# BAB V HASIL DAN PENGUJIAN

Bab ini berisi tentang hasil, analisis, dan pengujian sistem prediksi cuaca di Kabupaten Sambas menggunakan algoritma *random forest*.

## Hasil *Data Understanding*

Pada tahap ini berisi hasil dari tahapan-tahapan *data understanding* yang terdapat pada perancangan sebelumnya dengan tujuan untuk memahami struktur, kualitas, serta karakteristik dari dataset yang digunakan.

### Sumber data

Dataset diperoleh penulis melalui laman resmi BMKG http://dataonline.bmkg.go.id/home yang berisi data cuaca harian Kabupaten Sambas dengan rentang data tahun 2023-2024 dengan jumlah 366 data dan 11 atribut.

Tabel 5. 1 Hasil Dataset

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fitur** | **Keterangan** | **Jenis Data** |
| 1. | Tanggal | Berisi data waktu, dalam hal ini berformat DD-MM-YYYY | Kategorikal |
| 2. | Tn | Berisi datatemperatur minimum dalam satuan derajat celcius | Numerik |
| 3. | Tx | Berisidata temperatur maksimum dalam satuan derajat celcius | Numerik |
| 4. | Tavg | Berisi datatemperatur rata-rata dalam satuan derajat celcius | Numerik |
| 5. | RH\_avg | Berisi datakelembapan rata-rata dalam satuan derajat celcius | Numerik |
| 6. | RR | Berisidata curah hujan dalam satuan milimeter | Numerik |
| 7. | ss | Berisidata lamanya penyinaran matahari dalam satuan jam | Numerik |
| 8. | ff\_x | Berisidata nilai maksimum kecepatan angin yang tercatat dalam satuan meter per detik | Numerik |
| 9. | ddd\_x | Berisi data arah angin yang tercatat pada saat kecepatan angin mencapai nilai maksimum dalam satuan derajat | Numerik |
| 10. | ff\_avg | Berisi data nilai kecepatan angin rata-rata dalam satuan meter per detik | Numerik |
| 22 | ddd\_car | Berisi data arah angin terbanyak | Kategorikal |

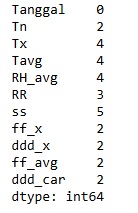
Berikut adalah kode program yang digunakan pada tahap import data:

|  |
| --- |
| **import** pandas **as** pd  ​  data = pd.read\_excel("dataset cuaca.xlsx", header=1, index\_col=**None**)  data.head() |

### Hasil Eksplorasi Nilai Yang Tidak Valid

Setelah dataset dideklarasikan menjadi variable data, dilakukan pengecekan terhadap data yang bernilai tidak valid. Maksud dari nilai yang tidak valid dalam penelitian ini ialah data yang kosong, dan data yang berisi inputan 8888 dan 9999. Berikut kode program yang digunakan untuk melihat data yang tidak valid:

|  |
| --- |
| data.isnull().sum() |



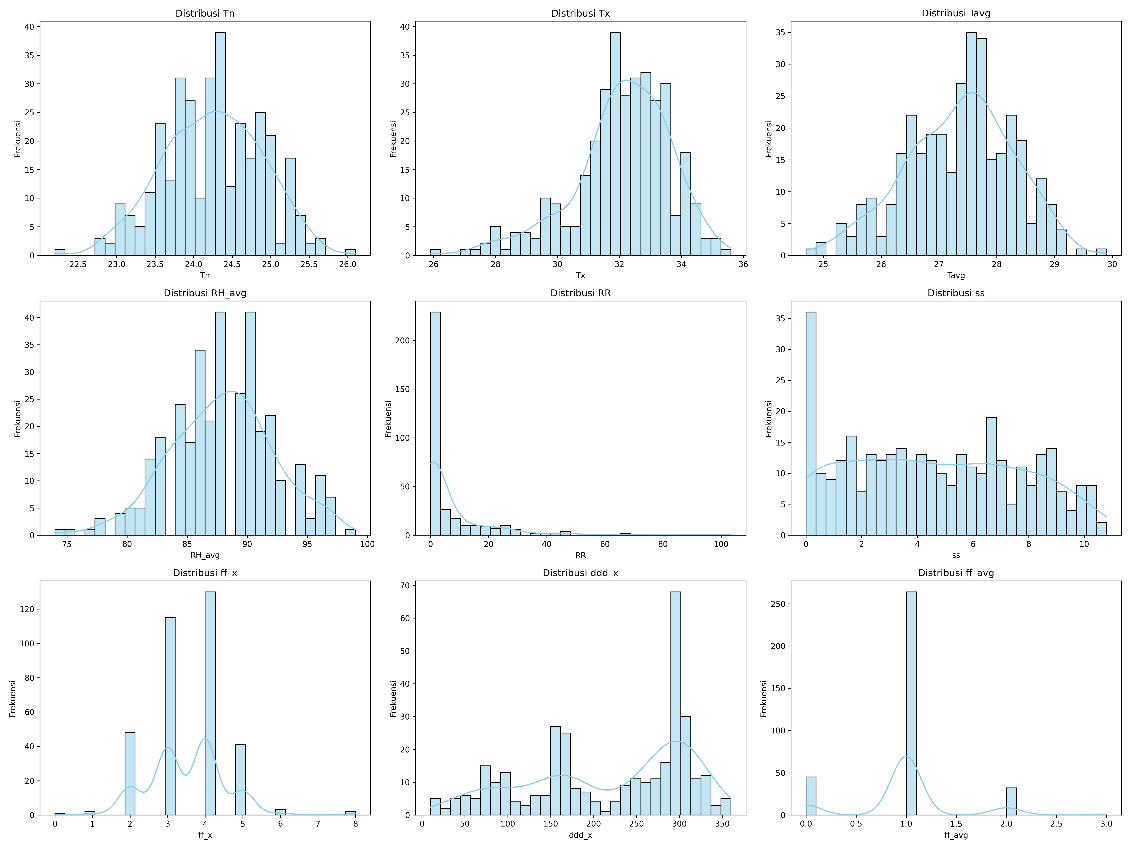
Gambar 5. 1 Hasil Eksplorasi Nilai Yang Tidak Valid

Diketahui ada beberapa baris data yang berisikan nilai tidak valid, selanjutnya data yang berisi nilai tidak valid akan dihapus.

* + 1. **Hasil Eksplorasi Distribusi Data**

Setelah dilakukan handling missing value, akan dilakukan pengecekan penyebaran data. Berikut kode program yang digunakan untuk melihat data yang tidak valid. Berikut kode program yang digunakan untuk menyiapkanhistogram nya:

|  |
| --- |
| **import** matplotlib.pyplot **as** plt  **import** seaborn **as** sns  ​  # Set ukuran plot  plt.figure(figsize=(20, 15))  ​  # Loop untuk setiap kolom dan buat histogram  **for** i, col **in** enumerate(numerik\_cols, 1):     plt.subplot(3, 3, i)     sns.histplot(data=data, x=col, kde=**True**, bins=30, color='skyblue')     plt.title(f'Distribusi {col}', fontsize=12)     plt.xlabel(col)     plt.ylabel('Frekuensi')  ​  plt.tight\_layout()  plt.savefig('distribusi\_histogram.png', dpi=300)  plt.show()  ​ |



Gambar 5. 2 Hasil Data

Gambar 5.2 merupakan visualisasi histogram distribusi data yang hanya bertipe numerik*.*

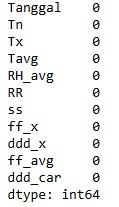
## Hasil *Data* *Preprocessing*

Pada tahap ini akan dilakukan tahapan *preprocessing, preprocessing* adalah proses untuk mempersiapkan data sebelum sebelum dilatih oleh model.

### *Cleaning*

Pada tahap ini akan dilakukan tahapan menangani data dengan nilai yang tidak valid, dalam hal ini akan dilakukan penghapusan agar dikemudian tidak menjadi *noise* pada pemodelan. Berikut kode program yang digunakan untuk menangani nilai yang tidak valid:

|  |
| --- |
| data = data.dropna().reset\_index(drop=**True**)  data = data[~data['RR'].isin([8888, 9999])]  data.isnull().sum() |

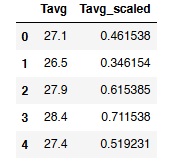


Gambar 5. 3 Hasil Handling Missing Value

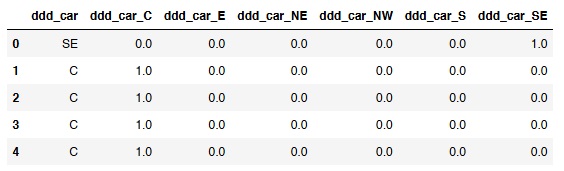
Setelah dilakukan tahapan *cleaning,* data yang digunakan untuk tahapan *data transformation* berjumlah 342 baris.

### *Data Transformation*

*Data Transformation* adalah tahapan dalam *preprocessing* yang berfungsi untuk mengubah dan menyesuaikan format data agar siap digunakan dalam analisis atau pemodelan machine learning. Strategi yang digunakan adalah untuk data yang bertipe numerikal menggunakan pendekatan minmax, sedangkan data yang bertipe kategorikal menggunakan one hot dengan bantuan pustka sklearn. Berikut adalah contoh perbandingan nya:



Gambar 5. 5 Hasil Data Transformation (Numerikal)



Gambar 5. 6 Hasil Data Transformation (Kategorikal)

### Feature Engineering

Pada tahapan ini, dilakukan pemilihan fitur dan target yang paling relevan untuk meningkatkan performa model. dalam penulisan ini target yang dipilih oleh penulis ialah Tavg, dan atribut yang dihapus ialah Tn, Tx dan Tanggal.

Tabel 5. 2 Hasil Feature Engineering

|  |  |  |  |
| --- | --- | --- | --- |
| **No** | **Fitur** | **Keterangan** | **Jenis Data** |
| 1. | Tavg | Berisi datatemperatur rata-rata dalam satuan derajat celcius | Target |
| 2. | RH\_avg | Berisi datakelembapan rata-rata dalam satuan derajat celcius | Fitur |
| 3. | RR | Berisidata curah hujan dalam satuan milimeter | Fitur |
| 4. | ss | Berisidata lamanya penyinaran matahari dalam satuan jam | Fitur |
| 5. | ff\_x | Berisidata nilai maksimum kecepatan angin yang tercatat dalam satuan meter per detik | Fitur |
| 6. | ddd\_x | Berisi data arah angin yang tercatat pada saat kecepatan angin mencapai nilai maksimum dalam satuan derajat | Fitur |
| 7. | ff\_avg | Berisi data nilai kecepatan angin rata-rata dalam satuan meter per detik | Fitur |
| 8. | ddd\_car | Berisi data arah angin terbanyak | Fitur |

## Hasil Pemodelan

Langkah selanjutnya yaitu pemodelan. Pemodelan menggunakan metode *Random Forest Regressor*. Adapun hasil dari tahapan-tahapan nya sebagai berikut:

1. *Spliting Data*

Data pada tahap ini ialah data yang telah melalui tahapan *preprocessing*. Adapun hasil dalam tahapan ini ialah 239 untuk data latih, dan 103 untuk data uji dengan bantuan fungsi train\_test\_split dari pustaka sklearn.

1. *Cross Validation*

Setelah data dibagi menjadi data latih (239) dan data uji (103), proses cross-validation dilakukan pada data latih menggunakan teknik k-fold dengan nilai k = 3.

1. *Pengujian Model Dan Tuning*

Model yang digunakan dalam penelitian ini adalah *Random Forest Regressor*. Untuk mendapatkan performa model yang optimal, dilakukan proses tuning hyperparameter menggunakan metode *Grid Search Cross Validation (GridSearchCV)*. Teknik ini mencoba berbagai kombinasi parameter, pada penelitian ini penulis menggunakan strategi sebagai berikut.

Tabel 5. 3 Parameter Pemodelan Dan Tuning

|  |  |
| --- | --- |
| **Parameter** | **Nilai** |
| n\_estimators | 100, 150, 200 |
| max\_depth | 20, 50, 80 |
| max\_features | 0.3, 0.6, 0.8 |
| min\_samples\_leaf | 1, 5, 10 |

## Hasil Evaluasi Pemodelan

Hasil tuning menunjukkan bahwa kombinasi parameter terbaik yang diperoleh adalah n\_estimators 100, max\_depth 20, max\_features 0.8, min\_samples\_leaf 5. Model dengan parameter tersebut menghasilkan skor evaluasi sebagai berikut:

Tabel 5. 7 Hasil Evaluasi

|  |  |
| --- | --- |
| **Matrik Evaluasi** | **Hasil** |
| MSE | 0.1996 |
| RMSE | 0.4468 |
| MAE | 0.3525 |
| R2 | 74% |

## Hasil Implementasi Website

Hasil dari pemodelan selanjutnya digunakan untuk membuat *integrasi* dari model ke *website*, pada fase ini ditampilkan juga hasil dari pembangunan *website*.

### Hasil *Integrasi*

Pada fase ini dijelaskan tahap *integrasi* cara menyimpan dan menggunakan model yang disimpan, pertama adalah menyimpan pemodelan ipynb yang dibangun kedalam file pkl (*Pickle*) menggunakan *library* Python dengan kode program dibawah:

|  |
| --- |
| save\_model(model.best\_estimator\_, "rfr\_cuaca.pkl")  ​ |

Kode program di atas berfungsi untuk menyimpan pemodelan *Random Forest*  kedalam file *pickle.*

Setelah berhasil menyimpan pemodelan dalam file pkl langkah selanjutnya yaitu dengan memanggil file pkl tersebut kedalam aplikasi *website* yang dibangun menggunakan kerangka kerja *python* yaitu *Streamlit* kedalam file .py seperti pada kode program dibawah:

|  |
| --- |
| rfr = pickle.load(open("rfr\_cuaca.pkl", "rb")) |

Kode program di atas berfungsi untuk memanggil file *pickle* kedalam *website* menggunakan pustaka *pickle*.

### Hasil Halaman Beranda

Halaman Beranda pada website prediksi cuaca yang diimplementasikan menggunakan *Streamlit* berfungsi sebagai halaman utama yang pertama kali diakses oleh pengguna. Pada halaman ini, pengguna disambut dengan tampilan yang sederhana dan informatif, yang memudahkan mereka untuk langsung melakukan input data cuaca yang diperlukan.



Gambar 5. 7 Halaman Beranda