

Optimisasi Prediksi Cuaca di Seattle Menggunakan Metode Naïve Bayes dengan Metodologi CRISP-DM

Rifai Nugroho¹, Kristanto², Rafly Sidiq H.³

* Program Studi Teknik Informatika,
Universitas Komputer Indonesia
10121295¹, 101213098², 10121311³

ABSTRAK

Penelitian ini bertujuan untuk mengoptimalkan prediksi cuaca di Seattle menggunakan metode Naïve Bayes dengan menerapkan metodologi CRISP-DM. Data yang digunakan melibatkan data cuaca tahunan di Seattle dari tahun 2012 hingga 2015. Tahap-tahap CRISP-DM, yaitu Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation, dan Deployment, diterapkan secara sistematis untuk meningkatkan kualitas prediksi cuaca. Metode Naïve Bayes digunakan dalam tahap Modeling untuk mengklasifikasikan dan memprediksi kondisi cuaca berdasarkan data historis. Evaluasi model dilakukan dengan menggunakan Confusion Matrix dan metrik evaluasi lainnya untuk mengukur keakuratan prediksi. Hasil penelitian ini diharapkan dapat memberikan kontribusi dalam pengembangan sistem prediksi cuaca yang lebih andal.

Kata Kunci: Seattle, Naïve Bayes, CRISP-DM, Data Cuaca Tahunan, Prediksi Kondisi Cuaca

1. PENDAHULUAN

1.1. Latar Belakang

Cuaca merupakan aspek integral dalam kehidupan sehari-hari yang memiliki dampak signifikan pada berbagai aktivitas manusia. Kemampuan untuk memprediksi cuaca dengan akurat memberikan keuntungan besar, terutama dalam perencanaan kegiatan luar ruangan, manajemen sumber daya, dan pengambilan keputusan berbasis cuaca. Seattle, sebagai kota metropolitan yang dinamis, membutuhkan prediksi cuaca yang handal untuk mendukung aktivitas warganya.

Dalam beberapa tahun terakhir, teknik prediksi cuaca menggunakan pendekatan data mining, khususnya algoritma Naive Bayes, telah menjadi semakin relevan dan efektif. Penggunaan metode ini memungkinkan analisis yang lebih mendalam terhadap pola cuaca berdasarkan data historis, sehingga memberikan hasil prediksi yang lebih akurat.

1.2. Tujuan Penelitian

Tujuan utama dari penelitian ini adalah menerapkan algoritma Naive Bayes dalam memprediksi keputusan cuaca di Seattle. Melalui pendekatan ini, kami bertujuan untuk menghasilkan model prediksi cuaca yang dapat memberikan informasi berharga kepada masyarakat, pihak berkepentingan di bidang transportasi, pariwisata, dan sektor-sektor lain yang sangat dipengaruhi oleh kondisi cuaca.

1.3. Relevansi Penelitian

Dengan meningkatnya kebutuhan akan prediksi cuaca yang akurat, implementasi metode Naive Bayes dalam analisis data cuaca Seattle memberikan kontribusi penting dalam pengembangan sistem prediksi yang dapat diandalkan. Hasil penelitian ini diharapkan dapat memberikan pandangan yang lebih baik tentang cara mengoptimalkan pemanfaatan data cuaca historis untuk meramalkan kondisi cuaca di masa depan.

1.4. Metodologi CRISP-DM

Dalam penelitian ini, kami mengadopsi metodologi CRISP-DM (Cross-Industry Standard Process for Data Mining) sebagai kerangka kerja untuk merinci langkah-langkah dalam proses prediksi cuaca. Pendekatan ini memberikan struktur yang sistematis dan terukur, memungkinkan pemahaman yang lebih baik terhadap setiap tahap pengolahan data dan implementasi algoritma Naive Bayes.

1.5. Ruang Lingkup Penelitian

Penelitian ini akan menggunakan data cuaca Seattle dari tahun 2012 hingga 2015 sebagai sumber utama informasi. Data tersebut melibatkan sejumlah atribut, termasuk suhu, kelembaban, kecepatan angin, dan kondisi cuaca umum. Proses analisis akan mencakup pemahaman bisnis, eksplorasi data, persiapan data, pembangunan model, evaluasi, dan penyebaran hasil.

Dengan latar belakang ini, kami berharap bahwa penelitian ini dapat memberikan wawasan yang berharga dan aplikatif dalam konteks prediksi cuaca di Seattle menggunakan metode Naive Bayes dan metodologi CRISP-DM.

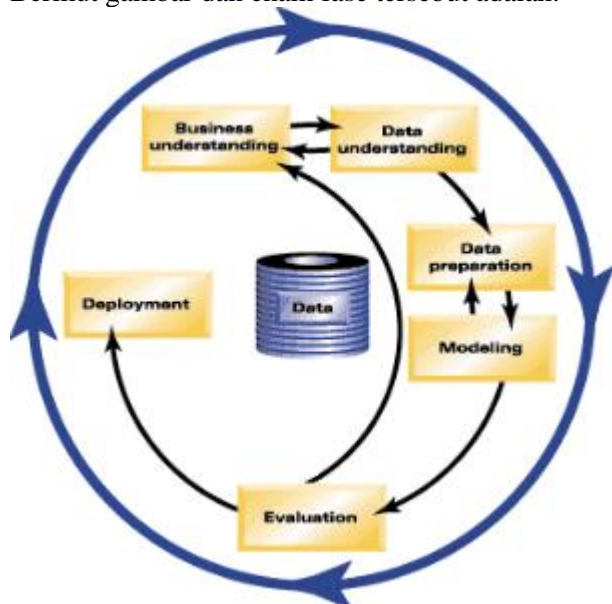
2. Metodologi

Dalam pengembangan model prediksi cuaca di Seattle, kami menerapkan Metode CRISP-DM (Cross-Industry Standard Process for Data Mining) sebagai landasan metodologi. CRISP-DM memberikan kerangka kerja yang terstruktur untuk menavigasi langkah-langkah dalam proses penggalian data. Berikut adalah rincian perluasan pada setiap fase dalam Metode CRISP-DM:

2.1. Metode CRISP-DM

Dalam metode analisis data ini digunakan metode Cross-Industry Standart Process for Data Mining (CRISP-DM) karena proses ini salah satu tujuannya untuk menemukan pola yang menarik dan bermakna dalam Data Mining

Berikut gambar dan enam fase tersebut adalah:



Gambar 1. Proses Data Mining CRISP-DM

2.1.1. Business Understanding

Dalam tahap ini, fokus kami adalah memahami dengan mendalam tujuan bisnis di balik prediksi cuaca. Kami melakukan analisis kebutuhan pemangku kepentingan, memastikan bahwa tujuan prediksi cuaca sesuai dengan ekspektasi dan kepentingan bisnis yang ada.

2.1.2. Data Understanding

Langkah pertama yang kami tempuh adalah menganalisis dataset cuaca Seattle dari tahun 2012 hingga 2015. Kami melakukan eksplorasi data untuk memahami distribusi, pola, dan karakteristik utama dari dataset yang akan digunakan untuk proses prediksi.

2.1.3. Data Preparation

Persiapan data merupakan tahap kritis dalam memastikan keberhasilan prediksi. Kami melakukan pembersihan data secara menyeluruh, menangani nilai yang hilang, outliers, dan mengonversi data ke format yang sesuai. Hal ini dilakukan untuk memastikan data yang digunakan dalam proses selanjutnya adalah data yang berkualitas tinggi.

2.1.4. Modeling

Algoritma Naive Bayes dipilih sebagai model prediksi cuaca. Kami menjelajahi berbagai parameter dan konfigurasi untuk memastikan model ini sesuai dengan karakteristik data cuaca Seattle. Proses ini mencakup pelatihan model dengan dataset yang telah dipersiapkan.

2.1.5. Evaluation

Setelah model terlatih, kami menguji kinerjanya menggunakan data uji. Evaluasi dilakukan dengan memeriksa tingkat akurasi prediksi, memastikan bahwa model dapat memberikan hasil yang dapat diandalkan dan sesuai dengan kebutuhan bisnis.

2.1.6. Deployment

Hasil analisis dan temuan dari model prediksi cuaca disusun dalam laporan yang rinci. Laporan ini akan disajikan kepada pemangku kepentingan untuk memberikan wawasan yang bermanfaat dan mendukung pengambilan keputusan.

2.2. Algoritma Naive Bayes

Algoritma Naive Bayes merupakan salah satu algoritma yang terdapat pada Teknik klasifikasi. Naive Bayes merupakan pengklasifikasian dengan metode probabilitas dan statistik yang dikemukakan oleh ilmuwan Inggris Thomas Bayes, yaitu memprediksi peluang di masa depan berdasarkan pengalaman dimasa sebelumnya sehingga dikenal sebagai Teorema Bayes. Teorema tersebut dikombinasikan dengan Naive dimana diasumsikan kondisi antar atribut saling bebas. Klasifikasi Naive Bayes diasumsikan bahwa ada atau tidak ciri tertentu dari sebuah kelas tidak ada hubungannya dengan ciri dari kelas lainnya.

Persamaan dari teorema Bayes adalah :

Persamaan dari teorema Bayes
adalah :
$$P(H|X) = \frac{P(X|H) \cdot P(H)}{P(X)}$$

Keterangan:

X: Sampel data yang memiliki kelas (label) yang tidak diketahui

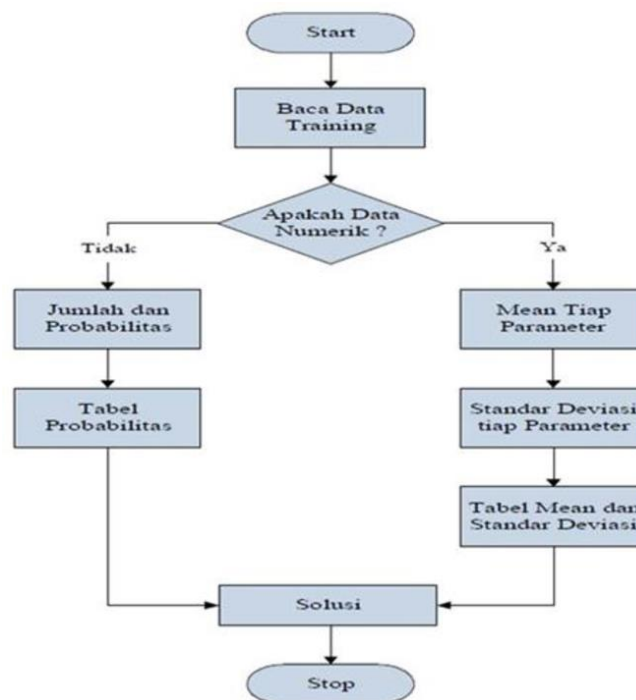
H: Hipotesa bahwa X adalah data kelas (label)

P(H): Peluang dari hipotesa H

P(X): Peluang dari data sampel yang diamati

P(X|H): Peluang dari data sampel X bila diasumsikan bahwa hipotesa benar

Gambar 2. Teorema Naive Bayes



Gambar 3. Alur algoritma Naïve Bayes

Dengan menerapkan CRISP-DM dan algoritma Naive Bayes secara holistik, kami bertujuan untuk menghasilkan model prediksi cuaca yang tidak hanya akurat tetapi juga relevan dengan kebutuhan dan tujuan bisnis yang ditetapkan.

3. Analisis Data

3.1. Pemahaman Data Cuaca Seattle (2012 - 2015)

Dalam fase ini, kami melakukan analisis statistik awal terhadap data cuaca Seattle pada rentang waktu 2012 hingga 2015. Dataset ini terdiri dari total 1461 data.csv, dan setiap entitas data memiliki atribut seperti tanggal (date), tingkat presipitasi (precipitation) dalam satuan milimeter, suhu maksimum (temp_max) dalam derajat Celsius, suhu minimum (temp_min) dalam derajat Celsius, kecepatan angin (wind) dalam meter per detik, dan kondisi cuaca (weather) yang mencakup lima kategori: sun, rain, fog, drizzle, dan snow.

Analisis statistik awal ini dilakukan untuk memahami distribusi data cuaca dan trennya. Kami akan mengeksplorasi variabilitas, nilai rata-rata, dan kemungkinan pola yang muncul dari atribut-atribut tersebut. Visualisasi grafis juga akan digunakan untuk memberikan pemahaman yang lebih mendalam terhadap karakteristik data cuaca Seattle selama periode tersebut.

3.2. Pengelolaan Data

Dengan total 1461 data.csv, langkah pertama dalam pengelolaan data adalah melakukan seleksi data yang relevan untuk implementasi algoritma Naive Bayes. Kami akan mempertimbangkan semua atribut yang telah disebutkan sebelumnya, yaitu tanggal, tingkat presipitasi, suhu maksimum, suhu minimum, kecepatan angin, dan kondisi cuaca.

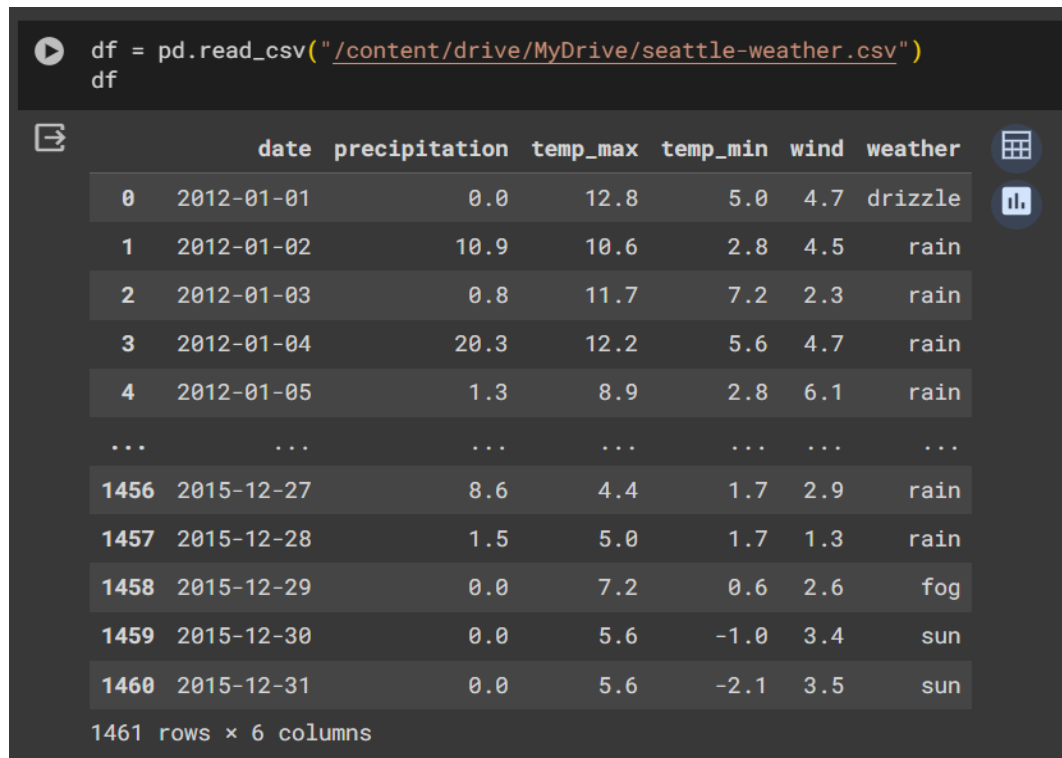
Proses pembersihan data mencakup penanganan nilai yang hilang, penanganan outlier, dan memastikan konsistensi format data. Selain itu, kami juga akan melakukan transformasi data yang diperlukan agar sesuai dengan kebutuhan algoritma Naive Bayes. Misalnya, dapat dilakukan encoding pada atribut kondisi cuaca untuk mengonversi kategori menjadi variabel numerik yang dapat dipahami oleh algoritma.

Dengan pengelolaan data yang cermat, diharapkan bahwa data yang digunakan dalam proses selanjutnya akan memiliki kualitas tinggi dan relevan untuk mendukung prediksi cuaca menggunakan algoritma Naive Bayes.

4. Pemodelan Data

4.1. Implementasi Algoritma Naive Bayes

Langkah selanjutnya dalam metodologi ini adalah implementasi algoritma Naive Bayes untuk memprediksi kondisi cuaca berdasarkan atribut-atribut yang telah dipersiapkan sebelumnya. Data yang telah diolah dan diformat dengan baik akan digunakan sebagai data latih (training) untuk melatih model Naive Bayes.



```
df = pd.read_csv("/content/drive/MyDrive/seattle-weather.csv")
df
```

	date	precipitation	temp_max	temp_min	wind	weather
0	2012-01-01	0.0	12.8	5.0	4.7	drizzle
1	2012-01-02	10.9	10.6	2.8	4.5	rain
2	2012-01-03	0.8	11.7	7.2	2.3	rain
3	2012-01-04	20.3	12.2	5.6	4.7	rain
4	2012-01-05	1.3	8.9	2.8	6.1	rain
...
1456	2015-12-27	8.6	4.4	1.7	2.9	rain
1457	2015-12-28	1.5	5.0	1.7	1.3	rain
1458	2015-12-29	0.0	7.2	0.6	2.6	fog
1459	2015-12-30	0.0	5.6	-1.0	3.4	sun
1460	2015-12-31	0.0	5.6	-2.1	3.5	sun

1461 rows x 6 columns

Gambar 4. Dataset yang Digunakan

```
en = LabelEncoder()
df['weather'] = en.fit_transform(df['weather'])
df
```

date	precipitation	temp_max	temp_min	wind	weather
2012-01-01	0.0	12.8	5.0	4.7	0
2012-01-02	10.9	10.6	2.8	4.5	2
2012-01-03	0.8	11.7	7.2	2.3	2
2012-01-04	20.3	12.2	5.6	4.7	2
2012-01-05	1.3	8.9	2.8	6.1	2
...
2015-12-27	8.6	4.4	1.7	2.9	2
2015-12-28	1.5	5.0	1.7	1.3	2
2015-12-29	0.0	7.2	0.6	2.6	1
2015-12-30	0.0	5.6	-1.0	3.4	4
2015-12-31	0.0	5.6	-2.1	3.5	4

1461 rows x 5 columns

Gambar 4.1. Merubah Tipe Data Weather dari Object ke Numeric (proses pembersihan data)

```
[1001] x
```

date	precipitation	temp_max	temp_min	wind
2012-01-01	0.0	12.8	5.0	4.7
2012-01-02	10.9	10.6	2.8	4.5
2012-01-03	0.8	11.7	7.2	2.3
2012-01-04	20.3	12.2	5.6	4.7
2012-01-05	1.3	8.9	2.8	6.1
...
2015-12-27	8.6	4.4	1.7	2.9
2015-12-28	1.5	5.0	1.7	1.3
2015-12-29	0.0	7.2	0.6	2.6
2015-12-30	0.0	5.6	-1.0	3.4
2015-12-31	0.0	5.6	-2.1	3.5

1461 rows x 4 columns

Gambar 4.2. variabel x untuk menampilkan 4 fitur (precipitation, temp_max, temp_min, dan wind) karena date digunakan pada visualisasi data, maka fitur date masih tampil pada output program (dataset sudah bersih)

```
[1702] y
```

date	
2012-01-01	0
2012-01-02	2
2012-01-03	2
2012-01-04	2
2012-01-05	2
..	
2015-12-27	2
2015-12-28	2
2015-12-29	1
2015-12-30	4
2015-12-31	4

Name: weather, Length: 1461, dtype: int64

Gambar 4.3. Variabel y untuk menampilkan fitur weather yang sudah berubah tipe data menjadi numeric, karena date digunakan pada visualisasi data, maka fitur date masih tampil pada output program (dataset sudah bersih)

Dalam implementasi ini, kami menggunakan dua pendekatan untuk mengukur performa algoritma Naive Bayes. Pertama, implementasi dilakukan menggunakan lingkungan pemrograman Python, khususnya dengan menggunakan Jupyter Notebook. Kami membagi data menjadi dua bagian, yaitu 60% sebagai data latih (training) dan 40% sebagai data uji (testing). Hasil prediksi dari model akan dibandingkan dengan label sebenarnya pada data uji untuk mengukur tingkat akurasi secara programatik.

```
▼ Membagi data latih dan data uji
```

```
[1707] random_state = 42
      X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.4, shuffle=False, random_state=random_state)
```

Gambar 4.4. Membagi Data Latih dan Data Uji

```
▼ Menampilkan shape data train dan test
```

```
[1708] for data_name, data in zip(["X_train", "X_test", "y_train", "y_test"], [X_train, X_test, y_train, y_test]):
      print(f"{data_name} shape:", data.shape)
```

X_train shape: (876, 4)
X_test shape: (585, 4)
y_train shape: (876,)
y_test shape: (585,)

Gambar 4.5. Menampilkan shape pada setiap train dan test

```

v Menampilkan training data X

[100]X_train
array([[ -0.48277769, -0.34200935, -0.48403759,  0.93669489],
       [ 1.20621014, -0.65187744, -0.93022175,  0.80432762],
       [-0.35881528, -0.49694339, -0.03785342, -0.65171238],
       ...,
       [ 0.10604376,  0.6721044 ,  1.0978881 ,  0.47340944],
       [-0.48277769,  0.43266088,  0.75310942, -0.58552875],
       [ 0.38495918, -0.03214126,  0.65170393, -1.24736511]])

```

Gambar 4.6. Menampilkan data *X_train*

```

v Menampilkan testing data X

[1711]X_test
array([[ -0.48277769,  0.43266088,  0.75310942,  0.80432762],
       [-0.48277769,  0.6721044 ,  0.53001734, -0.51934511],
       [-0.48277769,  0.51717036,  0.53001734,  0.07630762],
       ...,
       [-0.48277769, -1.13076448, -1.37640592, -0.45316147],
       [-0.48277769, -1.35612309, -1.7009035 ,  0.07630762],
       [-0.48277769, -1.35612309, -1.92399558,  0.14249125]])

```

Gambar 4.7. Menampilkan data *X_test*


```
Menampilkan training data y

y_train
array([0, 2, 2, 2, 2, 2, 2, 4, 2, 2, 4, 4, 4, 3, 3, 3, 3, 3, 3, 2, 2,
2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 2, 2,
2, 0, 2, 2, 2, 4, 2, 2, 2, 4, 2, 2, 3, 4, 3, 3, 4, 2, 4, 2, 2, 3,
4, 4, 2, 2, 2, 3, 3, 2, 3, 2, 3, 2, 2, 2, 2, 2, 4, 4, 2, 0, 2, 2,
2, 2, 2, 2, 4, 2, 4, 3, 2, 4, 4, 4, 2, 2, 2, 0, 4, 2, 2, 2, 2, 2,
2, 4, 2, 4, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4, 2, 4, 4,
4, 4, 4, 0, 4, 2, 2, 4, 2, 2, 2, 2, 2, 4, 4, 2, 4, 2, 2, 2,
4, 2, 2, 4, 2, 2, 2, 4, 2, 2, 4, 4, 4, 2, 4, 2, 2, 4, 2, 2, 0,
2, 2, 4, 2, 2, 2, 2, 2, 4, 0, 4, 4, 2, 2, 0, 1, 0, 2, 2, 2, 2,
4, 4, 4, 2, 4, 2, 2, 4, 4, 0, 0, 0, 4, 4, 4, 0, 4, 4, 4, 2, 0,
4, 0, 4, 4, 4, 4, 4, 4, 4, 4, 0, 0, 4, 2, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 4, 4, 4, 4, 4, 1, 4, 0, 0,
0, 2, 1, 1, 4, 0, 0, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0, 0, 0, 2,
2, 2, 2, 4, 4, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 0, 2, 2, 2, 2, 2, 2, 2, 0,
1, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2, 3, 3, 2,
3, 3, 2, 2, 2, 2, 2, 3, 2, 2, 2, 2, 0, 0, 4, 4, 2, 2, 2, 2, 2,
2, 3, 0, 4, 4, 4, 0, 0, 0, 0, 0, 0, 2, 2, 2, 2, 2, 2, 2, 2,
2, 2, 0, 2, 2, 2, 2, 4, 2, 2, 2, 2, 2, 2, 0, 2, 2, 4, 2, 2,
2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 2, 2, 2, 0, 1, 2, 2, 2, 2, 2,
2, 4, 2, 2, 2, 3, 4, 4, 4, 4, 4, 2, 2, 2, 0, 4, 4, 4, 4, 2, 2,
2, 2, 4, 2, 2, 2, 2, 1, 2, 0, 2, 2, 4, 2, 4, 4, 4, 1, 4, 2,
2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 4, 2, 1, 2, 4, 4,
2, 2, 2, 2, 4, 2, 2, 2, 2, 4, 4, 4, 2, 4, 4, 4, 4, 4, 4, 4,
2, 4, 4, 4, 4, 4, 2, 4, 2, 2, 4, 2, 2, 2, 2, 2, 4, 4, 4, 4,
1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 1, 4, 4, 4,
4, 4, 1, 4, 4, 4, 4, 2, 1, 4, 4, 4, 4, 4, 4, 2, 4, 4, 4, 2, 1,
4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 2, 2, 2, 4, 4, 4, 2, 2, 2, 4,
1, 4, 4, 4, 4, 1, 2, 2, 4, 4, 4, 2, 4, 2, 2, 4, 2, 4, 2, 2,
2, 2, 2, 4, 4, 2, 2, 2, 4, 2, 2, 2, 2, 1, 4, 4, 1, 1, 4, 4, 4,
4, 4, 4, 4, 2, 4, 4, 2, 2, 2, 2, 2, 0, 2, 2, 2, 4, 2, 4, 1, 2,
4, 2, 2, 4, 2, 2, 4, 4, 4, 4, 1, 4, 4, 4, 2, 2, 2, 2, 4, 4,
4, 4, 4, 4, 4, 4, 2, 2, 4, 2, 2, 4, 3, 2, 2, 2, 4, 4, 4,
2, 4, 4, 2, 2, 4, 2, 1, 4, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 4,
4, 4, 4, 2, 1, 4, 4, 4, 4, 2, 2, 4, 2, 2, 4, 4, 4, 4, 4, 3,
2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 4, 4, 2, 2,
2, 2, 2, 4, 2, 2, 2, 1, 1, 2, 2, 2, 2, 2, 4, 2, 4, 4, 4, 4,
2, 2, 2, 2, 4, 4, 4, 4, 2, 4, 2, 4, 4, 4, 4, 4, 4, 4, 2,
2, 2, 4, 2, 4, 2, 2, 2, 4, 2, 2, 4, 4, 4, 4, 4, 2, 2, 4, 4,
2, 2, 2, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 4, 2])
```

Gambar 4.8. Menampilkan data y_train

```
▼ Menampilkan testing data y

In [1703]: y_test

Out[1703]: array([4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 4, 2, 2,
2, 4, 2, 2, 4, 4, 4, 4, 4, 4, 2, 2, 4, 4, 4, 4, 4, 4, 1, 4, 1, 4,
4, 1, 4, 4, 4, 4, 4, 4, 4, 4, 1, 4, 4, 2, 2, 4, 4, 4, 4, 4, 4, 4,
4, 4, 2, 4, 4, 4, 1, 1, 4, 4, 4, 2, 2, 2, 4, 2, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 4, 2, 4, 1, 1, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 2, 2, 4, 1, 4, 2, 2, 2, 2, 2, 1, 1, 2, 4, 4, 4, 4, 4,
1, 1, 1, 1, 1, 2, 2, 4, 2, 2, 2, 4, 2, 2, 4, 2, 2, 2, 2, 2, 2, 2,
2, 2, 2, 2, 2, 1, 2, 2, 2, 2, 2, 4, 1, 2, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 2, 2, 2, 2, 2, 2, 2, 2, 2, 3, 4, 4, 1, 4, 2, 2, 2, 4, 2, 2,
2, 2, 4, 1, 1, 4, 4, 2, 2, 2, 2, 4, 4, 2, 2, 1, 1, 2, 2, 1, 4, 4,
4, 2, 1, 2, 2, 1, 1, 1, 2, 2, 2, 1, 1, 1, 2, 1, 2, 2, 2, 1, 1, 2,
2, 2, 1, 1, 2, 1, 4, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 2, 1,
2, 4, 1, 4, 4, 2, 2, 4, 4, 4, 4, 2, 2, 2, 4, 4, 4, 4, 4, 4, 1,
1, 1, 2, 2, 4, 2, 2, 2, 2, 2, 4, 4, 2, 2, 2, 2, 2, 2, 4, 2, 4, 4,
2, 2, 2, 4, 2, 4, 4, 2, 2, 4, 4, 2, 4, 4, 2, 2, 4, 4, 4, 4, 4,
2, 4, 2, 2, 2, 1, 2, 2, 4, 4, 4, 4, 4, 2, 1, 4, 4, 4, 4, 1, 2,
2, 2, 1, 4, 4, 4, 4, 1, 4, 4, 4, 4, 4, 4, 4, 4, 4, 2, 2, 4,
4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 4, 0, 4, 4, 4, 2, 4, 4, 4, 4, 4,
4, 4, 2, 4, 1, 4, 4, 4, 4, 4, 0, 4, 0, 4, 4, 4, 4, 4, 4, 4, 4,
4, 4, 4, 4, 4, 4, 2, 1, 2, 1, 4, 4, 4, 4, 4, 4, 1, 4, 4, 4, 1,
4, 4, 4, 2, 4, 2, 4, 4, 4, 4, 0, 2, 4, 0, 0, 4, 4, 4, 4, 2, 2, 2,
4, 2, 4, 4, 4, 2, 2, 2, 4, 4, 1, 4, 4, 2, 4, 4, 2, 2, 4, 4, 2, 1,
4, 4, 1, 2, 4, 4, 4, 4, 1, 1, 1, 4, 4, 4, 0, 2, 1, 2, 2, 4, 2, 2,
1, 1, 4, 2, 2, 2, 1, 1, 1, 1, 1, 2, 2, 1, 2, 2, 2, 2, 2, 2, 4,
2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 4, 1, 2, 2, 4, 4,
4, 4, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 4, 2, 2, 2, 2,
1, 2, 2, 2, 2, 2, 2, 4, 2, 2, 1, 4, 4])
```

Gambar 4.9. Menampilkan data y_test

5. Evaluasi Data

5.1. Validasi Model

Pada tahap evaluasi, kami melakukan validasi terhadap model yang telah dihasilkan. Validasi dilakukan dengan dua pendekatan, pertama menggunakan perhitungan akurasi secara programatik dengan Python di Jupyter Notebook. Perhitungan akurasi ini melibatkan perbandingan hasil prediksi dengan label sebenarnya pada data uji.

```
▼ sc digunakan untuk penskalaan fitur

X_train = sc.fit_transform digunakan untuk mean dan standar deviasi pada data train
X_test = sc.fit_transform digunakan untuk mean dan standar deviasi pada data test

In [1709]: sc = StandardScaler()
Out[1709]: X_train = sc.fit_transform(X_train)
          X_test = sc.transform(X_test)
```

Gambar 5.1. Menampilkan perhitungan mean dan standar deviasi

```
[1714]classifier = GaussianNB()
      classifier.fit(X_train, y_train)
```

▼ GaussianNB
 GaussianNB()

Gambar 5.2. Menampilkan Model Klasifikasi Gaussian Naïve Bayes

▼ Prediksi kelas menggunakan model lasifikasi

```
[1715]y_pred = classifier.predict(X_test)
      print(y_pred)
```

```
[4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 4 2 2 2 4 2 2 4 4 4 4 4 2 2 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 2 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4
 4 4 4 2 2 2 4 2 4 4 4 4 4 4 4 4 4 4 2 2 4 2 4 4 4 4 4 4 4 4 4 4 4
 4 4 4 2 2 4 4 4 2 2 2 2 2 4 4 2 4 4 4 4 4 4 4 4 4 2 2 4 2 2 2 4 2
 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 4 2 4 4 4 4 4 4 4 4 4 2 2 2 2 2 2
 2 2 3 4 0 4 4 2 2 2 4 2 2 2 2 4 4 4 4 4 2 2 2 4 2 2 4 4 2 2 4 4 0 3
 4 2 2 4 4 4 2 2 2 4 4 2 4 2 2 2 4 4 2 2 2 4 2 4 4 4 4 4 2 2 2 2 2
 2 2 4 2 4 2 4 4 4 2 2 4 4 4 2 2 2 4 4 4 4 4 4 4 4 4 2 2 4 2 2 2 2
 4 4 2 2 2 2 2 2 4 2 4 4 2 2 2 4 2 4 4 2 2 4 4 2 2 4 4 4 4 4 2 4 2
 2 2 4 2 2 4 4 4 4 4 2 4 4 4 4 4 2 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4
 4 2 2 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 4 4 4 4 4 4 4 2 4 4 4 4 4
 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 2 4 2 4 4 4 4 4 4 4 4 4 4 4 4 4 2
 4 2 4 4 4 4 4 2 4 4 4 4 4 4 4 2 2 2 4 2 4 4 4 2 2 2 4 4 4 2 4 4 2 2 4
 4 2 4 4 4 4 2 4 4 4 4 4 4 4 4 4 4 2 2 4 2 2 4 2 2 4 4 4 2 2 4 4 4 2
 2 4 2 2 2 2 2 2 2 4 2 4 2 2 2 2 2 2 2 2 2 2 2 2 4 4 4 2 2 4 4 4 0 3 2
 2 2 2 2 2 2 2 2 2 2 2 2 2 4 2 2 2 2 2 2 2 2 2 2 4 3 2 4 4 4]
```

Gambar 5.3. Menampilkan Hasil y_pred

▼ Penghitungan matriks confusion

```
[1716]cm = confusion_matrix(y_test, y_pred)
      cm
```

```
array([[ 0,  0,  0,  0,  7],
       [ 1,  0,  0,  0, 75],
       [ 0,  0, 217,  3,  0],
       [ 0,  0,  0,  1,  0],
       [ 2,  0,  0,  0, 279]])
```

Gambar 5.4. Menampilkan Perhitungan Matrix Confusion

✓ Laporan klasifikasi

```
[1717]akurasi = classification_report(y_test, y_pred)
      print(akurasi)
```

	precision	recall	f1-score	support
0	0.00	0.00	0.00	7
1	0.00	0.00	0.00	76
2	1.00	0.99	0.99	220
3	0.25	1.00	0.40	1
4	0.77	0.99	0.87	281
accuracy			0.85	585
macro avg	0.40	0.60	0.45	585
weighted avg	0.75	0.85	0.79	585

Gambar 5.5. Menampilkan Classification Report

✓ Tingkat akurasi

```
[1718]akurasi = accuracy_score(y_test, y_pred)
      print("Tingkat akurasi : %d persen"%(akurasi*100))
```

Tingkat akurasi : 84 persen

Gambar 5.6. Menampilkan Tingkat Akurasi

Menampilkan DataFrame dari data hasil prediksi

```
[1719] ydata = pd.DataFrame()
ydata['y_test'] = pd.DataFrame(y_test)
ydata['y_pred'] = pd.DataFrame(y_pred)
ydata
```

	y_test	y_pred
0	4	4
1	4	4
2	4	4
3	4	4
4	4	4
...
580	2	3
581	2	2
582	1	4
583	4	4
584	4	4

585 rows x 2 columns

Gambar 5.7. Menampilkan DataFrame Hasil *y_test* dan *y_pred*

Selain validasi secara programatik, kami juga melakukan validasi manual menggunakan metode tulisan tangan. Sebanyak 40 sampel data uji dipilih, di mana 60% dari sampel tersebut digunakan untuk data latih dan 40% sebagai data uji. Hasil prediksi dari model Naive Bayes akan dibandingkan dengan label sebenarnya, dan akurasi akan dihitung secara manual.

Berikut adalah validasi manual:

Precipitation	Temp_max	Temp_min	wind	weather	
0.5	13.9	5.6	2.6	Rain	Pengandaian
10.2	10.6	3.3	4.5	Rain	
11.2	12.8	7.2	5.9	Rain	Precipitation
16.3	11.7	5.6	6.3	Rain	6-4 sangat rendah
21.3	21.7	16.1	2.6	Rain	5-10 Rendah
25.4	15.6	11.1	3.2	Rain	11-15 sedang
32.3	12.8	6.7	2.7	Rain	16-20 agak tinggi
0.5	16.1	11.7	6.3	Rain	21-25 tinggi
1.5	5.0	3.3	1.7	Rain	26-30 sangat tinggi
0.5	3.9	0.0	2.4	Rain	31-35 kelabu tinggi
2.0	8.3	1.7	9.5	Rain	
0.0	0.0	-7.1	3.1	Sun	temp_max dan temp_min
0.0	1.1	-4.3	4.7	Sun	≤ -4 sangat dingin
0.0	6.7	3.3	2.0	Sun	-3-0 dingin
0.0	13.3	7.2	4.01	Sun	1-5 normal
0.0	16.7	11.1	2.9	Sun	6-10 agak panas
0.0	22.2	13.9	2.6	Sun	11-15 panas
0.0	6.7	0.0	1.6	Sun	16-20 keik
0.0	33.3	17.2	3.9	Sun	21-25 sangat keik
0.0	1.1	-0.6	1.9	Drizzle	
0.0	7.2	0.6	1.3	Drizzle	wind
0.0	13.9	7.2	1.3	Drizzle	0-2 normal
0.0	20.0	5.6	4.4	Drizzle	3-5 berangin
0.0	22.2	13.3	1.7	Drizzle	6-8 leceang
0.0	30.0	16.1	3.3	Drizzle	7-9 badai
0.0	1.7	-2.1	0.9	Fog	
0.0	7.2	3.3	1.9	Fog	
0.0	14.4	8.9	1.7	Fog	
0.0	17.8	10.6	1.8	Fog	
0.0	22.2	11.1	2.5	Fog	
0.0	27.2	17.8	4.1	Fog	
0.0	27.8	13.3	6.5	Fog	
1.3	5.0	-1.1	3.4	Snow	
5.6	8.3	0.6	3.7	Snow	
16.5	5.6	2.3	4.2	Snow	
10.8	0.0	-2.8	5.0	Snow	
27.6	6.7	3.3	5.5	Snow	
5.6	4.4	-4.3	5.3	Snow	
5.2	-8.1	-2.8	1.6	Snow	
0.8	5.0	1.1	7.0	Snow	

Gambar 4. Tabel data yang digunakan

Data Training

$P(C_i)$

$$P(\text{Rain}) = \frac{1}{40} = 0,025$$

$$P(\text{Sun}) = \frac{8}{40} = 0,200$$

$$P(\text{Drizzle}) = \frac{6}{40} = 0,150$$

$$P(\text{Fog}) = \frac{7}{40} = 0,175$$

$$P(\text{Snow}) = \frac{8}{40} = 0,200$$

Data latih 1 = 1009

$P(X C_i)$	Rain	Sun	Drizzle	Fog	Snow
Precipitation	$\frac{5}{11}$	$\frac{8}{8}$	$\frac{6}{6}$	$\frac{7}{7}$	$\frac{8}{8}$
max	$\frac{1}{11}$	$\frac{8}{8}$	$\frac{2}{6}$	$\frac{3}{7}$	$\frac{0}{8}$
min	$\frac{3}{11}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{2}{7}$	$\frac{0}{8}$
wind	$\frac{5}{11}$	$\frac{3}{8}$	$\frac{4}{6}$	$\frac{5}{7}$	$\frac{1}{8}$
$P(X C_i)$	0,0054	0,0117	0,0370	0,0075	0
$P(C_i) \times P(X C_i)$	0,0009	0,0023	0,0056	0,0153	0
tertinggi	0,0153 (salah)				

Data latih 2 = 1010

	Rain	Sun	Drizzle	Fog	Snow
Precipitation	$\frac{5}{11}$	$\frac{8}{8}$	$\frac{6}{6}$	$\frac{7}{7}$	$\frac{8}{8}$
max	$\frac{1}{11}$	$\frac{8}{8}$	$\frac{2}{6}$	$\frac{3}{7}$	$\frac{0}{8}$
min	$\frac{3}{11}$	$\frac{2}{8}$	$\frac{1}{6}$	$\frac{2}{7}$	$\frac{0}{8}$
wind	$\frac{5}{11}$	$\frac{3}{8}$	$\frac{4}{6}$	$\frac{5}{7}$	$\frac{1}{8}$
$P(X C_i)$	0,0054	0,0047	0,0370	0,0075	0
$P(C_i) \times P(X C_i)$	0,0009	0,0047	0,0056	0,0153	0
tertinggi	0,0153 (salah)				

Data latih 3 = 1011

	Rain	Sun	Drizzle	Fog	Snow
Precipitation	$\frac{5}{11}$	$\frac{8}{8}$	$\frac{6}{6}$	$\frac{7}{7}$	$\frac{8}{8}$
max	$\frac{1}{11}$	$\frac{8}{8}$	$\frac{2}{6}$	$\frac{3}{7}$	$\frac{0}{8}$
min	$\frac{3}{11}$	$\frac{2}{8}$	$\frac{1}{6}$	$\frac{2}{7}$	$\frac{0}{8}$
wind	$\frac{5}{11}$	$\frac{3}{8}$	$\frac{4}{6}$	$\frac{5}{7}$	$\frac{1}{8}$
$P(X C_i)$	0,0054	0,0117	0,0370	0,0075	0
$P(C_i) \times P(X C_i)$	0,0009	0,0047	0,0056	0,0153	0
tertinggi	0,0153 (Benar)				

Data latih 4 = 1012

	Rain	Sun	Drizzle	Fog	Snow
Precipitation	$\frac{5}{11}$	$\frac{8}{8}$	$\frac{6}{6}$	$\frac{7}{7}$	$\frac{8}{8}$
max	$\frac{1}{11}$	$\frac{2}{8}$	$\frac{2}{6}$	$\frac{3}{7}$	$\frac{0}{8}$
min	$\frac{3}{11}$	$\frac{2}{8}$	$\frac{1}{6}$	$\frac{2}{7}$	$\frac{0}{8}$
wind	$\frac{5}{11}$	$\frac{3}{8}$	$\frac{4}{6}$	$\frac{5}{7}$	$\frac{1}{8}$
$P(X C_i)$	0,0054	0,0117	0,0370	0,0075	0
$P(C_i) \times P(X C_i)$	0,0009	0,0023	0,0056	0,0153	0
tertinggi	0,0153 (Benar)				

Data latih 5 = 1013

	Rain	Sun	Drizzle	Fog	Snow
Precipitation	$\frac{5}{11}$	$\frac{8}{8}$	$\frac{6}{6}$	$\frac{7}{7}$	$\frac{8}{8}$
max	$\frac{1}{11}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{0}{8}$
min	$\frac{3}{11}$	$\frac{2}{8}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{0}{8}$
wind	$\frac{5}{11}$	$\frac{2}{8}$	$\frac{4}{6}$	$\frac{5}{7}$	$\frac{1}{8}$
$P(X C_i)$	0,0054	0,0117	0,0370	0,0075	0
$P(C_i) \times P(X C_i)$	0,0009	0,0023	0,0056	0,0051	0
tertinggi	0,0051 (Benar)				

Data latih 6 = 1014

	Rain	Sun	Drizzle	Fog	Snow
Precipitation	$\frac{5}{11}$	$\frac{8}{8}$	$\frac{6}{6}$	$\frac{7}{7}$	$\frac{8}{8}$
max	$\frac{1}{11}$	$\frac{1}{8}$	$\frac{1}{6}$	$\frac{1}{7}$	$\frac{0}{8}$
min	$\frac{3}{11}$	$\frac{2}{8}$	$\frac{1}{6}$	$\frac{2}{7}$	$\frac{0}{8}$
wind	$\frac{5}{11}$	$\frac{3}{8}$	$\frac{4}{6}$	$\frac{5}{7}$	$\frac{1}{8}$
$P(X C_i)$	0,0054	0,0117	0,0370	0,0075	0
$P(C_i) \times P(X C_i)$	0,0009	0,0023	0,0056	0,0051	0
tertinggi	0,0051 (Benar)				

$$\text{Akurasi} = \left(\frac{4}{6}\right) \times 100\%$$

$$\approx 66,67\%$$

Gambar 5. Perhitungan Data Latih

Data Uji:

Data Uji 1 = 1369

	Rain	Sun	Drizzle	Fog	Snow
Precipitation =	5/6	8/8	6/6	7/7	3/6
max =	1/11	1/8	1/6	1/7	0/3
min =	7/11	1/8	1/6	2/7	0/8
wind =	5/11	3/8	4/6	5/7	8/8
$P(x C_i)$	0,0034	0,0038	0,0037	0,0038	0
$P(C_i) \times P(x C_i)$	0,00063	0,0012	0,0028	0,0038	0
tertinggi =	0,00451 (Benar)				

Data Uji 2 = 1370

	Rain	Sun	Drizzle	Fog	Snow
Precipitation =	5/11	0/8	6/6	7/7	3/6
max =	1/11	7/8	7/6	3/7	0/8
min =	7/11	1/8	1/6	2/7	0/8
wind =	5/11	3/8	4/6	5/7	1/8
$P(x C_i)$	0,0034	0,0117	0,0037	0,0038	0
$P(C_i) \times P(x C_i)$	0,0009	0,0023	0,0037	0,0038	0
tertinggi =	0,0053 (Benar)				

Data Uji 3 = 1371

	Rain	Sun	Drizzle	Fog	Snow
Precipitation =	5/11	4/8	6/6	7/7	3/6
max =	5/11	1/8	1/6	1/7	0/8
min =	7/11	1/8	1/6	2/7	0/8
wind =	5/11	3/8	4/6	5/7	1/8
$P(x C_i)$	0,0034	0,0038	0,0037	0,0038	0
$P(C_i) \times P(x C_i)$	0,00047	0,0012	0,0028	0,0038	0
tertinggi =	0,0051 (Benar)				

Data Uji 4 = 1372

	Rain	Sun	Drizzle	Fog	Snow
Precipitation =	5/11	8/8	6/6	7/7	3/6
max =	1/11	1/8	1/6	1/7	0/3
min =	7/11	7/8	1/6	2/7	0/8
wind =	5/11	4/8	7/6	1/7	6/8
$P(x C_i)$	0,0034	0,0117	0,0037	0,0038	0
$P(C_i) \times P(x C_i)$	0,0006	0,0031	0,0037	0,0038	0
tertinggi =	0,0051 (Benar)				

Data Uji 5 = 1373

	Rain	Sun	Drizzle	Fog	Snow
Precipitation =	5/11	8/8	6/6	7/7	3/6
max =	1/11	7/8	7/6	3/7	0/8
min =	7/11	1/8	1/6	2/7	0/8
wind =	5/11	4/8	7/6	1/7	1/8
$P(x C_i)$	0,0034	0,0117	0,0037	0,0038	0
$P(C_i) \times P(x C_i)$	0,0006	0,0031	0,0037	0,0038	0
tertinggi =	0,00313 (Benar)				

Data Uji 6 = 1374

	Rain	Sun	Drizzle	Fog	Snow
Precipitation =	5/11	0/8	6/6	7/7	3/6
max =	1/11	7/8	7/6	3/7	0/8
min =	7/11	1/8	1/6	2/7	0/8
wind =	5/11	3/8	4/6	5/7	1/8
$P(x C_i)$	0,0034	0,0117	0,0037	0,0038	0
$P(C_i) \times P(x C_i)$	0,0005	0,0023	0,0037	0,0038	0
tertinggi =	0,0153 (Salah)				

$$\text{akurasi} = (5/6) \times 100\% \\ \approx 83,33\%$$

Gambar 6. Perhitungan Data Uji

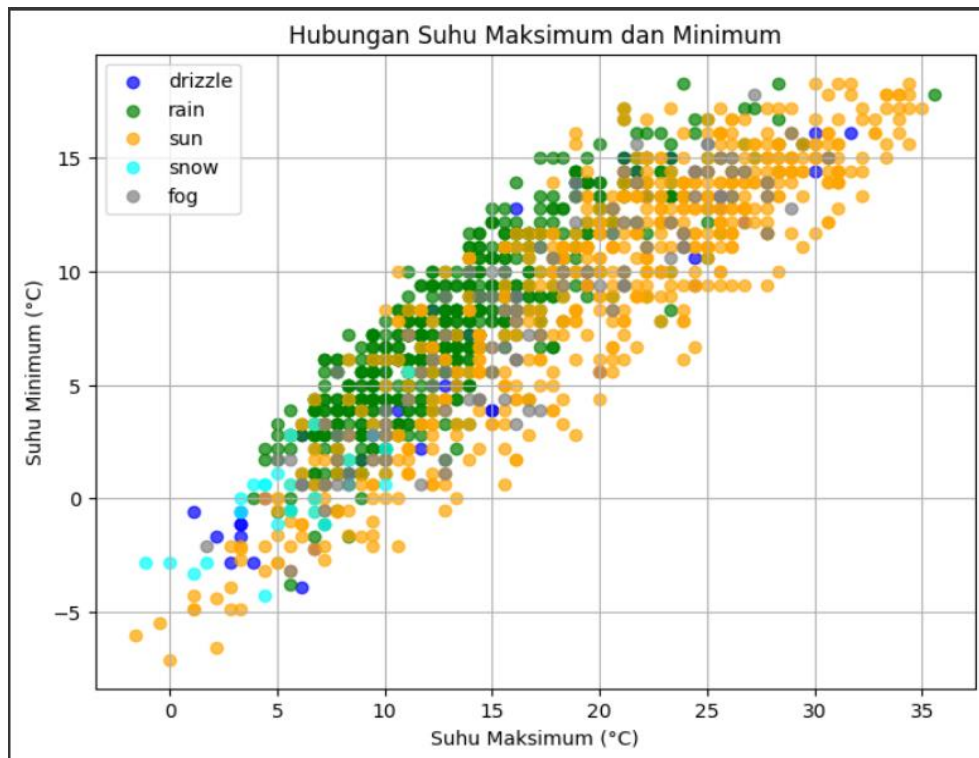
Hasil perhitungan akurasi pada data training mencapai 66,67%, sedangkan pada data testing mencapai 83,33%. Ini menunjukkan bahwa model Naive Bayes mampu memberikan prediksi cuaca dengan tingkat keakuratan yang baik.

Dengan melakukan dua jenis validasi ini, diharapkan dapat memberikan kepercayaan yang lebih tinggi terhadap keakuratan dan konsistensi model prediksi cuaca yang telah dikembangkan.

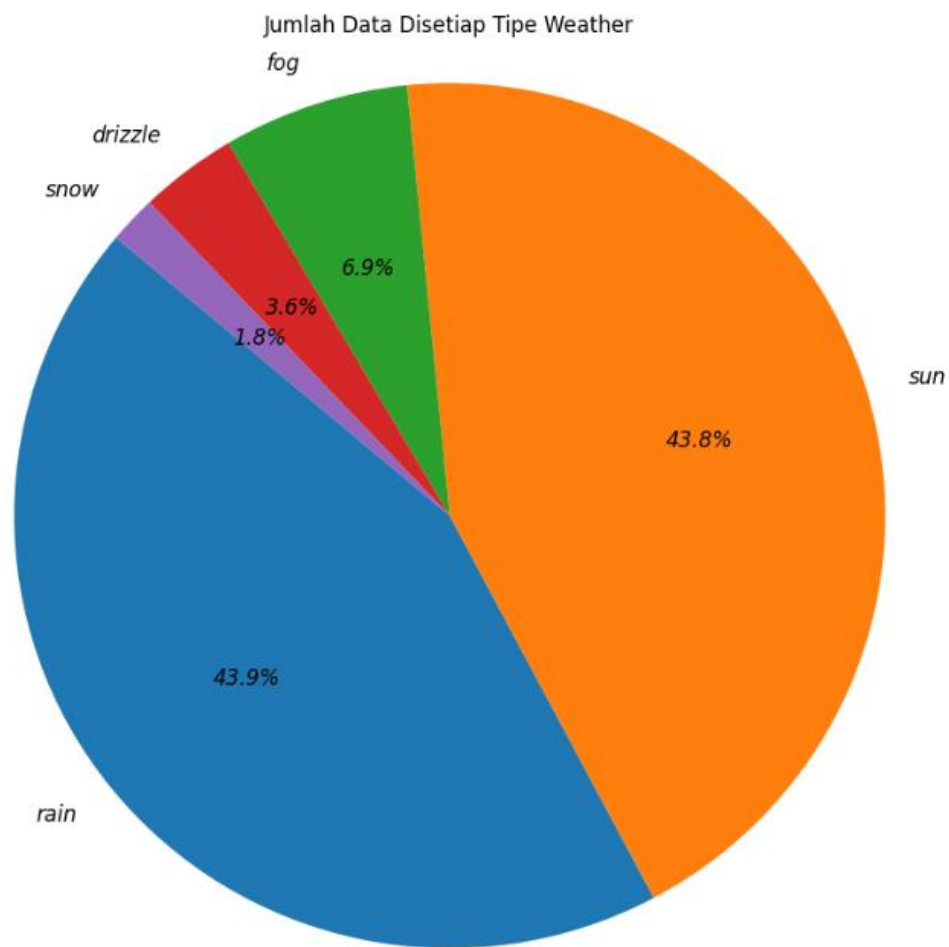
6. Penyebaran Data

6.1. Produksi Laporan

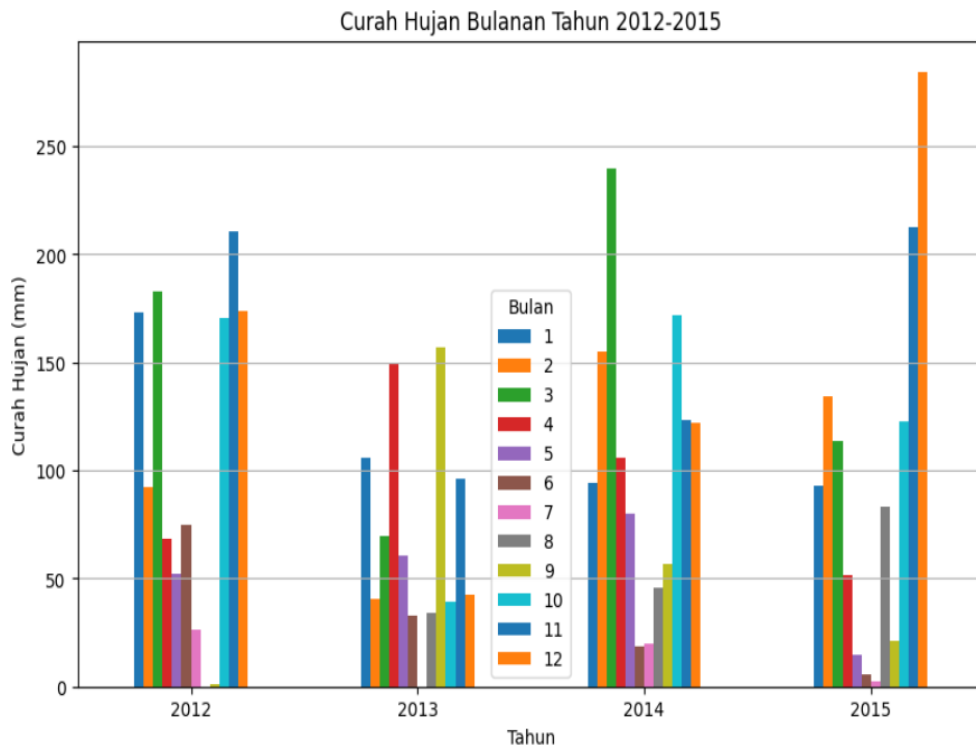
Setelah berhasil mengimplementasikan algoritma Naive Bayes dan mengevaluasi model prediksi cuaca, langkah selanjutnya adalah menyusun laporan akhir. Laporan ini mencakup temuan-temuan signifikan yang dihasilkan dari analisis data cuaca Seattle tahun 2012-2015 menggunakan metode Naive Bayes. Laporan akan memaparkan secara terinci hasil prediksi cuaca, tingkat akurasi, dan pola-pola menarik yang teridentifikasi selama proses analisis.



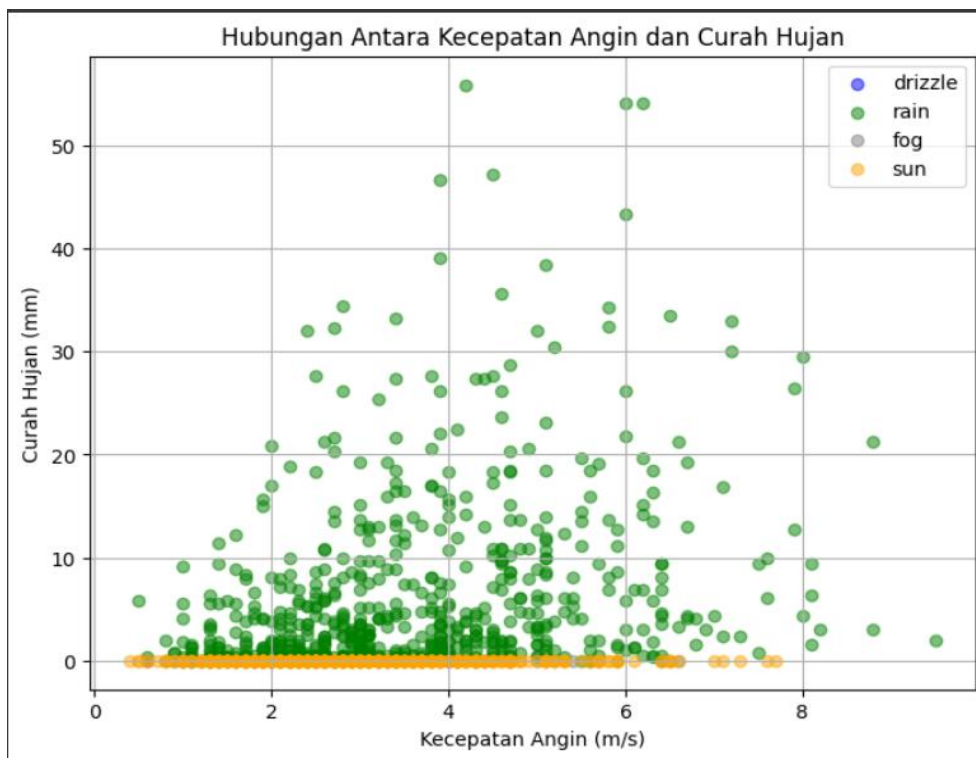
Gambar 6.1. Menampilkan Antara Hubungan Suhu Maksimum dan Suhu Minimum



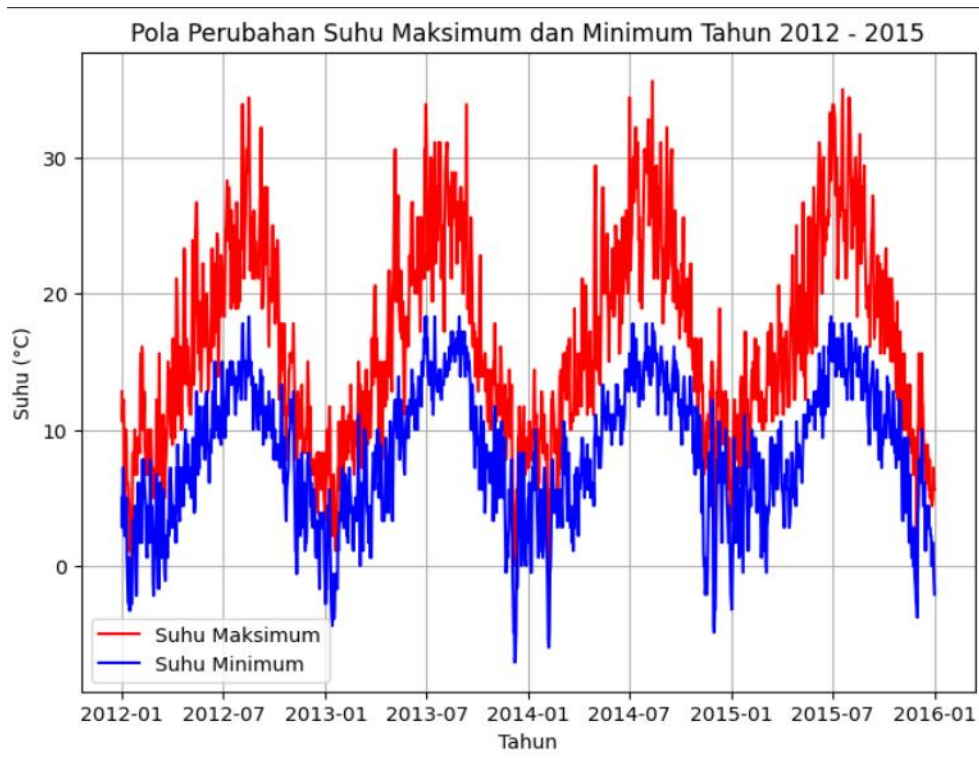
Gambar 6.2. Menampilkan Jumlah Data Pada Setiap Tipe Weather



Gambar 6.3. Menampilkan Curah Hujan Setiap Bulan dari Tahun 2012 – 2015



Gambar 6.4. Menampilkan Hubungan Antara Kecepatan Angin dan Curah Hujan



Gambar 6.5. Menampilkan Perbandingan Antara Suhu Maksimum dan Suhu Minimum dari Tahun 2012 - 2015

7. Kesimpulan

Melalui penerapan algoritma Naive Bayes dalam meramalkan kondisi cuaca di Seattle berdasarkan data cuaca tahun 2012-2015, dapat disimpulkan bahwa metode ini memberikan tingkat akurasi yang signifikan. Dengan menggunakan Metodologi CRISP-DM sebagai panduan, tahapan analisis data, pemodelan dengan Naive Bayes, hingga evaluasi performa model berhasil dilaksanakan.

Temuan utama dari analisis ini mengindikasikan bahwa prediksi cuaca dapat diandalkan dengan memanfaatkan atribut-atribut tertentu. Hasil akhir laporan mencakup temuan dan rekomendasi praktis untuk mendukung pemangku kepentingan dalam pengambilan keputusan terkait aktivitas yang sensitif terhadap kondisi cuaca di Seattle. Dengan demikian, penerapan Naive Bayes dalam prediksi cuaca tidak hanya meningkatkan pemahaman terhadap pola cuaca, tetapi juga memberikan nilai tambah berupa panduan praktis.

Daftar Pustaka

G. I. Webb. "Encyclopedia of Machine Learning and Data Mining", Encyclopedia of Machine Learning and Data Mining

F. A. Hermawati, Data Mining, Yogyakarta: CV ANDI OFFSET, 2013