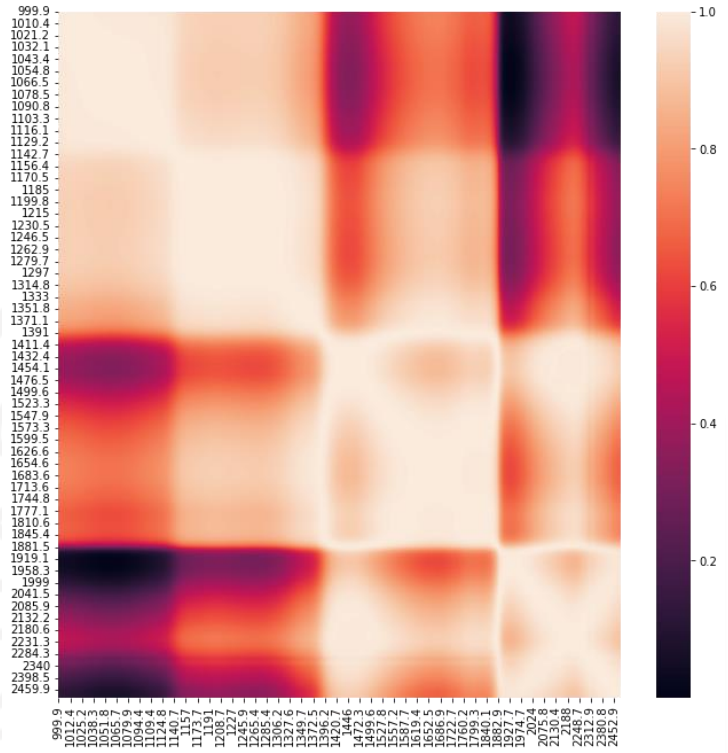
4.3.3 Pearson Correlation

Pearson Correlation adalah seleksi fitur yang mengukur kekuatan hubungan linier antara dua variabel. Nilai seleksi fitur terletak pada -1 dan 1. Nilai *Pearson Correlation* untuk mendapatkan fitur-fitur yang relevan terlihat berdasarkan *Pearson's* tertinggi. Untuk mencari *Pearson's* terlebih dahulu untuk mevisualisasikan korelasi seperti yang ditampilkan pada Gambar 4.21

```
corrmat = x_train.corr()
fig, ax = plt.subplots()
fig.set_size_inches(11,11)
sns.heatmap(corrmat)
```

Gambar 4.21 Kode program visualisasi korelasi seleksi fitur *Pearson Correlation*

Untuk keberhasilan mencari korelasi dan visualisasi korelasi, dibutuhkan fungsi corr() serta *package "*seaborn*"* yang diatur dengan ukuran panjang 11 dan lebar 11 juga. Untuk hasil visualisasi korelasi dapat dilihat pada Gambar 4.22



Gambar 4.22 Visualisasi korelasi x_train

Berikutnya, membuat *dataframe* yang diisi dengan fitur dengan korelasi setiap fitur seperti yang ditampilkan pada Gambar 4.23.

```
x_train_plus_output = x_train
x_train_plus_output['Vit C (mg/100g)'] = df[['Vit C (mg/100g)']]

cor = x_train_plus_output.corr()

df_cor_target = pd.DataFrame(abs(cor["Vit C (mg/100g)"]))
df_cor_target.index.name="Features"
df_cor_target.columns = ["Korelasi "]
df_cor_target = df_cor_target.drop('Vit C (mg/100g)')

print(df_cor_target.nlargest(100,columns="Korelasi"))
```

Gambar 4.23 Kode program mendapatkan 100 fitur terbaik *Pearson Correlation*

Untuk membuat suatu *dataframe* korelasi, diperulkan fungsi DataFrame() yang diisi dengan parameter fitur dan korelasi setiap fitur. Korelasi setiap fitur didapatkan dari mencari korelasi dari vitamin C

```
          Korelasi
Features
1047.1    0.137123
1046.7    0.136406
1458.2    0.136308
1461.5    0.136306
1450.9    0.136270
...            ...
1060.4    0.134663
1044.2    0.134621
1048      0.134606
1476.5    0.134605
1040.8    0.134571

[100 rows x 1 columns]
```

Gambar 4.24 100 fitur terbaik berdasarkan seleksi fitur *Pearson Correlation.*

4.3.4 Fisher Score

Fisher Score merupakan salah seleksi fitur yang berbasis pada *ranking* dari ratio pada masing-masing variabel prediktor. Bertujuan untuk menghapus fitur yang tidak relevan dan bersifat redundan.

```
fisher_score = fisher_score.fisher_score(X_train.to_numpy(), y_train.to_numpy())
```

Gambar 4.25 Kode program pendefinisian *Fisher Score*

```
fisher_ranking = pd.DataFrame(fisher_score)
features = pd.DataFrame(X_train.columns)
fisher_ranking = pd.concat([features,fisher_ranking],axis=1)

fisher_ranking.columns = ["Input_Features","Fisher_Ranking"]

print(fisher_ranking.nsmallest(100,columns="Fisher_Ranking"))
```

Gambar 4.26 Kode program mendapatkan 100 fitur terbaik *Fisher Score*

Pada kode program Gambar 4.26 mencari *fisher ranking* terlebih dahulu kemudian membuat dataframe yang berisi gabungan *fisher scores* dan fitur dengan fungsi concat(). Setelah berhasil membuat dataframe, Mengurutkan *ranking* fitur dari 0 hingga 99 sehingga berjumlah 100

```
      Input_Features  Fisher_Ranking
1556          2500.2               0
260           1111.3               1
1188          1845.4               2
1406          2184.3               3
1189          1846.7               4
...              ...             ...
795             1442              95
799           1445.2              96
1405          2182.4              97
804           1449.3              98
809           1453.3              99

[100 rows x 2 columns]
```

Gambar 4.27 100 fitur terbaik berdasarkan seleksi fitur *Fisher Score.*

4.3.5 *Sequential Forward Selection* (SFS)

Sequential Forward Selection (SFS) merupakan metode seleksi fitur yang membutuhkan *Learning Algorithm* untuk mencari subset terbaik dari keseluruhan fitur yang berisi kombinasi fitur dengan jumlah tertentu.

```
lr_model =  LinearRegression()
sfs     =     sfs(lr_model,     k_features=20,     forward=True,     verbose=2,
scoring='neg_root_mean_squared_error', n_jobs=-1)

sfs = sfs.fit(X_train, y_train)
```

Gambar 4.28 Kode program pendefinisian *Learning Algorithm* dan latih model

Pada kode program Gambar 4.28. diawali dengan pendefinisian *Linear Regression* sebagai *learning algorithm* yang dipilih. Kemudian diatur dengan fungsi sfs() yang menyeleksi fitur hingga 20 fitur. Parameter “forward” diberi nilai “True” yang artinya seleksi fitur berjalan dengan metode SFS, parameter “verbose” diberi nilai dua yang memiliki arti agar bisa dilihat *summary* dari setiap iterasi, parameter “scoring” menggunakan “neg_root_mean_squared_error” yang berarti pengujian model menggunakan RMSE bernilai negatif, dan parameter “n_jobs” bernilai -1 yang artinya akan menggunakan semua CPU *cores* yang tersedia.

Sequential Forward Selection dilakukan dengan melatih lima skenario dengan model *Linear Regression* dengan 20, 40, 60, 80, dan 100 iterasi. Masing-masing skenario akan dicari nilai terbaik. Untuk prosesnya pertama dipilih satu fitur terlebih dahulu dari 1557 fitur yang tersedia secara *random*. Setelah dipilih fitur pertama, dicari “scoring” dengan RMSE negatif.

Selanjutnya dicari fitur kedua dengan mengkombinasikan dengan fitur pertama untuk mendapatkan “scoring” terbaik. Dilakukan terus-menerus hingga iterasi yang ingin dicapai.

4.3.6 Backward Elimination (BE)

Backward Elimination menggunakan seluruh *set* fitur lengkap dan kemudian bertahap menghapus fitur-fitur yang tidak relevan. Seleksi fitur yang hampir mirip dengan SFS, tetapi prosesnya kebalikan dengan menghapus dari seluruh *set* fitur hingga fitur yang dipilih. Untuk *Learning Algorithm* juga sama dengan menggunakan *Linear Regression.* Tetapi pada penelitian ini tidak menggunakan 1557 fitur, tetapi hanya menggunakan 300 fitur, dengan alas an untuk meringankan beban komputasi menggunakan seleksi fitur.

```
fisher_score          =          fisher_score.fisher_score(x_train.to_numpy(),
y_train.to_numpy())

fisher_ranking = pd.DataFrame(fisher_score)
features = pd.DataFrame(x_train.columns)
fisher_ranking = pd.concat([features,fisher_ranking],axis=1)

fisher_ranking.columns = ["Features","Fisher_Ranking"]

print(fisher_ranking.nsmallest(300,columns="Fisher_Ranking"))
```

Gambar 4.29 Kode program *Backward Elimination* untuk memilih 300 fitur terpenting

```
     Features  Fisher_Ranking
1556   2500.2               0
260    1111.3               1
1188   1845.4               2
1406   2184.3               3
1189   1846.7               4
...       ...             ...
1493     2357             295
802    1447.6             296
801    1446.8             297
800      1446             298
798    1444.4             299

[300 rows x 2 columns]
```

Gambar 4.30 300 fitur terbaik berdasarkan seleksi fitur *Fisher Score.*

Selanjutnya, setelah *Fisher Score* dipilih untuk menyeleksi terlebih dahulu hingga 300 fitur, dilanjutkan dengan membuat *dataframe* BE yang dapat dilihat pada Gambar 4.31.

```
fisher_ranking = pd.DataFrame(fisher_score)
features = pd.DataFrame(x_train.columns)
fisher_ranking = pd.concat([features,fisher_ranking],axis=1)

# Assign the column name
fisher_ranking.columns = ["Features","Fisher_Ranking"]

# Print features score
print(fisher_ranking.nsmallest(300,columns="Fisher_Ranking"))
```

Gambar 4.31 Kode program mendapatkan 300 fitur terbaik *Fisher Score*

Dataframe dibuat dan diisi dengan beberapa kolom, *fisher score dan* fitur. Keduanya digabungkan dalam *dataframe* menggunakan fungsi "concat". Selanjutnya adalah mengambil 300 fitur terbaik dari seleksi fitur *Fisher Score* yang diurutkan dari ranking terkecil.

4.3.7 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) bertujuan untuk mencari jumlah fitur optimal dengan menghilangkan fitur yang paling tidak penting, lemah, dan paling tidak memengaruhi keberhasil model.

```
cols = list(x_train.columns)
model = LinearRegression()

rfe = RFE(estimator=model, n_features_to_select=20)

X_rfe = rfe.fit_transform(x_train, y_train)

model.fit(X_rfe,y_train)
temp = pd.Series(rfe.support_,index = cols)
selected_features_rfe = temp[temp==True].index
print(selected_features_rfe)
```

Gambar 4.32 Kode program seleksi fitur RFE untuk mencari 20 fitur terbaik

Linear algorithm yang digunakan adalah *Linear Regression*. Dilanjutkan dengan inisialisasi seleksi fitur RFE dengan fungsi RFE() yang memiliki parameter "estimator" yang bernilai model L*inear Regression* dan "n_features_to_select" banyaknya fitur yang diambil. Lima skenario juga diterapkan untuk mencari fitur seperti dua metode *wrapper* lainnya.

4.3.8 *Least Absolute Shrinkage and Selection Operator* (LASSO)

Least Absolute Shrinkage and Selection Operator (LASSO) melakukan regularisasi terhadap fitur dan identifikasi fitur yang paling informatif dan paling tidak redundan untuk

prediksi variabel respons. LASSO mealakukan penalti dengan cara meminimalisasi kuadrat terkecil dengan regularisasi L1.

```
pipeline = Pipeline([
                      ('scaler',StandardScaler()),
                      ('model',Lasso())
])

#buat objek GridSearchCV
search = GridSearchCV(pipeline,
                       {'model__alpha':np.arange(0.1,10,0.1)},
                       cv = 5, scoring="neg_mean_squared_error",verbose=3
                       )

search.fit(x_train,y_train)

search.best_params_

coefficients = search.best_estimator_.named_steps['model'].coef_
importance = np.abs(coefficients)
```

Gambar 4.33 Kode program *scalling dataset*

Pada kode program Gambar objek "Pipeline" melakukan *scalling* menggunakan *StandardScaler* dan objek LASSO. Selanjutnya diperlukan optimasi α LASSO. Untuk pengujian α dilakukan dari nilai 0,1 hingga 10 dengan nilai *step* sebesar 0,1. Untuk "scoring" menggunakan "neg_mean_squared_error" yaitu MSE bernilai negatif dan verbose bernilai tiga.

Proses selanjutnya melatih "x_train" dan "y_train" dengan fungsi fit() dilanjutkan dengan mencari nilai α terbaik. Setelah mendapat nilai α, selanjutnya mengambil nilai koefisien dari proses LASSO *Regression.*

```
np.array(features)[importance > 0]
```

Gambar 4.34 Kode program mengambil fitur nilai *importance* lebih dari nol

Pada kode program Gambar menjelaskan untuk mengambil fitur yang memiliki nilai *importance* diatas 0. Sedangkan fitur dengan nilai *importance* di bawah 0 tidak digunakan. Total fitur yang diambil adalah

4.3.9 *Elastic Net*

Elastic Net merupakan modifikasi dari pendekatan *Multiple Linear Regression* yang dirancang untuk memecahkan masalah penggunaan fitur pada data berdimensi tinggi. Dengan menggunakan dua istilah penalti (L1 dan L2) yang merupakan gabungan antara LASSO dan *Ridge Regression.*

```
def rmse_cv(model):
    rmse     =     np.sqrt(-cross_val_score(model,     x_train,     y_train,
scoring="neg_mean_squared_error", cv = 5))
    return(rmse)

alphas = [0.0005, 0.001, 0.01, 0.03, 0.05, 0.1]
l1_ratios = [1.5, 1.1, 1, 0.9, 0.8, 0.7, 0.5]

cv_elastic   =   [rmse_cv(ElasticNet(alpha   =   alpha,   l1_ratio=l1_ratio,
             tol=0.9)).mean()
             for (alpha, l1_ratio) in product(alphas, l1_ratios)]
```

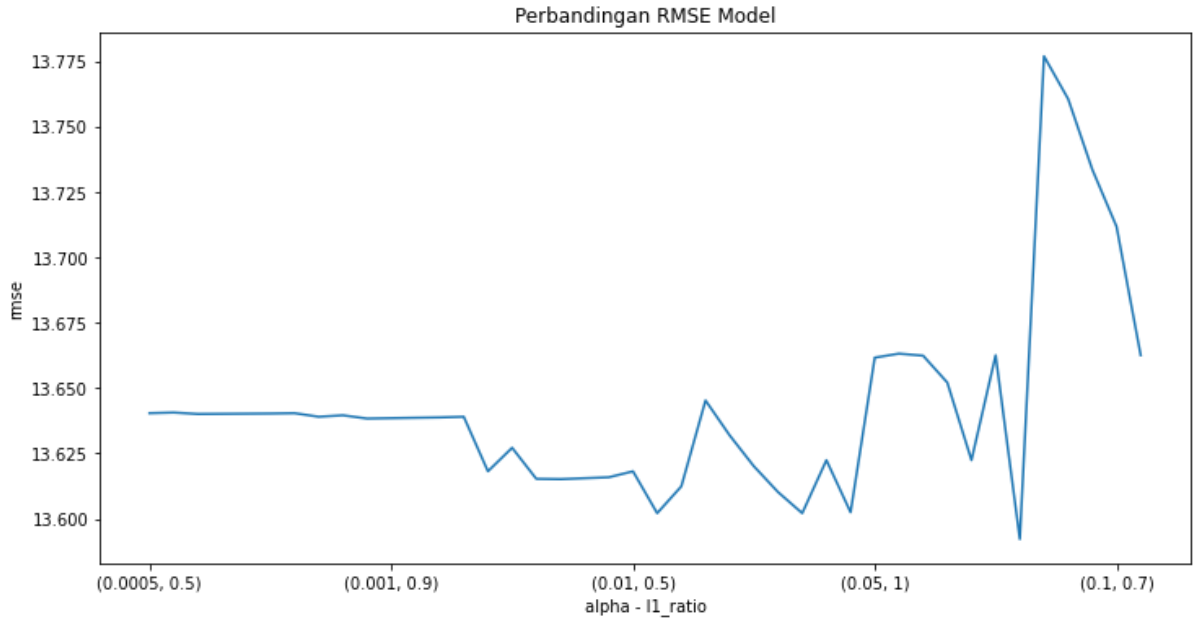
Gambar 4.34 Kode program seleksi fitur *Elastic Net*

Pada kode program Gambar, fungsi rmse cv() akan melakukan perhitungan nilai RMSE dari *Elastic Net* menggunakan nilai "aplhas" dan "l1_ratios". Nilai RMSE terkecil dari "alphas" dan "l1_ratios" akan dipilih untuk dijadikan parameter fungsi ElasticNet

```
matplotlib.rcParams['figure.figsize'] = (12.0, 6.0)
idx = list(product(alphas, l1_ratios))
p_cv_elastic = pd.Series(cv_elastic, index = idx)
p_cv_elastic.plot(title = "Perbandingan RMSE Model")
plt.xlabel("alpha - l1_ratio")
plt.ylabel("rmse")
```

Gambar 4.35 Kode program visualisasi nilai RMSE dan pasangan nilai "alphas" dan "l1_ratio"

Pada kode program Gambar akan menjelaskan cara untuk memvisualisasi nilai RMSE nilai "alphas" dan "l1_ratios". Pada Gambar merupakan visualisasi nilai RMSE pasang nilai "alphas" dan "l1_ratios".



Gambar 4.36 Visualisasi nilai RMSE (sumbu y) dan pasangan nilai "alphas" dan "l1_ratio" (sumbu x)

Hasil dari visualisasi kode program Gambar 4.36 menunjukkan beberapa nilai pasangan nilai "alphas" dan "l1_ratio" dengan nilai RMSE. Untuk nilai RMSE terkecil (yaitu sekitar 0,76) ada pada pasangan nilai "alphas" sebesar 0,001 dan "l1_ratio" sebesar 0,9, pasangan nilai inilah yang akan diambil sebagai parameter fungsi ElasticNet(). Selanjutnya, pasangan nilai "alphas" dan "l1_ratio" ini akan digunakan sebagai parameter fungsi ElasticNet() untuk melakukan seleksi fitur *Elastic Net*.

```
elastic = ElasticNet(alpha=0.05, l1_ratio=1, tol=0.9)
```

Gambar 4.37 Kode program pendefinisian seleksi fitur *Elastic Net*

Pada Gambar 4.37 ditunjukkan pendefinisian seleksi fitur *Elastic Net*. Nilai kofisien pada fungsi ElasticNet() digunakan untuk penentu apakah fitur layak untuk diambil.

```
elastic.fit(x_train, y_train)

coef = pd.Series(elastic.coef_, index = x_train.columns)

print("Elastic Net mengambil " + str(sum(coef != 0)) + " variabel dan
menghilangkan " +  str(sum(coef == 0)) + " variabel")
```

Gambar 4.38 Kode program untuk melihat total fitur yang diambil

Dari Gambar 4.38 terdapat 196 fitur dari seleksi fitur *Elastic Net*, Dari 196 fitur ini memiliki arti bahwa koefisien tidak sama dengan 0.

```
df_coef = pd.DataFrame(coef, columns=["Coeficients"])
df_fitur_terpilih = df_coef[(df_coef != 0).all(1)]

df_fitur_terpilih.nlargest(196, columns="Coeficients")
```

Gambar 4.39 Kode program mendapatkan 196 fitur terbaik *Elastic Net*

4.4 Regresi Menggunakan Data Hasil Seleksi Fitur

Setelah berhasil melakukan proses seleksi fitur dengan sembilan metode yang berbeda, selanjutnya dilakukan proses regresi menggunakan metode *Linear Regression* dan *Random Forest Regression* menggunakan fitur yang telah diseleksi. Metode regresi *Random Forest Regression* dilakukan dengan tiga skenario yang memiliki *trees* berbeda-beda, yaitu 100, 150, dan 200.

4.4.1 *Linear Regression* Menggunakan Data Hasil Seleksi Fitur

Penelitian ini menggunakan *Linear Regression* untuk melakukan prediksi regresi data NIRS mangga. Untuk pengujian data menggunakan MSE, RMSE, MAE, dan R^2. Dilanjutkan dengan menggunakan *cross-validation* seperti yang ditunjukkan pada Gambar 4.40.

```
cv = KFold(n_splits=10, random_state=1, shuffle=True)
```

Gambar 4.40 Kode program mengatur *cross-validation*

Pada fungsi KFold() diisi dengan beberapa parameter, n_splits bernilai 10 yang membagi *cross-validation* menjadi 10 “fold”, random_state sebagai pengatur keacakan data, dan shuffle bernilai “True”. Selanjutnya adalah melatih dan menguji menggunakan *Linear Regression*.

```
%%time

metrics = {'rmse': 'neg_root_mean_squared_error',
               'mse': 'neg_mean_squared_error',
               'mae': 'neg_mean_absolute_error',
               'r2': 'r2'}

n_feat = range(1, 101)

for nfeat in n_feat:
    print("================================================")
    start_time = time.time()

x_train_selected                                                    =
x_train[features_anova.nlargest(100,columns="Score").iloc[0:nfeat, 0]]
x_test_selected                                                     =
x_test[features_anova.nlargest(100,columns="Score").iloc[0:nfeat, 0]]
```

```
    lr_model = LinearRegression()

    lr_model.fit(x_train_selected, y_train)
    y_pred_lr = lr_model.predict(x_test_selected)

    scores    =    cross_validate(lr_model,    x_train_selected,    y_train,
scoring=metrics, cv=cv, return_train_score=True)

    print("MSE model Linear Regression data Train dengan " + str(nfeat) + "
fitur: "
          + str(abs(round(scores['train_mse'].mean(), 2))))
    print("RMSE model Linear Regression data Train dengan " + str(nfeat) +
" fitur:"
          + str(abs(round(scores['train_rmse'].mean(), 2))))
    print("MAE model Linear Regression data Train dengan " + str(nfeat) + "
fitur:"
          + str(abs(round(scores['train_mae'].mean(), 2))))
    print("R2 model Linear Regression data Train dengan " + str(nfeat) + "
fitur:"
          + str(abs(round(scores['train_r2'].mean(), 2))))
    print("-------------------------")
    print("MSE model Linear Regression data Test dengan " + str(nfeat) + "
fitur:"
          + str(abs(round(mean_squared_error(y_test, y_pred_lr), 2))))
    print("RMSE model Linear Regression data Test dengan " + str(nfeat) + "
fitur:"
          +  str(abs(round(mean_squared_error(y_test,  y_pred_lr,  squared =
False), 2))))
    print("MAE model Linear Regression data Test dengan " + str(nfeat) + "
fitur:"
          + str(abs(round(mean_absolute_error(y_test, y_pred_lr), 2))))
    print("R2 model Linear Regression data Test dengan " + str(nfeat) + "
fitur:"
          + str(abs(round(r2_score(y_test, y_pred_lr), 2))))
    print(" ")

    end_time = time.time()
    print("Total waktu: ", end_time - start_time)
```

Gambar 4.41 Kode program pelatihan dan pengujian model *Linear Regression*

Pertama, seperti yang dijelaskan pada kode program gambar 4.41. Tentukan variabel "metrics" yang berisi empat metode pengujian, yaitu MSE, RMSE, MAE, dan R^2. Kemudian tentukan nilai variabel "n_feat" dengan fitur yang berhasil diseleksi oleh metode seleksi fitur. Selanjutnya buat iterasi menggunakan "for" untuk melakukan perulangan dari satu hingga seratus. "x_train_selected" dan "x_test_selected" dibuat untuk menampung fitur hasil seleksi fitur sebelumnya. Dilanjutkan dengan pendefinisian metode regresi yang digunakan, yaitu *Linear Regression.* Latih metode dan melakukan prediksi dengan variabel "x_test_selected". Kemudian hitung model data train dengan "cross_validate" yang berisi memiliki parameter model yang digunakan, "scoring" yang diisi dengan nilai metode pengujian yang telah

diinisialisasi di awal. Selanjutnya munculkan hasil dari pengujian model data latih dan pengujian. Untuk hasil *Linear Regression* menggunakan salah satu metode seleksi fitur seperti yang ditunjukkan pada Gambar 4.42.

```
MSE model Linear Regression data Train dengan 2 fitur: 174.55
RMSE model Linear Regression data Train dengan 2 fitur:13.21
MAE model Linear Regression data Train dengan 2 fitur:10.64
R2 model Linear Regression data Train dengan 2 fitur:0.09
------------------------
MSE model Linear Regression data Test dengan 2 fitur:197.18
RMSE model Linear Regression data Test dengan 2 fitur:14.04
MAE model Linear Regression data Test dengan 2 fitur:10.86
R2 model Linear Regression data Test dengan 2 fitur:0.14

Total waktu:  0.1798877716064453
```

Gambar 4.42 Nilai pengujian model *Linear Regression*

Proses *Linear Regression* dengan 2 fitur menghasilkan hasil yang masih lumayan buruk. Proses regresi akan dilakukan 100 kali untuk melihat fitur yang menghasilkan performa terbaik untuk seleksi fitur ANOVA. Proses ini juga akan dilakukan terhadap semua metode seleksi fitur.

4.4.2 *Random Forest Regression* Menggunakan Data Hasil Seleksi Fitur

Penelitian ini menggunakan *Random Forest Regression* untuk melakukan prediksi regresi data NIRS mangga. Untuk pengujian data menggunakan MSE, RMSE, MAE, dan R^2. *Random Forest Regression* melakukan tiga skenario berbeda dengan menggunakan nilai trees yang berbeda-beda. Dengan cara *trial and error* ini diharapkan mendapatkan nilai yang salah satunya mencapai nilai terbaik.

```
%%time

#tentukan metode scoring yang digunakan
metrics = {'rmse': 'neg_root_mean_squared_error',
               'mse': 'neg_mean_squared_error',
               'mae': 'neg_mean_absolute_error',
               'r2': 'r2'}

#tentukan total fitur yang digunakan dalam proses klasifikasi SVM ini
n_feat = range(1, 101)
n_trees = [100]
```

```
for nfeat in n_feat:
    for ntrees in n_trees:
        print("================================================")
        start_time = time.time()

        #ambil n fitur input hasil seleksi fitur MI
        x_train_selected                                                =
x_train[fisher_ranking.nlargest(100,columns="Fisher_Ranking").iloc[0:nfeat,
0]]
        x_test_selected                                                 =
x_test[fisher_ranking.nlargest(100,columns="Fisher_Ranking").iloc[0:nfeat,
0]]

        #Create a Gaussian Classifier
        rfg_model = RandomForestRegressor(n_estimators=ntrees)

        #Train the model using the training sets
        rfg_model.fit(x_train_selected, y_train)
        y_pred_rfg=rfg_model.predict(x_test_selected)

        #hitung score model dari data train
        scores  =  cross_validate(rfg_model,  x_train_selected,  y_train,
scoring=metrics, cv=cv, return_train_score=True)

        print("MSE model Random Forest Regression data Train dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              + str(abs(round(scores['train_mse'].mean(), 2))))
        print("RMSE model Random Forest Regression data Train dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              + str(abs(round(scores['train_rmse'].mean(), 2))))
        print("MAE model Random Forest Regression data Train dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              + str(abs(round(scores['train_mae'].mean(), 2))))
        print("R2 model Random Forest Regression data Train dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              + str(abs(round(scores['train_r2'].mean(), 2))))
        print("-------------------------")
        print("MSE model Random Forest Regression data Test dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              + str(abs(round(mean_squared_error(y_test, y_pred_rfg), 2))))
        print("RMSE model Random Forest Regression data Test dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              +    str(abs(round(mean_squared_error(y_test,    y_pred_rfg,
squared = False), 2))))
        print("MAE model Random Forest Regression data Test dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              +   str(abs(round(mean_absolute_error(y_test,   y_pred_rfg),
2))))
        print("R2 model Random Forest Regression data Test dengan " +
str(nfeat) + " fitur dan " + str(ntrees) + " trees:"
              + str(abs(round(r2_score(y_test, y_pred_rfg), 2))))
        print(" ")
```

Gambar 4.43 Kode program pelatihan dan pengujian model *Random Forest Regression*

Pertama, seperti yang dijelaskan pada Gambar 4.43, tentukan variabel “metrics” yang berisi empat metode pengujian, yaitu MSE, RMSE, MAE, dan R^2. Kemudian tentukan nilai

variabel “n_feat” dengan fitur yang berhasil diseleksi oleh metode seleksi fitur. Tentukan juga n_tress sebagai banyaknya tress pada metode *Random Forest Regression*. Selanjutnya buat iterasi menggunakan “for” untuk melakukan perulangan dari satu hingga seratus. “x_train_selected” dan “x_test_selected” dibuat untuk menampung fitur hasil seleksi fitur sebelumnya. Dilanjutkan dengan pendefinisian metode regresi yang digunakan, yaitu *Random Forest Regression.* Latih metode dan melakukan prediksi dengan variabel “x_test_selected”. Kemudian hitung model data train dengan “cross_validate” yang berisi memiliki parameter model yang digunakan, “scoring” yang diisi dengan nilai metode pengujian yang telah diinisialisasi di awal. Selanjutnya munculkan hasil dari pengujian model data latih dan pengujian. Untuk hasil *Random Forest Regression* menggunakan salah satu metode seleksi fitur seperti yang ditunjukkan pada Gambar 4.44.

```
MSE model Random Forest Regression data Train dengan 1 fitur dan 100 trees:43.76
RMSE model Random Forest Regression data Train dengan 1 fitur dan 100 trees:6.61
MAE model Random Forest Regression data Train dengan 1 fitur dan 100 trees:5.14
R2 model Random Forest Regression data Train dengan 1 fitur dan 100 trees:0.77
-------------------------
MSE model Random Forest Regression data Test dengan 1 fitur dan 100 trees:279.66
RMSE model Random Forest Regression data Test dengan 1 fitur dan 100 trees:16.72
MAE model Random Forest Regression data Test dengan 1 fitur dan 100 trees:13.25
R2 model Random Forest Regression data Test dengan 1 fitur dan 100 trees:0.62

Total waktu:  1.3227825164794922
================================================
MSE model Random Forest Regression data Train dengan 2 fitur dan 100 trees:43.13
RMSE model Random Forest Regression data Train dengan 2 fitur dan 100 trees:6.56
MAE model Random Forest Regression data Train dengan 2 fitur dan 100 trees:5.09
R2 model Random Forest Regression data Train dengan 2 fitur dan 100 trees:0.77
```

Gambar 4.44 Nilai pengujian model *Random Forest Regression*

Proses *Random Forest Regression* dengan 2 fitur menghasilkan hasil yang mengalami peningkatan menjadi baik. Proses regresi akan dilakukan 100 kali untuk melihat fitur yang menghasilkan performa terbaik untuk seleksi fitur *Fisher Score*. Proses ini juga akan dilakukan terhadap semua metode seleksi fitur.

4.5 Hasil Pengujian Model Regresi

Hasil pengujian menggunakan empat metode regresi terhadap data NIRS mangga telah dilakukan. Hasil tersebut dimasukkan ke dalam tabel untuk menunjukkan hasil metode seleksi

fitur terbaik. Tabel akan dibagi menjadi empat, masing-masing tabel diuji dengan metode regresi yang. Diawali dengan pengujian terhadap metode regresi *Linear Regression*.

Tabel 4.1 Hasil Pengujian Regresi *Linear Regression*

Metode Seleksi Fitur	Jumlah Fitur Terbaik	MSE	RMSE	MAE	R^2
Tanpa Seleksi Fitur	1557	187.48	13.42	10.89	-0.17
ANOVA	10	179.54	13.4	10.31	0.04
Mutual Information	3	163.29	12.78	9.76	0.05
Fisher Score	33	132.19	11.5	9.51	0.23
Pearson Correlation	10	179,54	13.4	10.31	0.04
Forward Selection	20	164.63	12.83	10.89	0.05
Backward Elimination	20	148.38	12.18	9.62	0.14
Recursive Feature Elimination	20	179.22	13.39	10.32	0.04
LASSO	55	225.82	15.03	12.36	0.31
Elastic Net	196	152.48	12.35	10.33	0.12

Berdasarkan Tabel 4.1, Dapat dilihat bahwa dengan metode *Linear Regression* menggunakan seleksi fitur *Fisher Score* memiliki performa terbaik dengan fitur terbaik terdapat pada fitur 33 dengan 132.19 MSE, 11.5 RMSE, 9.51 MAE, dan 0.23 R^2.

Tabel 4.2 Hasil Pengujian Random Forest Regression 100 *trees*

Metode Seleksi Fitur	Jumlah Fitur Terbaik	MSE	RMSE	MAE	R^2
Tanpa Seleksi Fitur	1557	38.68	6.21	4.74	0.8
ANOVA	5	195.16	13.97	11.36	0.13
Mutual Information	74	173.55	13.17	10.4	0.06
Fisher Score	51	161.94	12.73	10.04	0.06
Pearson Correlation	89	197.01	14.04	11.11	0.14
Forward Selection	100	176.79	13.3	10.56	0.03
Backward Elimination	60	177.87	13.3	10.56	0.03

Recursive Feature Elimination	100	194.81	13.96	11.32	0.13
LASSO	55	178.64	13.37	10.45	0.04
Elastic Net	95	170.4	13.05	10.5	0.01

Berdasarkan Tabel 4.2, Dapat dilihat bahwa dengan metode *Random Forest Regression* dengan 100 tress dengan menggunakan seleksi fitur *Fisher Score* memiliki performa terbaik dengan fitur terbaik terdapat pada fitur 51 dengan 161.94 MSE, 12.73 RMSE, 10.04 MAE, dan 0.06 R^2.

Tabel 4.3 Hasil Pengujian Random Forest Regression 150 *trees*

Metode Seleksi Fitur	Jumlah Fitur Terbaik	MSE	RMSE	MAE	R^2
Tanpa Seleksi Fitur	1557	38.68	6.21	4.74	0.8
ANOVA	78	198	14.07	11.4	0.15
Mutual Information	87	173.56	13.17	10.42	0.01
Fisher Score	15	164.2	12.81	10.3	0.05
Pearson Correlation	70	197.61	14.06	11.27	0.15
Forward Selection	100	177.08	13.31	10.57	0.03
Backward Elimination	60	179.85	13.41	10.47	0.04
Recursive Feature Elimination	100	194.61	13.95	11.38	0.13
LASSO	55	179.61	13.4	10.51	0.04
Elastic Net	94	168.59	12.98	10.29	0.02

Berdasarkan Tabel 4.3, Dapat dilihat bahwa dengan metode *Random Forest Regression* dengan 150 tress dengan menggunakan seleksi fitur *Fisher Score* memiliki performa terbaik dengan fitur terbaik terdapat pada fitur 15 dengan 164.2 MSE, 12.81 RMSE, 10.3 MAE, dan 0.05 R^2.

Tabel 4.4 Hasil pengujian Random Forest Regression 200 *trees*

Metode Seleksi Fitur	Jumlah Fitur Terbaik	Accuracy	Precision	Recall	Waktu Eksekusi
Tanpa Seleksi Fitur	1557	38.68	6.21	4.74	0.8
ANOVA	80	196.07	14	11.14	0.14
Mutual Information	66	177.54	13.32	10.54	0.03
Fisher Score	13	166.07	12.89	10.35	0.04
ReliefF	91	196.24	14.01	11.17	0.14
Forward Selection	100	175.34	13.24	10.57	0.02
Backward Elimination	100	173.5	13.17	10.42	0.01
Recursive Feature Elimination	100	193.27	13.9	11.33	0.12
LASSO	55	179.66	13.4	10.62	0.04
Elastic Net	88	167.59	12.95	10.35	0.03

Berdasarkan Tabel 4.4, Dapat dilihat bahwa dengan metode *Random Forest Regression* dengan 150 tress dengan menggunakan seleksi fitur *Fisher Score* memiliki performa terbaik dengan fitur terbaik terdapat pada fitur 15 dengan 166.07 MSE, 12.89 RMSE, 10.35 MAE, dan 0.

4.5.1 Waktu Eksekusi Kode Setiap Seleksi Fitur

Setiap metode seleksi fitur memiliki waktu eksekusi yang berbeda-beda. Untuk mendapatkan waktu eksekusi ditambahkan kode *"*%%time" di setiap *cell* kode yang akan dieksekusi. *Import library* time sebelum melakukan eksekusi seleksi fitur agar kode "%%time" dapat berjalan. Lamanya waktu eksekusi dihitung dalam detik (s). Pada tabel 4.1 dijelaskan metode seleksi fitur yang digunakan dan lamanya waktu eksekusi masing-masing metode seleksi fitur.

Tabel 4.5 Waktu Eksekusi Kode dari Setiap Metode Selekesi Fitur

Metode Seleksi Fitur	Jumlah Fitur	Waktu Eksekusi
ANOVA	100	0.83
Mutual Information	100	1.86

Fisher Score	100	0.403
Pearson Correlation	100	25
Forward Selection	20	81
	40	234
	60	428
	80	843
	100	1214
Backward Elimination	20	1735
	40	1642
	60	1580
	80	1497
	100	1354
Recursive Feature Elimination	20	21.3
	40	19.3
	60	97
	80	106
	100	20.9
LASSO	55	0.282
Elastic Net	196	42.6

Menurut tabel 4.5, metode seleksi fitur tercepat adalah LASSO diikuti dengan *Fisher Score* sedangkan waktu seleksi fitur terlama adalah *Backward Elimination* dengan skenario 20 seleksi fitur. Terbukti bahwa kateogri *Filter* dan *Embedded* lebih cepat dan singkat daripada *Wrapper* yang membutuhkan waktu komputasi yang lama.

BAB V
KESIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian ini didapat kesimpulan sebagai berikut:

a. *Fisher Score* adalah metode seleksi fitur yang memiliki performa terbaik dibandingkan dengan Sembilan metode lainnya.
b. Metode regresi yang dilakukan *Linear Regression* dapat mengungguli hasil dari *Random Forest Regression* yang telah melakukan tiga skenario berbeda mengguankan tiga *trees* yang berbeda.
c. Hasil yang diperoleh dari prediksi vitamin c pada mangga berbeda tergantung kepada model regresi, metode seleksi fitur. *Linear Regression* tanpa seleksi fitur dan menggunakan seleksi fitur, mendapatkan performa lebih baik dibandingkan dengan performa metode *Random Forest Regression.*
d. Waktu komputasi untuk metode kategori *Wrapper* lebih lama dibandingkan dengan metode kategori *Filter* dan *Embedded.*

5.2 Saran

Penelitian ini masih memiliki banyak kekurangan yang perlu diperbaiki sehingga membutuhkan saran unuk penelitian selanjutnya:

a. Menggunakan metode regresi lainnya
b. Menggunakan metode seleksi fitur lainnya untuk mendapatkan
c. Menambahkan skenario *trees* pada seleksi fitur untuk mendapatkan hasil yang lebih baik

DAFTAR PUSTAKA

Akkaya, B. (2021). The Effect of Recursive Feature Elimination with Cross-Validation Method on Classification Performance with Different Sizes of Datasets. *IV International Conference on Data Science and Applications (ICONDATA'21)*, *June*, 4–6. https://www.researchgate.net/publication/354253728_The_Effect_of_Recursive_Feature_Elimination_with_Cross-Validation_Method_on_Classification_Performance_with_Different_Sizes_of_Datasets

Al-Jawarneh, A. S., Ismail, M. T., & Awajan, A. M. (2021). Elastic Net Regression and Empirical Mode Decomposition for Enhancing the Accuracy of the Model Selection. *International Journal of Mathematical, Engineering and Management Sciences*, *6*(2), 564–583. https://doi.org/10.33889/IJMEMS.2021.6.2.034

Amini, F., & Hu, G. (2021). A two-layer feature selection method using Genetic Algorithm and Elastic Net. *Expert Systems with Applications*, *166*(September 2020), 114072. https://doi.org/10.1016/j.eswa.2020.114072

Anukrishna, P. R., & Paul, V. (2017). A review on feature selection for high dimensional data. *Proceedings of the International Conference on Inventive Systems and Control, ICISC 2017*, *5*(6), 395–402. https://doi.org/10.1109/ICISC.2017.8068746

Beraha, M., Metelli, A. M., Papini, M., Tirinzoni, A., & Restelli, M. (2019). Feature Selection via Mutual Information: New Theoretical Insights. *Proceedings of the International Joint Conference on Neural Networks*, *2019-July*. https://doi.org/10.1109/IJCNN.2019.8852410

Breiman, L. E. O. (2001). *Random Forests*. 5–32.

Chandra, B. (2015). Gene Selection Methods for Microarray Data. In *Applied Computing in Medicine and Health*. Elsevier Inc. https://doi.org/10.1016/B978-0-12-803468-2.00003-5

Devianti, D., Sufardi, S., Zulfahrizal, Z., & Munawar, A. A. (2019). Near Infrared Reflectance Spectroscopy: Prediksi Cepat dan Simultan Kadar Unsur Hara Makro pada Tanah Pertanian. *AgriTECH*, *39*(1), 12. https://doi.org/10.22146/agritech.42430

Effrosynidis, D., & Arampatzis, A. (2021). An evaluation of feature selection methods for environmental data. *Ecological Informatics*, *61*(April 2020), 101224. https://doi.org/10.1016/j.ecoinf.2021.101224

Fashoto, S., Mbunge, E., Ogunleye, G., & Van den Burg, J. (2021). Implementation of Machine Learning for Predicting Maize Crop Yields Using Multiple Linear Regression and Backward Elimination. *Malaysian Journal of Computing*, *6*(1), 679. https://doi.org/10.24191/mjoc.v6i1.8822

Filzmoser, P., & Nordhausen, K. (2021). Robust linear regression for high-dimensional data: An overview. *Wiley Interdisciplinary Reviews: Computational Statistics*, *13*(4), 1–18. https://doi.org/10.1002/wics.1524

Frénay, B., Doquire, G., & Verleysen, M. (2013). Is mutual information adequate for feature selection in regression? *Neural Networks*, *48*, 1–7. https://doi.org/10.1016/j.neunet.2013.07.003

Fu, G. H., Xu, Q. S., Li, H. D., Cao, D. S., & Liang, Y. I. Z. (2011). Elastic net grouping variable selection combined with partial least squares regression (EN-PLSR) for the analysis of strongly multi-collinear spectroscopic data. *Applied Spectroscopy*, *65*(4), 402–408. https://doi.org/10.1366/10-06069

Ghojogh, B., & Crowley, M. (2019). *The Theory Behind Overfitting, Cross Validation, Regularization, Bagging, and Boosting: Tutorial*. 1–23. http://arxiv.org/abs/1905.12787

Gupta, A. K., Singh, V., Mathur, P., & Travieso-Gonzalez, C. M. (2021). Prediction of COVID-19 pandemic measuring criteria using support vector machine, prophet and linear regression models in Indian scenario. *Journal of Interdisciplinary Mathematics*, *24*(1), 89–108. https://doi.org/10.1080/09720502.2020.1833458

Haque, M. M., Rahman, A., Hagare, D., & Chowdhury, R. K. (2018). A comparative assessment of variable selection methods in urban water demand forecasting. *Water (Switzerland)*, *10*(4), 1–15. https://doi.org/10.3390/w10040419

Harahap, Y. M., Bu'ulolo, F., & Sitepu, H. R. (2013). Faktor-Faktor Yang Mempengaruhi Permintaan Air Minum Pada Perusahaan Daerah Air Minum (PDAM) Tirtanadi Medan. *Saintia Matematika*, *1*(4), 323–336.

Hasanah, U. (2018). Penentuan Kadar Vitamin C Pada Mangga Kweni Dengan Menggunakan Metode Iodometri. *Jurnal Keluarga Sehat Sejahtera*, *16*(31), 90–95. https://doi.org/10.24114/jkss.v16i31.10176

Hendrawan, A., Huizen, L. M., Pinem, A. P. R., & Wicaksana, D. A. (2021). Implementasi Pemilihan Fitur Metode Wrapper dan Embedded dalam Prediksi Ketepatan Kelulusan Mahasiswa. *Seminar Nasional Penelitian Dan Pengabdian Kepada Masyarakat (SNNPKM)*, 330–335.

Herwanto, H. W., Widiyaningtyas, T., & Indriana, P. (2019). Penerapan Algoritme Linear Regression untuk Prediksi Hasil Panen Tanaman Padi. *Jurnal Nasional Teknik Elektro Dan Teknologi Informasi (JNTETI)*, *8*(4), 364. https://doi.org/10.22146/jnteti.v8i4.537

Hsu, H. H., Hsieh, C. W., & Lu, M. Da. (2011). Hybrid feature selection by combining filters and wrappers. *Expert Systems with Applications*, *38*(7), 8144–8150. https://doi.org/10.1016/j.eswa.2010.12.156

Imani, M. B., Keyvanpour, M. R., & Azmi, R. (2013). A novel embedded feature selection method: A comparative study in the application of text categorization. *Applied Artificial Intelligence*, *27*(5), 408–427. https://doi.org/10.1080/08839514.2013.774211

Indarwati, T., Irawati, T., & Rimawati, E. (2019). Penggunaan Metode Linear Regression Untuk Prediksi Penjualan Smartphone. *Jurnal Teknologi Informasi Dan Komunikasi (TIKomSiN)*, *6*(2), 2–7. https://doi.org/10.30646/tikomsin.v6i2.369

Iniyan, S., & Jebakumar, R. (2021). Mutual Information Feature Selection (MIFS) Based Crop Yield Prediction on Corn and Soybean Crops Using Multilayer Stacked Ensemble Regression (MSER). *Wireless Personal Communications*, *0123456789*. https://doi.org/10.1007/s11277-021-08712-9

Islam, A. S. (2021). *Perbandingan Single Exponential Smoothing dan Metode Single Moving Average untuk Peramalan Tingkat Penjualan Batu Kapur di Kabupaten Jember*. http://repository.unmuhjember.ac.id/11557/%0Ahttp://repository.unmuhjember.ac.id/11557/1/1.PENDAHULUAN.pdf

Kaushik, S. (2016). *Introduction to Feature Selection methods with an example (or how to select the right variables?)*.

Lamba, M., Munjal, G., & Gigras, Y. (2018). Feature Selection of Micro-array expression data (FSM) - A Review. *Procedia Computer Science*, *132*, 1619–1625. https://doi.org/10.1016/j.procs.2018.05.127

Le Thi, H. A., Nguyen, V. V., & Ouchani, S. (2008). Gene selection for cancer classification using DCA. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, *5139 LNAI*, 62–72. https://doi.org/10.1007/978-3-540-88192-6_8

Li, Y., Zou, C., Berecibar, M., Nanini-Maury, E., Chan, J. C. W., van den Bossche, P., Van Mierlo, J., & Omar, N. (2018). Random forest regression for online capacity estimation of lithium-ion batteries. *Applied Energy*, *232*(February), 197–210. https://doi.org/10.1016/j.apenergy.2018.09.182

M., A. D. N., H., M. A., M. T., N., K. A., N. D., & Jusoh, M. H. (2020). Comparative Analysis of PCA and ANOVA for Assessing the Subset Feature Selection of the Geomagnetic Disturbance Storm Time. *Journal of Electrical & Electronic Systems Research*, *17*(DEC 2020), 8–16. https://doi.org/10.24191/jeesr.v17i1.002

Manurung, H. T. (2015). Analisis Pengaruh ROE, EPS, NPM dan MVA Terhadap Harga Saham (Studi Kasus Pada Perusahaan Manufaktur Go Public Sektor Food dan Beverages di BEI Tahun 2009-2013. *Diponegoro Journal of Accounting*, *4*(4), 1–16.

Mechram, S., Rahadi, B., & Kusuma, Z. (2021). *Teknologi Nirs (Near Infrared Reflectance Spectroscopy) Untuk Mendeteksi Kesuburan Tanah Studi Kasus di Provinsi Aceh :*

Review Nirs Technology (Near Infrared Reflectance Spectroscopy) for Detecting Soil Fertility Case Study in Aceh Province : Review P. *2021*, 71–75.

Mulyati, T. ana. (2021). Analisa Kadar Vitamin C Mangga Podang (Mangifera indica L.) pada berbagai Tingkat Kematangan dengan Metode Spektroskopi UV-VIS. *Journal of Herbal, Clinical and Pharmaceutical Science (HERCLIPS)*, *2*(02), 31. https://doi.org/10.30587/herclips.v2i02.2572

Munawar, A. A., von Hörsten, D., Wegener, J. K., Pawelzik, E., & Mörlein, D. (2016). Rapid and non-destructive prediction of mango quality attributes using Fourier transform near infrared spectroscopy and chemometrics. *Engineering in Agriculture, Environment and Food*, *9*(3), 208–215. https://doi.org/10.1016/j.eaef.2015.12.004

Nagaria, J., & Senthil Velan, S. (2020). Utilizing Exploratory Data Analysis for the Prediction of Campus Placement for Educational Institutions. *2020 11th International Conference on Computing, Communication and Networking Technologies, ICCCNT 2020*. https://doi.org/10.1109/ICCCNT49239.2020.9225441

Nigon, T. J., Yang, C., Paiao, G. D., Mulla, D. J., Knight, J. F., & Fernández, F. G. (2020). Prediction of early season nitrogen uptake in maize using high-resolution aerial hyperspectral imagery. *Remote Sensing*, *12*(8). https://doi.org/10.3390/RS12081234

Normawati, D., & Ismi, D. P. (2019). K-Fold Cross Validation for Selection of Cardiovascular Disease Diagnosis Features by Applying Rule-Based Datamining. *Signal and Image Processing Letters*, *1*(2), 23–35. https://doi.org/10.31763/simple.v1i2.3

Ottley, A., Garnett, R., & Wan, R. (2019). Follow the clicks: Learning and anticipating mouse interactions during exploratory data analysis. *Computer Graphics Forum*, *38*(3), 41–52. https://doi.org/10.1111/cgf.13670

Prasad, N. R., Patel, N. R., & Danodia, A. (2021). Crop yield prediction in cotton for regional level using random forest approach. *Spatial Information Research*, *29*(2), 195–206. https://doi.org/10.1007/s41324-020-00346-6

Rachman, T. (2018). Penerapan Aplikasi RapidMiner Untuk Prediksi Nilai Tukar Rupiah Terhadap US Dollar Dengan Metode Regresi Linier. *Angewandte Chemie International Edition, 6(11), 951–952.*, 10–27.

Rahat Hossain, M., Maung Than Oo, A., & B M Shawkat Ali, A. (2013). The Combined Effect of Applying Feature Selection and Parameter Optimization on Machine Learning Techniques for Solar Power Prediction. *American Journal of Energy Research*, *1*(1), 7–16. https://doi.org/10.12691/ajer-1-1-2

Rahman, N., Ofika, M., & Said, I. (2015). Analisis Kadar Vitamin C Mangga Gadung (Mangifera SP) dan Mangga Golek (Mangifera Indica L) Berdasarkan Tingkat Kematangan dengan Menggunakan Metode Iodimetri. *Jurnal Akademika Kimia*, *4*(1), 33–37. http://jurnal.untad.ac.id/jurnal/index.php/JAK/article/view/7844

Rajab, M., & Wang, D. (2020). Practical Challenges and Recommendations of Filter Methods for Feature Selection. *Journal of Information and Knowledge Management*, *19*(1). https://doi.org/10.1142/S0219649220400195

Ramadhani, N. L. (2022). *PENENTUAN KUALITAS SECARA FISIK DAN KIMIA SERTA PREDIKSI VITAMIN C PADA BUAH APEL FUJI (MALUS DOMESTICA BORKH.)*. *8.5.2017*, 2003–2005.

Rendall, R., Castillo, I., Schmidt, A., Chin, S. T., Chiang, L. H., & Reis, M. (2019). Wide spectrum feature selection (WiSe) for regression model building. *Computers and Chemical Engineering*, *121*, 99–110. https://doi.org/10.1016/j.compchemeng.2018.10.005

Sahu, B., Dehuri, S., & Jagadev, A. (2018). A Study on the Relevance of Feature Selection Methods in Microarray Data. *The Open Bioinformatics Journal*, *11*(1), 117–139. https://doi.org/10.2174/1875036201811010117

Sainin, M. S., & Alfred, R. (2011). A genetic based wrapper feature selection approach using Nearest Neighbour Distance Matrix. *Conference on Data Mining and Optimization*, *July*, 237–242. https://doi.org/10.1109/DMO.2011.5976534

Singh, U., Hur, M., Dorman, K., & Wurtele, E. S. (2020). MetaOmGraph: A workbench for interactive exploratory data analysis of large expression datasets. *Nucleic Acids Research*, *48*(4), E23. https://doi.org/10.1093/nar/gkz1209

Srisungsittisunti, B. (2018). Forward Feature Selection for Ensembles to Predict Brix Values in Mango Fruits based on NIR Spectroscopy Technique. *International Journal of Science*, *15*(2), 43–57.

Suarsa, I. W. (2015). *SPEKTROSKOPI*. *1*. https://doi.org/10.1007/BF00504655

Suprayogi, I., Trimaijon, & Mahyudin. (2014). Model Prediksi Liku Kalibrasi Menggunakan Pendekatan Jaringan Saraf Tiruan (ZST) (Studi Kasus : Sub DAS Siak Hulu). *Jurnal Online Mahasiswa Fakultas Teknik Universitas Riau*, *1*(1), 1–18.

Tummers, J., Catal, C., Tobi, H., Tekinerdogan, B., & Leusink, G. (2020). Coronaviruses and people with intellectual disability: an exploratory data analysis. *Journal of Intellectual Disability Research*, *64*(7), 475–481. https://doi.org/10.1111/jir.12730

Venkat, N. (2018). *The Curse of Dimensionality: Inside Out*. 1–8. https://github.com/nmakes/curse-of-dimensionality

Wagschal, U. (2016). Regression analysis. In *Handbook of Research Methods and Applications in Political Science* (Issue July). https://doi.org/10.7748/nr1996.10.4.1.318.c6066

Wang, S., Tang, J., & Liu, H. (2017). *Feature Selection BT - Encyclopedia of Machine Learning and Data Mining* (C. Sammut & G. I. Webb (eds.); pp. 503–511). Springer US. https://doi.org/10.1007/978-1-4899-7687-1_101

Wang, S., Tang, J., & Liu, H. (2020). Encyclopedia of Machine Learning and Data Science. *Encyclopedia of Machine Learning and Data Science*, *January*. https://doi.org/10.1007/978-1-4899-7502-7

Yan, D., Chi, G., & Lai, K. K. (2020). Financial distress prediction and feature selection in multiple periods by lassoing unconstrained distributed lag non-linear models. *Mathematics*, *8*(8). https://doi.org/10.3390/MATH8081275

Yang, L., Liu, S., Tsoka, S., & Papageorgiou, L. G. (2016). Mathematical programming for piecewise linear regression analysis. *Expert Systems with Applications*, *44*, 156–167. https://doi.org/10.1016/j.eswa.2015.08.034

Yuhua, Q., Xiangqian, D., & Huili, G. (2013). Application of high-dimensional feature selection in near-infrared spectroscopy of cigarettes' qualitative evaluation. *Spectroscopy Letters*, *46*(6), 397–402. https://doi.org/10.1080/00387010.2012.746373

Yuliati, N., & Kurniawati, E. (2017). Analisis Kadar Vitamin C Dan Fruktosa Pada Buah Mangga (Mangifera indica L.) Varietas Podang Urang Dan Podang Lumut Metode Spektrofotometri Uv-Vis. *Jurnal Wiyata*, *4*(1), 49–57.

Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., & Saeed, J. (2020). A Comprehensive Review of Dimensionality Reduction Techniques for Feature Selection and Feature Extraction. *Journal of Applied Science and Technology Trends*, *1*(2), 56–70. https://doi.org/10.38094/jastt1224

Zhang, W., Wu, C., Li, Y., Wang, L., & Samui, P. (2021). Assessment of pile drivability using random forest regression and multivariate adaptive regression splines. *Georisk*, *15*(1), 27–40. https://doi.org/10.1080/17499518.2019.1674340

LAMPIRAN

Seluruh hasil seleksi fitur:

Kategori Filter:

1. ANOVA:

100 fitur - ['1047.1', '1046.7', '1458.2', '1461.5', '1450.9', '1451.7', '1459.1', '1452.5', '1446.8', '1446', '1045', '1453.3', '1459.9', '1460.7', '1454.1', '1051.8', '1462.3', '1450.1', '1457.4', '1445.2', '1051', '1037.9', '1447.6', '1455', '1456.6', '1455.8', '1442.8', '1463.2', '1449.3', '1036.7', '1448.5', '1441.2', '1442', '1443.6', '1444.4', '1464', '1050.5', '1047.6', '1060', '1464.8', '1052.2', '1048.8', '1470.6', '1054.4', '1049.3', '1038.3', '1466.5', '1469', '1440.4', '1465.6', '1471.5', '1045.5', '1468.1', '1469.8', '1467.3', '1437.2', '1040.4', '1042.5', '1439.6', '1438', '1059.6', '1050.1', '1033.8', '1051.4', '1046.3', '1057.8', '1032.6', '1438.8', '1472.3', '1049.7', '1054', '1033.4', '1473.1', '1058.3', '1436.4', '1056.1', '1056.5', '1038.8', '1040', '1474', '1034.6', '1036.3', '1031.7', '1039.6', '1042.9', '1048.4', '1433.2', '1434', '1435.6', '1037.5', '1434.8', '1043.8', '1474.8', '1044.6', '1037.1', '1060.4', '1044.2', '1048', '1476.5', '1040.8']

2. Mutual Information

100 fitur - ['1003', '1009.2', '1003.8', '1008.8', '1003.4', '1007.7', '1008.1', '1004.2', '1009.6', '1002.6', '1010', '1008.5', '1002.2', '1001.8', '1133.7', '1012.8', '1007.3', '1138.2', '1005.7', '1134.2', '1011.2', '1006.5', '1006.1', '1004.5', '1001.1', '999.9', '1001.4', '1138.7', '1010.4', '1137.7', '1010.8', '1005.3', '1004.9', '1014', '1000.7', '1000.3', '1134.7', '1015.2', '1139.2', '1132.2', '1006.9', '1136.2', '1137.2', '1013.2', '1129.7', '1135.2', '1139.7', '1014.4', '1130.7', '1623.5', '1015.6', '1013.6', '1624.5', '2214.1', '1012', '1014.8', '1016', '1129.2', '2212.2', '1136.7', '1131.2', '1622.5', '1125.8', '1011.6', '1751.8', '1094', '1133.2', '1130.2', '1012.4', '1135.7', '1128.3', '1132.7', '1131.7', '1016.4', '1621.5', '1018', '1626.6', '1753', '1625.5', '1017.2', '2223.6', '1633.7', '1092.1', '2219.8', '1750.7', '2216', '2210.3', '1016.8', '2217.9', '2202.8', '1630.6', '1620.5', '2221.7', '1127.8', '1126.3', '1631.7', '1017.6', '1086.6', '1619.4', '1128.7']

3. Fisher:

100 fitur - ['2500.2', '1111.3', '1845.4', '2184.3', '1846.7', '1852', '1853.3', '1857.3', '1863.9', '1838.8', '1816.9', '2197.2', '1820.7', '2204.7', '1837.5', '1865.3', '1905', '1896.7', '1913.5', '1893.9', '1872', '1874.7', '1869.3', '1882.9', '1885.6', '2189.8', '2178.8', '1891.1', '2210.3', '1744.8', '1751.8', '1723.9', '1726.2', '1732', '1733.1', '1734.3', '1763.8', '1789.3', '1798', '1804.3', '1805.5', '1806.8', '1809.3', '1777.1', '1768.6', '1769.8', '1781.9', '1785.6', '1814.4', '1923.4', '2085.9', '2102.8', '2107.9', '2109.6', '2101.1', '2082.5', '2067.6', '2116.5', '2155.2', '2151.6', '2077.5', '1974.7', '1936.3', '1946.5', '1982.2', '2000.6', '1714.8', '1715.9', '1494.4', '1496.1', '2188', '1489.2', '1478.2', '1479.9', '1482.4', '1488.4', '1538.7', '1527.8', '1511.8', '1490.9', '1425.4', '1426.9', '1428.5', '1430.1', '1431.7', '1406.8', '1414.5', '1419.9', '1455.8', '1458.2', '1462.3', '1464', '1465.6', '1469', '1440.4', '1442', '1445.2', '2182.4', '1449.3', '1453.3']

4. Pearson Correlation:

100 fitur - ['1047.1', '1046.7', '1458.2', '1461.5', '1450.9', '1451.7', '1459.1', '1452.5', '1446.8', '1446', '1045', '1453.3', '1459.9', '1460.7', '1454.1', '1051.8', '1462.3', '1450.1', '1457.4', '1445.2', '1051', '1037.9', '1447.6', '1455', '1456.6', '1455.8', '1442.8', '1463.2', '1449.3', '1036.7', '1448.5', '1441.2', '1442', '1443.6', '1444.4', '1464', '1050.5', '1047.6', '1060', '1464.8', '1052.2', '1048.8', '1470.6', '1054.4', '1049.3', '1038.3', '1466.5', '1469', '1440.4', '1465.6', '1471.5', '1045.5', '1468.1', '1469.8', '1467.3', '1437.2', '1040.4', '1042.5', '1439.6', '1438', '1059.6', '1050.1', '1033.8', '1051.4', '1046.3', '1057.8', '1032.6', '1438.8', '1472.3', '1049.7', '1054', '1033.4', '1473.1', '1058.3', '1436.4', '1056.1', '1056.5', '1038.8', '1040', '1474', '1034.6', '1036.3', '1031.7', '1039.6', '1042.9', '1048.4', '1433.2', '1434', '1435.6', '1037.5', '1434.8', '1043.8', '1474.8', '1044.6', '1037.1', '1060.4', '1044.2', '1048', '1476.5', '1040.8']

Kategori *Wrapper*:

1. Sequential Forward Selection

20 fitur - ['1043.4', '1045.9', '1047.1', '1051', '1052.2', '1052.7', '1054', '1070.5', '1085.3', '1085.7', '1087.1', '1087.6', '1089.8', '1094.4', '1096.3', '1422.2', '1427.7', '1446.8', '1654.6', '1732']

40 fitur - ['1043.4', '1045.9', '1047.1', '1050.5', '1051', '1052.2', '1052.7', '1054', '1060', '1070.5', '1084.8', '1085.3', '1085.7', '1087.1', '1087.6', '1089.4', '1089.8', '1094.4', '1094.9', '1096.3', '1101.4', '1104.2', '1391', '1422.2', '1424.6', '1425.4', '1427.7', '1439.6', '1446', '1446.8', '1452.5', '1454.1', '1464', '1540.5', '1555.3', '1654.6', '1660.9', '1666.3', '1669.5', '1732']

60 fitur - ['1041.7', '1043.4', '1045.9', '1047.1', '1047.6', '1050.5', '1051', '1052.2', '1052.7', '1054', '1057.4', '1060', '1070.5', '1081.7', '1084.8', '1085.3', '1085.7', '1086.6', '1087.1', '1087.6', '1089.4', '1089.8', '1091.2', '1094.4', '1094.9', '1096.3', '1100.5', '1101.4', '1104.2', '1115.2', '1149.3', '1391', '1397.7', '1406', '1414.5', '1422.2', '1424.6', '1425.4', '1427.7', '1436.4', '1439.6', '1446', '1446.8', '1452.5', '1454.1', '1464', '1466.5', '1485.8', '1490.1', '1494.4', '1521.6', '1540.5', '1555.3', '1559.1', '1654.6', '1660.9', '1666.3', '1669.5', '1681.4', '1732']

80 fitur - ['1009.6', '1041.7', '1043.4', '1045.9', '1047.1', '1047.6', '1048.4', '1050.5', '1051', '1052.2', '1052.7', '1054', '1057.4', '1060', '1062.6', '1065.7', '1070.5', '1081.7', '1084.8', '1085.3', '1085.7', '1086.6', '1087.1', '1087.6', '1088.9', '1089.4', '1089.8', '1091.2', '1094.4', '1094.9', '1096.3', '1098.6', '1100.5', '1101.4', '1104.2', '1115.2', '1149.3', '1390.2', '1391', '1397.7', '1406', '1411.4', '1414.5', '1422.2', '1424.6', '1425.4', '1427.7', '1436.4', '1439.6', '1446', '1446.8', '1450.9', '1452.5', '1454.1', '1464', '1466.5', '1485', '1485.8', '1490.1', '1494.4', '1518.9', '1521.6', '1527.8', '1528.7', '1536', '1536.9', '1540.5', '1555.3', '1559.1', '1560', '1654.6', '1659.9', '1660.9', '1664.1', '1666.3', '1669.5', '1674.9', '1680.3', '1681.4', '1732']

100 fitur - ['1009.6', '1020.8', '1023.2', '1040.8', '1041.7', '1043.4', '1045.9', '1047.1', '1047.6', '1048.4', '1050.5', '1051', '1052.2', '1052.7', '1054', '1057.4', '1060', '1062.6', '1065.7', '1070.5', '1081.7', '1084.8', '1085.3', '1085.7', '1086.6', '1087.1', '1087.6', '1088.9', '1089.4', '1089.8', '1091.2', '1094.4', '1094.9', '1096.3', '1098.6', '1100.5', '1101.4', '1104.2', '1114.2', '1115.2', '1143.2', '1149.3', '1158', '1191.5', '1200.3', '1299', '1323.5', '1382.1', '1390.2', '1391', '1397.7', '1406', '1409.1', '1411.4', '1414.5', '1422.2', '1424.6', '1425.4', '1427.7', '1436.4', '1438', '1439.6',

'1446', '1446.8', '1450.9', '1452.5', '1454.1', '1464', '1466.5', '1485', '1485.8', '1490.1', '1490.9', '1494.4', '1518.9', '1521.6', '1527.8', '1528.7', '1536', '1536.9', '1540.5', '1555.3', '1559.1', '1560', '1563.8', '1654.6', '1659.9', '1660.9', '1664.1', '1666.3', '1669.5', '1674.9', '1680.3', '1681.4', '1732', '1741.3', '1763.8', '1900.8', '2331.6', '2367.8']

2. Backward Elimination

20 fitur - ['1138.2', '1139.2', '1132.2', '1358.9', '1134.2', '1361', '1276', '1270.9', '1273.4', '1266.6', '1263.5', '1269.7', '1662', '1657.8', '1000.3', '1282.9', '1629.6', '1669.5', '1002.2', '1678.1']

40 fitur ['2500.2', '1857.3', '1732', '1768.6', '1785.6', '2102.8', '2000.6', '1496.1', '1478.2', '1538.7', '1455.8', '1442', '1671.7', '1612.4', '1623.5', '1600.5', '1595.5', '1591.6', '1586.7', '1621.5', '1620.5', '2414.1', '1615.4', '2411.8', '1557.2', '1578', '1567.6', '1679.2', '1678.1', '1674.9', '1672.7', '1643.1', '1637.9', '1632.7', '1648.3', '1668.4', '1666.3', '1665.2', '2376.5', '2365.6']

60 fitur - ['2500.2', '1857.3', '1838.8', '1837.5', '1905', '1896.7', '1913.5', '1885.6', '1723.9', '1726.2', '1732', '1798', '1768.6', '1785.6', '2102.8', '2000.6', '1496.1', '1478.2', '1538.7', '1425.4', '1455.8', '1442', '1669.5', '1671.7', '1711.4', '1612.4', '1623.5', '1600.5', '2350.6', '1595.5', '1592.6', '1591.6', '1586.7', '1621.5', '1620.5', '2414.1', '1615.4', '2411.8', '1561.9', '1557.2', '1551.6', '1580.9', '1578', '1567.6', '1679.2', '1678.1', '1674.9', '1672.7', '1645.1', '1643.1', '1637.9', '1632.7', '1648.3', '1668.4', '1666.3', '1665.2', '1664.1', '2376.5', '1652.5', '2365.6']

80 fitur - Fitur yang diambil adalah:
['2500.2', '1845.4', '1857.3', '1838.8', '1837.5', '1905', '1896.7', '1913.5', '1885.6', '1723.9', '1726.2', '1732', '1789.3', '1798', '1805.5', '1768.6', '1785.6', '1923.4', '2102.8', '2109.6', '2067.6', '2000.6', '1496.1', '1478.2', '1538.7', '1425.4', '1455.8', '1442', '1669.5', '1671.7', '1706.9', '1711.4', '1612.4', '1623.5', '1493.5', '1600.5', '2350.6', '1595.5', '1592.6', '1591.6', '2418.6', '1586.7', '1621.5', '1620.5', '2414.1', '1615.4', '2411.8', '1611.4', '1584.8', '1561.9', '1559.1', '1557.2', '1551.6', '1563.8', '1566.6', '1580.9', '1578', '1574.2', '1567.6', '1625.5', '1679.2', '1678.1', '1674.9', '1709.1', '1672.7', '2385.2', '1645.1', '1643.1', '1637.9', '1632.7', '1648.3', '1668.4', '1666.3', '1665.2', '1664.1', '2376.5', '1652.5', '2365.6', '1585.8', '1446.8']

100 fitur - ['2500.2', '1845.4', '2184.3', '1857.3', '1838.8', '1820.7', '1837.5', '1905', '1896.7', '1913.5', '1874.7', '1885.6', '1891.1', '1723.9', '1726.2', '1732', '1789.3', '1798', '1805.5', '1768.6', '1785.6', '1923.4', '2102.8', '2109.6', '2067.6', '2116.5', '1936.3', '2000.6', '1496.1', '1489.2', '1478.2', '1538.7', '1425.4', '1419.9', '1455.8', '1465.6', '1442', '1669.5', '1671.7', '1706.9', '1711.4', '1612.4', '1623.5', '1493.5', '1600.5', '2350.6', '1595.5', '1592.6', '1591.6', '2418.6', '1586.7', '1621.5', '1620.5', '2414.1', '1615.4', '2411.8', '1611.4', '1584.8', '1565.7', '1561.9', '1560', '1559.1', '1557.2', '1551.6', '1563.8', '1566.6', '1580.9', '1578', '1574.2', '1573.3', '1570.4', '1569.5', '1567.6', '1625.5', '1684.7', '1679.2', '1678.1', '1674.9', '1709.1', '1672.7', '2385.2', '1645.1', '1643.1', '1637.9', '1632.7', '1648.3', '1668.4', '1666.3', '1665.2', '1664.1', '2376.5', '2374.3', '1652.5', '1651.4', '2365.6', '1585.8', '2361.3', '1448.5', '1447.6', '1446.8']

3. Recursive Feature Elimination (RFE)

20 fitur - ['1393.2', '1400.7', '1509.2', '1535.1', '1562.8', '1567.6', '1590.6', '1786.9', '1834.9', '1838.8', '1841.4', '1848', '1857.3', '1863.9', 2121.7', '2123.5', '2302.6', '2304.7', '2352.8', '2354.9']

40 fitur - ['1393.2', '1393.9', '1400.7', '1431.7', '1435.6', '1436.4', '1447.6', '1449.3', '1509.2', '1535.1', '1562.8', '1565.7', '1567.6', '1572.3', '1581.9', '1590.6', '1597.5', '1775.8', '1786.9', '1791.8', '1798', '1810.6', '1811.8', '1834.9', '1838.8', '1841.4', '1848', '1857.3', '1863.9', '1868', '2121.7', '2123.5', '2155.2', '2157', '2302.6', '2304.7', '2325.3', '2333.7', '2352.8', '2354.9']

60 fitur - ['1393.2', '1393.9', '1400.7', '1431.7', '1435.6', '1436.4', '1447.6', '1449.3', '1504.8', '1509.2', '1535.1', '1539.6', '1562.8', '1565.7', '1567.6', '1572.3', '1580', '1581.9', '1586.7', '1590.6', '1592.6', '1597.5', '1742.4', '1775.8', '1786.9', '1791.8', '1798', '1810.6', '1811.8', '1834.9', '1838.8', '1841.4', '1848', '1857.3', '1863.9', '1868', '1881.5', '1895.3', '1905', '1907.8', '1936.3', '1995.9', '2070.9', '2089.2', '2092.6', '2116.5', '2121.7', '2123.5', '2148.1', '2155.2', '2157', '2173.3', '2195.4', '2199.1', '2302.6', '2304.7', '2325.3', '2333.7', '2352.8', '2354.9']

80 fitur - ['1393.2', '1393.9', '1399.2', '1400.7', '1431.7', '1435.6', '1436.4', '1447.6', '1449.3', '1459.1', '1477.3', '1497', '1504.8', '1509.2', '1535.1', '1539.6', '1562.8', '1565.7', '1567.6', '1572.3', '1577.1', '1580', '1581.9', '1586.7', '1590.6', '1592.6', '1595.5', '1597.5', '1742.4', '1748.3', '1767.4', '1775.8', '1783.2', '1786.9', '1791.8', '1798', '1810.6', '1811.8', '1834.9', '1838.8', '1841.4', '1848', '1857.3', '1863.9', '1868', '1881.5', '1895.3', '1905', '1907.8', '1936.3', '1995.9', '2013', '2025.6', '2070.9', '2089.2', '2090.9','2092.6', '2097.7', '2102.8', '2116.5', '2121.7', '2123.5', '2135.7', '2148.1', '2155.2', '2157', '2173.3', '2195.4', '2199.1', '2302.6', '2304.7', '2325.3', '2333.7', '2352.8', '2354.9', '2357', '2420.9', '2434.5', '2436.8', '2478.7']

100 fitur - ['1393.2', '1393.9', '1399.2', '1400.7', '1431.7', '1435.6', '1436.4', '1440.4', '1446', '1447.6', '1449.3', '1459.1', '1477.3', '1484.1', '1497', '1500.4', '1504.8', '1509.2', '1526', '1533.3', '1535.1', '1539.6', '1547', '1562.8', '1565.7', '1567.6', '1571.4', '1572.3', '1577.1', '1580', '1581.9', '1586.7', '1590.6', '1592.6', '1595.5', '1597.5', '1742.4', '1748.3', '1767.4', '1771', '1775.8', '1783.2', '1786.9', '1791.8', '1798', '1810.6', '1811.8', '1834.9', '1838.8', '1841.4', '1848', '1857.3', '1863.9', '1868', '1881.5', '1889.7', '1895.3', '1905', '1907.8', '1936.3', '1995.9', '2008.3', '2013', '2025.6', '2070.9', '2084.2', '2089.2', '2090.9', '2092.6', '2097.7', '2102.8', '2116.5', '2121.7', '2123.5', '2135.7', '2148.1', '2155.2', '2157', '2173.3', '2195.4', '2199.1', '2214.1', '2221.7', '2242.8', '2294.5', '2302.6', '2304.7', '2325.3', '2327.4', '2333.7', '2352.8', '2354.9', '2357', '2389.6', '2420.9', '2434.5', '2436.8', '2478.7', '2481.1', '2490.6']

Kategori *Embedded*:

1. LASSO:

55 fitur - ['1001.8', '1139.7', '1141.2', '1142.2', '1142.7', '1143.2', '1143.7', '1486.7', '1489.2', '1490.9', '1491.8', '1492.7', '1493.5', '1494.4', '1495.2', '1497.8', '1498.7', '1499.6', '1501.3', '1502.2', '1503.9', '1504.8', '1511.8', '1536.9', '1543.3', '1547.9', '1548.8', '1552.5', '1553.5', '1555.3', '1556.3', '1557.2', '1558.1', '1562.8', '1569.5', '1570.4', '1573.3', '1574.2', '1582.9', '1583.8', '1903.6', '1905', '1914.9', '2302.6', '2304.7', '2346.4', '2348.5', '2378.7', '2380.8', '2387.4', '2389.6', '2394', '2396.2', '2418.6', '2476.3']

2. Elastic Net

196 fitur - ['999.9', '1000.3', '1000.7', '1001.4', '1001.8', '1008.1', '1010.8', '1015.6', '1017.2', '1019.2', '1025.6', '1029.3', '1032.6', '1319.5', '1320.1', '1320.8', '1321.5', '1322.1', '1322.8', '1323.5', '1324.2', '1324.9', '1325.5', '1326.2', '1326.9', '1327.6', '1328.2', '1328.9', '1329.6', '1330.3', '1331', '1331.7', '1332.3', '1333', '1333.7', '1334.4', '1335.1', '1335.8', '1336.5', '1337.1', '1337.8', '1338.5', '1339.2', '1339.9', '1340.6', '1341.3', '1342', '1342.7', '1343.4', '1344.1', '1344.8', '1345.5', '1346.2', '1346.9', '1347.6', '1348.3', '1349', '1349.7', '1350.4', '1351.1', '1351.8', '1352.5', '1353.2', '1353.9', '1354.6', '1355.3', '1356', '1356.7', '1357.5', '1358.2', '1358.9', '1359.6', '1360.3', '1361', '1361.7', '1362.4', '1363.2', '1363.9', '1364.6', '1365.3', '1366', '1366.8', '1367.5', '1368.2', '1368.9', '1369.6', '1370.4', '1371.1', '1371.8', '1372.5', '1373.3', '1374', '1374.7', '1375.5', '1376.2', '1376.9', '1377.6', '1378.4', '1379.1', '1379.8', '1380.6', '1381.3', '1382.1', '1382.8', '1383.5', '1384.3', '1385', '1385.7', '1386.5', '1387.2', '1388', '1388.7', '1389.5', '1390.2', '1391', '1391.7', '1392.4', '1393.2', '1393.9', '1394.7', '1395.4', '1396.2', '1396.9', '1397.7', '1398.5', '1399.2', '1400', '1400.7', '1401.5', '1402.2', '1403', '1403.8', '1404.5', '1405.3', '1406', '1406.8', '1407.6', '1408.3', '1409.1', '1409.9', '1410.6', '1411.4', '1412.2', '1412.9', '1413.7', '1414.5', '1415.2', '1416', '1422.2', '1423', '1423.8', '1426.1', '1426.9', '1429.3', '1430.1', '1430.9', '1432.4', '1433.2', '1437.2', '1441.2', '1442.8', '1445.2', '1446', '1450.9', '1451.7', '1458.2', '1461.5', '1468.1', '1469', '1469.8', '1470.6', '1471.5', '1474', '1475.7', '1476.5', '1480.7', '1481.6', '1486.7', '1489.2', '1490.9', '1491.8', '1493.5', '1494.4', '1495.2', '1926.3', '1927.7', '2474', '2476.3', '2478.7', '2481.1', '2488.2', '2490.6', '2493', '2495.4', '2497.8', '2500.2']