

Pengembangan Aplikasi Berbasis Web dengan Rshiny untuk Data Klasifikasi Menggunakan Metode *Naive Bayes*

Muhammad Daffa Muafa
Program Studi Sarjana Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
muhammad.muafa@students.uii.ac.id

Lizda Iswari
Program Studi Sarjana Informatika
Universitas Islam Indonesia
Yogyakarta, Indonesia
lizda.iswari@uii.ac.id

Abstrak—Di era serba teknologi, data membawa banyak sekali manfaat jika pemiliknya dapat menggunakannya dengan benar. Dalam setiap data pasti terdapat sebuah informasi yang berharga. Untuk mengolah data dibutuhkan pengetahuan yang cukup mengenai *machine learning*. Dalam pengolahan data, salah satu metode perhitungan data dalam *Machine Learning*, adalah *Naive Bayes*. *Naive Bayes* merupakan metode klasifikasi yang cukup populer, karena cukup sederhana, efisien, dan memiliki tingkat akurasi yang baik. Penelitian ini mengembangkan sebuah aplikasi berbasis web *R Shiny*, yang dapat membantu pengguna dalam menggali informasi dari *dataset* yang mereka miliki, khususnya data klasifikasi menggunakan metode *Naive Bayes*. Aplikasi data klasifikasi menghasilkan *output* berupa informasi hasil klasifikasi dari *dataset* yang digunakan, serta menghasilkan data baru berupa data prediksi yang dapat diunduh. Aplikasi dirancang *user friendly*, sederhana, dan praktis karena aplikasi ini berbasis web. Pengguna tidak perlu mengunduh aplikasi untuk mengoperasikannya.

Keywords—*Data Science, Machine Learning, Data Classification, Naive Bayes, R, R Shiny*

I. PENDAHULUAN

Perkembangan teknologi informasi yang tiada henti memaksa manusia harus cepat beradaptasi jika tidak ingin ketinggalan zaman. Teknologi Informasi (TI) memberikan banyak sekali manfaat di berbagai hal, terutama di bidang ilmu pengetahuan. Manusia sangat mudah menemukan informasi, pengetahuan, dan wawasan hanya dalam hitungan menit bahkan detik. Menurut B. Uno dan Nina Lamatenggo, “Teknologi informasi adalah suatu teknologi yang digunakan untuk mengolah data.

Pengolahan itu termasuk memproses, mendapatkan, menyusun, menyimpan, memanipulasi data dalam berbagai cara untuk menghasilkan informasi yang berkualitas, yaitu informasi yang relevan, akurat, dan tepat waktu” [1]. TI mengandung ribuan bahkan jutaan data. Data tersebut diolah agar mendapatkan sebuah informasi yang diinginkan dan dapat dipahami oleh banyak orang. Data-data seperti inilah yang dikategorisasikan sebagai data-data ilmiah. Data ilmiah ini disebut sebagai *data science*

Data science merupakan sebuah ilmu komputasi yang digunakan untuk memperoleh informasi berharga dari sebuah data. *Data science* memerlukan keterampilan dalam ilmu komputer, matematika dan statistika, dan individu yang mempunyai pengetahuan mendalam terhadap suatu bidang. *Data Science* merupakan ilmu mencakup berbagai macam ilmu pengetahuan. Salah satu ilmu tersebut adalah *Machine Learning*

Machine learning termasuk salah satu ilmu *artificial intelligence* yang mempunyai tujuan untuk mengembangkan sebuah mesin atau komputer agar dapat mempelajari sesuatu dengan sendirinya tanpa arahan dari pengguna. Terdapat 2 teknik dasar *Machine Learning* yaitu *Supervised* dan *Unsupervised*. *Supervised* adalah sebuah teknik pembelajaran mesin yang bersumber dari informasi yang sudah tersedia. Teknik ini menghasilkan informasi yang nantinya dapat dimanfaatkan guna memprediksi suatu kejadian yang akan terjadi di waktu yang akan datang. Sedangkan, *Unsupervised* tidak mempunyai informasi yang dijadikan acuan sebelumnya. *Unsupervised* membantu menemukan pola yang tersembunyi pada data [2].

Teknik dasar *Machine Learning* yang

digunakan pada penelitian ini adalah *supervised learning*, dengan algoritma *Naive Bayes classifier*. *Naive Bayes classifier* menggunakan metode probabilitas statistik yang memprediksi probabilitas di masa yang akan datang, berdasarkan informasi di masa sebelumnya. *Naive Bayes* merupakan sebuah metode yang populer untuk klasifikasi teks karena sangat sederhana, praktis dan juga memiliki performa akurasi yang baik. *Naive Bayes* menggunakan perhitungan probabilitas. Konsep yang digunakan adalah *Teorema Bayes* yang dikemukakan oleh Thomas Bayes, salah seorang ilmuwan yang berasal dari Inggris. Teorema ini digunakan untuk menghitung peluang suatu kelas dari masing-masing kelompok data, untuk menentukan kelas mana yang paling ideal [3].

Visualisasi data akan ditampilkan pada *Shiny* yang merupakan salah satu paket dalam aplikasi *R* untuk membangun aplikasi berbasis web. *Shiny* dapat memberikan kemudahan kepada penggunanya dalam pembuatan aplikasi berbasis web, terutama untuk menampilkan visualisasi data dengan halaman website yang interaktif. *Server* pada *Shiny* menggunakan pemrograman yang reaktif sehingga memungkinkan penggunanya menggunakan logika, dan algoritma apapun dalam pengerjaannya [4]. Aplikasi yang mempunyai fungsi serupa adalah *WEKA* atau “Waikato Environment for Knowledge Analysis”.

WEKA adalah perangkat lunak berbasis *desktop*, yang dapat membantu penggunanya dalam mengolah informasi dari data yang mereka miliki. *WEKA* mewajibkan penggunanya untuk mengunduh aplikasinya melalui web. Diperlukan proses penginstalan aplikasi terlebih dahulu, agar pengguna dapat mengoperasikannya. Penelitian ini mengembangkan aplikasi berbasis web, yang memiliki fungsi serupa dengan *WEKA*. Fitur utama dari aplikasi yang dikembangkan adalah dapat memberikan output berupa informasi dari dataset pengguna, khususnya klasifikasi data menggunakan metode *Naive Bayes*. Aplikasi ini juga akan menghasilkan data baru berupa data prediksi yang dapat diunduh oleh pengguna.

II. TINJAUAN PUSTAKA

A. Naive Bayes Classifier

Naive Bayes adalah sebuah pengklasifikasi yang menggunakan metode statistik dan juga probabilitas, yaitu memprediksi probabilitas atau peluang di waktu yang akan datang berdasarkan data yang sudah ada. *Naive Bayes* akan menghitung probabilitas atau kemungkinan sebuah kelas, dari masing-masing kelompok yang terdapat pada data yang digunakan, dan menentukan kelas yang paling baik. Rumus dasar dari *Naive Bayes* yang ditemukan oleh Thomas Bayes adalah sebagai berikut:

$$P(X|Y) = \frac{P(Y|X) \times P(X)}{P(Y)}$$

- $P(X|Y)$ = Kemungkinan bersyarat. X terjadi jika Y benar
- $P(Y|X)$ = Kemungkinan bersyarat. Y terjadi jika X benar
- $P(X)$ = Kemungkinan X tanpa syarat
- $P(Y)$ = Kemungkinan Y tanpa syarat
- Kondisi X dan Y tidak boleh sama

Metode klasifikasi *Naive Bayes* akan melalui 2 tahap yaitu tahap *training* dan tahap *testing*. Karena *Naive Bayes* merupakan teknik *Supervised Learning*, maka dibutuhkan proses *training* sebagai tahap awal. Pada tahap *training*, *Naive Bayes* akan mempelajari data terlebih dahulu agar dapat membentuk model probabilitas. *Naive Bayes* hanya membutuhkan *data training* dengan jumlah yang sedikit dalam proses pembentukan model klasifikasi. Selanjutnya, model probabilitas yang telah dibentuk berdasarkan tahap *training*, akan diuji menggunakan sebuah data. Pengujian dilakukan untuk menguji metode yang digunakan, apakah dapat menghasilkan akurasi yang baik atau tidak. Tahap pengujian ini disebut tahap *testing*. Hasil dari data *testing* tersebut akan menghasilkan beberapa parameter sebagai berikut :

True Positive (TP) = Seberapa banyak data yang bernilai positif dan modelnya juga memprediksi positif.

True Negative (TN) = Seberapa banyak data yang bernilai negatif dan modelnya juga memprediksi negatif.

False Positive (FP) = Seberapa banyak data yang bernilai negatif dan modelnya juga memprediksi positif.

False Negative (FN) = Seberapa banyak data yang bernilai positif dan modelnya juga memprediksi negatif.

1. Accuracy

Accuracy adalah total keseluruhan dari seberapa sering model *Naive Bayes* bernilai benar. Rumus *Accuracy* adalah :

$$\frac{TP + TN}{Total}$$

2. Precision

Precision adalah seberapa sering sebuah prediksi bernilai benar jika model memprediksi positif. Rumus dari *precision* adalah :

$$\frac{TP}{FN + TP}$$

3. Sensitivity atau recall

Sensitivity merupakan sebuah kebenaran prediksi dari data seluruh data positif. Rumus dari *Sensitivity* adalah :

$$\frac{TP}{TP + FN}$$

4. Specificity

Specificity merupakan sebuah kebenaran prediksi dari data seluruh data negatif. Rumus dari *Specificity* adalah :

$$\frac{TN}{TN + FP}$$

5. F1-Score

F1-Score adalah perhitungan rata-rata dari *precision* dan juga *recall* atau *sensitivity*. Rumus dari *F1-Score* adalah :

$$2 \times \frac{Precision \times Sensitivity}{Precision + Sensitivity}$$

B. Data Classification

Data Classification atau klasifikasi data adalah sebuah metode pengelompokan berdasarkan karakteristik dari masing-masing objek. Klasifikasi mempunyai fungsi untuk mengklasifikasikan atau

mengelompokkan data menjadi kelas-kelas data yang menunjukkan isi dari data tersebut [5].

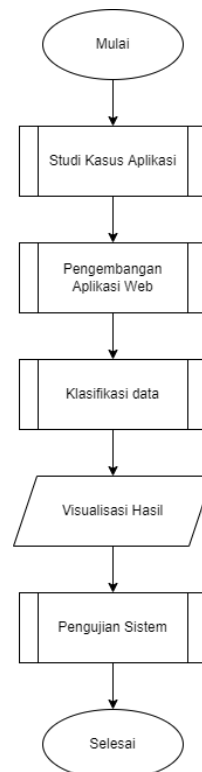
C. Bahasa Pemrograman R

R adalah bahasa pemrograman yang digunakan untuk komputasi, analisis data dan juga menampilkan grafik untuk statistika. *R* mempunyai banyak sekali *package* yang dapat dimanfaatkan sesuai dengan kebutuhan. *R* salah satu bahasa pemrograman yang interaktif sehingga pengguna dapat langsung melihat hasil perhitungannya [6].

D. R Shiny

R Shiny adalah sebuah *tool* dari *R* yang dapat memfasilitasi penggunaanya dalam pembuatan *user interface* berbasis web. *Shiny* menggunakan bahasa pemrograman *R* yang memungkinkan penggunaanya dapat membuat web yang interaktif. Hal ini diharapkan dapat mempermudah user dalam mengoperasikannya

III. METODE PERANCANGAN SISTEM



Gambar 1. Flowchart perancangan sistem

A. Studi Kasus Aplikasi

Perancangan sistem ini diawali dengan melakukan studi kasus terhadap aplikasi serupa yang sudah digunakan oleh banyak orang. Aplikasi berbasis *Machine learning* dan sains data yang dijadikan pembanding pada perancangan sistem ini adalah aplikasi “Waikato Environment for Knowledge Analysis” atau yang lebih sering didengar dengan “WEKA” [7].

WEKA adalah sebuah *tools machine learning* yang dapat memenuhi kebutuhan *data mining* dengan menggunakan berbagai macam algoritma *machine learning*. Penggunaanya dapat menggali informasi dari *dataset* yang mereka miliki, dan mereka akan diberikan fleksibilitas dalam memilih antara algoritma *supervised* atau *unsupervised learning*. Pengaplikasian *supervised learning* dengan metode *Naive Bayes* dilakukan untuk menjadi bahan komparasi.

Hasil perhitungan *Naive Bayes* menggunakan aplikasi WEKA memperoleh informasi berupa, penjabaran perhitungan data dalam bentuk tabel, *summary data* berupa *Correctly Classified Instance* atau data yang terklasifikasi benar, *Incorrectly Classified Instances* atau data yang terklasifikasi salah, *kappa statistik*, *detail Accuracy* yang berisi *precision*, *recall*, *f1-score* dan sebagainya.

B. Pengembangan Aplikasi

Pada tahap pengembangan aplikasi web, penelitian ini menggunakan salah satu paket dari R yaitu *R Shiny*. Pada proses pembuatannya, dibutuhkan 2 jenis script yaitu *UI* dan *server*. Berikut adalah penjelasan dari masing-masing script.

a. UI

UI atau *User Interface* adalah *script* yang digunakan sebagai tampilan utama dari aplikasi ini. Keindahan tampilan web ini bergantung pada *code UI*. *Script* ini menggunakan bahasa *R* dan juga *HTML* dalam pengkodeannya. Seperti pembuatan web pada umumnya, *script* ini memberikan kebebasan kepada penggunaanya untuk mengatur user interface sesuai dengan kebutuhan dan kreativitas mereka.

b. Server

Server berisi seluruh *code* pemodelan *Naive Bayes*. *Script* ini berperan untuk input *dataset*, *splitting data* menjadi *data training*, dan juga *data testing*. Bisa dikatakan *script* ini menjadi penentu

berhasil atau tidaknya perhitungan dan prediksi dari aplikasi ini. Jika *code* ini berhasil, maka hasil dari pemodelan tersebut akan ditampilkan oleh *UI*.

```
NB_Model <- reactive({  
  naiveBayes(f(), data = trainingData())  
})
```

Gambar 2. Pseudocode pemodelan *Naive Bayes* pada *data training*

C. Klasifikasi Data

Proses klasifikasi data merupakan hal yang paling utama pada penelitian ini, karena proses klasifikasi yang akan menentukan apakah aplikasi ini bisa bekerja atau tidak. Pada tahap ini akan dibuat *code* menggunakan bahasa pemrograman R untuk melakukan klasifikasi dengan algoritma *Naive Bayes*. Hasil dari klasifikasi tersebut yang akan dimanfaatkan oleh pengguna.

D. Visualisasi Data

Hasil dari data yang telah diklasifikasi akan divisualisasikan dengan menggunakan *R Shiny*. Beberapa informasi yang akan divisualisasikan mulai dari prediksi, *accuracy*, *specificity*, *sensitivity* dan sebagainya. Tahap ini adalah hal yang dibutuhkan banyak pengguna karena mereka dapat melihat hasil klasifikasi dari *dataset* yang mereka gunakan.

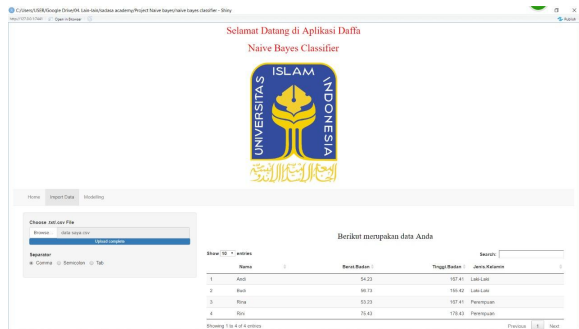
E. Pengujian Sistem

Pada tahap pengujian, sistem akan diuji coba apakah sistem dapat berjalan sesuai rencana, dan dapat memberikan hasil yang diharapkan atau tidak. Tahap ini harus dilakukan secara berulang kali demi mendapatkan hasil yang sempurna. Pengujian dimulai dari penggunaan berbagai macam *dataset* dengan format yang berbeda hingga hasil visualisasinya.

IV. IMPLEMENTASI SISTEM

A. Tampilan Aplikasi

Tampilan pada aplikasi ini dirancang agar *user friendly* dan langsung pada intinya. Tampilan aplikasi menggambarkan judul, penjelasan mengenai aplikasi, *navbar* yang berisi *upload dataset*, dan juga hasil *modeling* dari *Naive Bayes*.



Gambar 3. Tampilan dari aplikasi klasifikasi *Naive Bayes*

B. Dataset

Sistem akan menampilkan *dataset* yang telah di-import oleh pengguna agar dapat memastikan terlebih dahulu, bahwa data yang mereka gunakan sudah sesuai dengan keinginan mereka. Pengguna juga dapat mengatur *separator data* sesuai dengan pemisah pada *dataset* mereka yaitu *comma*, *semicolon*, dan, *tab*.

Index	Activity
1	000001
2	000002
3	000003
4	000004
5	000005
6	000006
7	000007
8	000008
9	000009
10	000010

Gambar 4. Contoh tampilan data yang digunakan pengguna.

Terdapat beberapa syarat *dataset* agar dapat digunakan untuk melakukan klasifikasi menggunakan metode *Naive Bayes classifier*. Data yang akan diinput harus sudah siap untuk dianalisis. Data dikatakan dapat dianalisis jika data sudah rapi dan juga lengkap. Apakah data tersebut secara penulisan sudah sistematis dan apakah terdapat baris atau kolom yang belum terisi. Selanjutnya adalah memastikan data yang digunakan telah melewati proses *data cleaning*. *Data cleaning* adalah proses menghapus data-data yang terduplikat, penulisan data yang salah atau tipografi. Menghapus berbagai macam anomali yang mungkin masih dapat ditemukan pada data tersebut dengan dipenuhinya syarat tersebut, proses klasifikasi data akan jauh lebih mudah dan cepat.

C. Pemodelan Naive Bayes

Pemodelan pada *Naive Bayes* melakukan *splitting data* atau membagi data menjadi data latih (*data training*) dan juga data uji (*data testing*).

Pembagian tersebut bertujuan agar sistem dapat mempelajari *dataset* terlebih dahulu, lalu akan dilakukan pengujian pada data tersebut untuk menentukan tingkat akurasi. Pada sistem ini *splitting data* dapat ditentukan sesuai dengan keinginan pengguna. Pada perhitungan *Naive Bayes*, dilakukan proses *data splitting* atau memisahkan *dataset* menjadi *data training* dan juga *data testing*.

Data testing digunakan untuk menguji apakah metode yang digunakan memberikan nilai atau akurasi yang baik dalam mempelajari *data training*. Hasil dari *data testing* tersebut yang nantinya akan divisualisasikan.

Jika nilai akurasi rendah terdapat beberapa kemungkinan, yaitu kesalahan pada *dataset* yang digunakan. Bisa jadi terjadi kesalahan pada *dataset*, seperti adanya *data missing*, *data outlier*, atau *data imbalance* (data tidak seimbang). Kemungkinan kedua adalah kesalahan pada pemilihan metode *classifier*. Ada kemungkinan *dataset* tersebut tidak cocok digunakan untuk *classifier Naive Bayes*.

D. Visualisasi Naive Bayes

Hasil dari perhitungan *Naive Bayes* tersebut akan divisualisasikan agar pengguna dapat lebih mudah memahami hasil dari klasifikasi yang dilakukan pada *dataset* mereka. Setelah dilakukan perhitungan dengan algoritma *Naive Bayes*, maka hasil dari perhitungan klasifikasi itu akan divisualisasikan melalui *framework R Shiny*.

Pada visualisasi *Naive Bayes*, *package* yang digunakan adalah “caret”. “Caret” merupakan sebuah *package* yang digunakan untuk menentukan apakah metode *Naive Bayes* yang digunakan baik atau tidak dilihat dari nilai *accuracy*, *sensitivity* dan *specificity*. Beberapa informasi penting yang akan ditampilkan adalah *accuracy*, *precision*, *sensitivity*, *specificity*, *F1-Score* dan sebagainya.

Dari informasi yang ditampilkan, pengguna dapat membaca apakah *dataset* yang mereka gunakan pada pemodelan *Naive Bayes* mempunyai akurasi yang tinggi atau tidak. Jika akurasi tinggi, maka pengguna dapat melakukan pengujian data mereka menggunakan *data testing* yang sudah mereka siapkan. Syarat *data testing* yang akan digunakan, tentu saja harus memiliki profil data yang sama dengan data yang telah di *training*. Hasil dari pengujian tersebut akan menghasilkan data baru yang dapat digunakan pengguna sebagai data prediksi.

V. PEMBAHASAN

Dalam perhitungan *Naive Bayes*, diberikan sebuah contoh kasus mengenai keputusan seseorang, apakah ia memutuskan untuk bermain keluar rumah atau tidak berdasarkan cuaca hari ini. Hal utama dalam perhitungan *Naive Bayes* adalah membaca *data training* terlebih dahulu. Data tersebut didapatkan dari *database* aplikasi WEKA yang bernama *weather-nominal*. Data tersebut memiliki atribut berupa *id*, *outlook*, *temperature*, *humidity*, *windy*, dan *play*.

id	outlook	temperature	humidity	windy	play
1	sunny	hot	high	FALSE	no
2	sunny	hot	high	TRUE	no
3	overcast	hot	high	FALSE	yes
4	rainy	mild	high	FALSE	yes
5	rainy	cool	normal	FALSE	yes
6	rainy	cool	normal	TRUE	no
7	overcast	cool	normal	TRUE	yes
8	sunny	mild	high	FALSE	no
9	sunny	cool	normal	FALSE	yes
10	rainy	mild	normal	FALSE	yes
11	sunny	mild	normal	TRUE	yes
12	overcast	mild	high	TRUE	yes
13	overcast	hot	normal	FALSE	yes
14	rainy	mild	high	TRUE	no

Gambar 5. Dataset *weather-nominal*

Pada contoh kasus tersebut, terdapat 14 *data training*. Peluang yang harus dihitung untuk *Naive Bayes* adalah *Class Probabilities* dan *Conditional Probabilities*.

Class Probabilities adalah nilai peluang dari masing-masing kelas. Kelas pada dataset *weather.nominal* adalah variabel “play”, yang memiliki 2 kelas yaitu “yes” atau “no”. Perhitungan dilakukan dengan cara membagi jumlah data untuk masing-masing kelas, dengan jumlah data seluruh kelas. Berikut adalah perhitungannya :

- $$P(\text{play}=\text{yes}) = \frac{\text{Jumlahplay}=\text{yes}}{(\text{Jumlahplay}=\text{yes} + \text{Jumlahplay}=\text{no})}$$

$$P(\text{play}=\text{yes}) = 9 / (9 + 5) = 9 / 14$$

$$P(\text{play}=\text{yes}) = 0.643$$
- $$P(\text{play}=\text{no}) = \frac{\text{Jumlahplay}=\text{no}}{(\text{Jumlahplay}=\text{no} + \text{Jumlahplay}=\text{yes})}$$

$$P(\text{play}=\text{no}) = 5 / (5 + 9) = 5 / 14$$

$$P(\text{play}=\text{no}) = 0.357$$

Dari perhitungan tersebut dapat ditentukan bahwa peluang untuk *play = yes* adalah 0.642 dan *play=no* adalah 0.357.

Selanjutnya menghitung *Conditional Probabilities* yaitu peluang masing-masing nilai terhadap nilai “play”. Apabila data numeris maka dihitung jumlah dan probabilitasnya, apabila data

ordinal, maka perlu dicari rata-rata dan juga standar deviasinya. Pada saat perhitungan data, apabila terdapat data yang bernilai nol(0), maka akan dihitung kembali menggunakan *Laplace Correction*. *Laplace Correction* adalah menambahkan 1 nilai pada sampel, dan menambahkan seluruh jumlah sampel pada sampel. Berikut adalah contoh perhitungan nilai “outlook” terhadap nilai “play”:

- $$P(\text{outlook}=\text{sunny}|\text{play}=\text{yes}) = \frac{\text{Jumlahoutlook}=\text{sunny}|\text{play}=\text{yes}}{(\text{Jumlahplay}=\text{yes})}$$

$$P(\text{outlook}=\text{sunny}|\text{play}=\text{yes}) = 2/9 = 0.222.$$
- $$P(\text{outlook}=\text{sunny}|\text{play}=\text{no}) = \frac{\text{Jumlahoutlook}=\text{sunny}|\text{play}=\text{no}}{(\text{Jumlahplay}=\text{no})}$$

$$P(\text{outlook}=\text{sunny}|\text{play}=\text{no}) = (3+1)/(5+3) = 0.5.$$
- $$P(\text{outlook}=\text{overcast}|\text{play}=\text{yes}) = \frac{\text{Jumlahoutlook}=\text{overcast}|\text{play}=\text{yes}}{(\text{Jumlahplay}=\text{yes})}$$

$$P(\text{outlook}=\text{overcast}|\text{play}=\text{yes}) = 4/9 = 0.444.$$
- $$P(\text{outlook}=\text{overcast}|\text{play}=\text{no}) \text{ (Laplace Correction)} = \frac{\text{Jumlahoutlook}=\text{overcast}|\text{play}=\text{no}}{(\text{Jumlahplay}=\text{no})}$$

$$P(\text{outlook}=\text{overcast}|\text{play}=\text{no}) = (0+1)/(5+3) = 0.125.$$
- $$P(\text{outlook}=\text{rainy}|\text{play}=\text{yes}) = \frac{\text{Jumlahoutlook}=\text{rainy}|\text{play}=\text{yes}}{(\text{Jumlahplay}=\text{yes})}$$

$$P(\text{outlook}=\text{rainy}|\text{play}=\text{yes}) = 3/9 = 0.333.$$
- $$P(\text{outlook}=\text{rainy}|\text{play}=\text{no}) = \frac{\text{Jumlahoutlook}=\text{rainy}|\text{play}=\text{no}}{(\text{Jumlahplay}=\text{no})}$$

$$P(\text{outlook}=\text{rainy}|\text{play}=\text{no}) = (2+1)/(5+3) = 0.375.$$

Setelah dihitung keseluruhan nilai, didapatkan data probabilitas masing-masing sebagai berikut :

Probabilitas “Play”

Jumlah data “play”		Probabilitas	
Yes	No	Yes	No
9	5	0.643	0.357

Probabilitas “Outlook”

Jumlah Data “outlook”	Jumlah data “play”		Probabilitas	
	Yes	No	Yes	No
Sunny	2	3	0.222	0.500
Overcast	4	0	0.444	0.125
Rainy	3	2	0.333	0.375

Probabilitas “Temperature”

Jumlah Data “temperature”	Jumlah data “play”		Probabilitas	
	Yes	No	Yes	No
Hot	2	2	0.222	0.400
Mild	4	2	0.444	0.400
Cool	3	1	0.333	0.200

Probabilitas “Humidity”

Jumlah Data “humidity”	Jumlah data “play”		Probabilitas	
	Yes	No	Yes	No
High	3	4	0.333	0.800
Normal	6	1	0.667	0.200

Probabilitas “Windy”

Jumlah Data “humidity”	Jumlah data “play”		Probabilitas	
	Yes	No	Yes	No
True	3	3	0.333	0.600
False	6	2	0.667	0.400

Berdasarkan perhitungan yang telah dilakukan, hasil nilai probabilitas tersebut akan dihitung menggunakan teorema Bayes. Sebagai contoh digunakan data ke-5 untuk pengujian yaitu

Outlook = Rainy, **Temperature** = Cool, **Humidity** = Normal, **Windy** = FALSE. Berikut adalah perhitungannya :

- $P(\text{play=yes}|X) = (P(\text{outlook} = \text{rainy}|\text{jumlah play} = \text{yes}) * P(\text{temperature} = \text{cool}|\text{jumlah play} = \text{yes}) * P(\text{humidity} = \text{normal}|\text{jumlah play} = \text{yes}) * P(\text{windy} = \text{FALSE}|\text{jumlah play} = \text{yes}) * P(\text{play=yes}))$
 $P(\text{play=yes}|X) = (0.333 * 0.333 * 0.667 * 0.667 * 0.643)$
 $P(\text{play=yes}|X) = 0.03172131$
- $P(\text{play=no}|X) = (P(\text{outlook} = \text{rainy}|\text{jumlah play} = \text{no}) * P(\text{temperature} = \text{cool}|\text{jumlah play} = \text{no}) * P(\text{humidity} = \text{normal}|\text{jumlah play} = \text{no}) * P(\text{windy} = \text{FALSE}|\text{jumlah play} = \text{no}) * P(\text{play=no}))$

$$\text{no}) * P(\text{play=no})$$

$$P(\text{play=no}|X) = (0.375 * 0.200 * 0.200 * 0.400 * 0.357)$$

$$P(\text{play=no}|X) = 0.002142$$

Berdasarkan perhitungan tersebut, didapatkan hasil $P(\text{play=yes}|X)$ dengan nilai paling tinggi, yaitu 0.03172131. Maka prediksi data ke-5 adalah “Yes”. Prediksi tersebut sudah sesuai dengan data yang ada, maka dinamakan *Correctly Classified Instance* yang merupakan hasil akurasi pemodelan. Perhitungan yang sama dilakukan dengan data ke-6, namun prediksi yang dihasilkan adalah “Yes”, yang dimana hasil tersebut tidak sesuai. Perbedaan hasil tersebut tersebut dinamakan *Incorrectly Classified Instance*.

VI. KOMPARASI HASIL

Aplikasi “Waikato Environment for Knowledge Analysis” atau yang disingkat *WEKA* adalah sebuah aplikasi berbasis *desktop*, yang mewajibkan penggunaanya untuk mengunduh aplikasi tersebut sebelum menggunakannya. Berbeda dengan aplikasi yang dikembangkan, pengguna tidak perlu mengunduh aplikasi untuk menggunakannya, karena aplikasi ini berbasis web. Aplikasi yang dikembangkan mengedepankan kemudahan akses dan kepraktisan. Pengguna dapat mengakses aplikasi melalui *website* tanpa perlu repot mengunduh aplikasinya seperti *WEKA*.

Hasil pemodelan yang dilakukan oleh *WEKA*, dibandingkan dengan aplikasi yang dikembangkan memberikan beberapa informasi yang serupa, seperti *Accuracy*, *F1 Score*, *recall* dan sebagainya. Hal yang membedakan adalah pada aplikasi yang dikembangkan, informasi yang diberikan dibuat seringkas mungkin agar pengguna lebih mudah membaca inti dari pemodelan naive bayes, seperti nilai *accuracy*, *f1-score*, *precision*, dan *recall*.

```

=== Evaluation on test split ===

Time taken to test model on test split: 0.01 seconds

=== Summary ===

Correctly Classified Instances      2948      98.2339 %
Incorrectly Classified Instances     53      1.7661 %
Kappa statistic                     0.9626
Mean absolute error                  0.0232
Root mean squared error              0.1361
Relative absolute error              4.9087 %
Root relative squared error          20.1232 %
Total Number of Instances          3001

=== Detailed Accuracy By Class ===

      TP Rate  PP Rate  Precision  Recall  F-Measure  MCC  ROC Area  PRC Area  Class
1.000  0.028  0.955  1.000  0.977  0.963  0.997  0.994  Domestic
0.972  0.000  1.000  0.972  0.986  0.963  0.997  0.998  International
Weighted Avg.  0.982  0.010  0.983  0.982  0.982  0.963  0.997  0.997

=== Confusion Matrix ===
      a  b  <-- classified as
1118   0  |  a = Domestic
 52 1030 |  b = International

```

Gambar 6. Hasil pemodelan *Naive Bayes* menggunakan aplikasi *WEKA*

Model Assessment

```

Confusion Matrix and Statistics

          Domestic International
Domestic    1378             50
International    63          2261

Accuracy : 0.9699
95% CI : (0.9639, 0.9751)
No Information Rate : 0.6159
P-Value [Acc > NIR] : <2e-16

Kappa : 0.9362

McNemar's Test P-Value : 0.259

Sensitivity : 0.9563
Specificity : 0.9784
Pos Pred Value : 0.9650
Neg Pred Value : 0.9729
Precision : 0.9650
Recall : 0.9563
F1 : 0.9606
Prevalence : 0.3841
Detection Rate : 0.3673
Detection Prevalence : 0.3806
Balanced Accuracy : 0.9673

'Positive' Class : Domestic

```

Gambar 7. Hasil pemodelan *Naive Bayes* menggunakan aplikasi yang dikembangkan.

Aplikasi yang dikembangkan memberikan sebuah fasilitas yang tidak diberikan oleh aplikasi *WEKA*, yaitu pengguna dapat menggunakan *data testing* lain dengan syarat, data tersebut memiliki profil data yang sama dengan data yang telah melalui proses pemodelan. Proses tersebut akan menghasilkan *dataset* baru berupa data prediksi. Aplikasi *WEKA* memfasilitasi penggunaanya untuk menggunakan *data testing* lain, namun *WEKA* tidak memberikan *output* berupa *dataset* baru yang berisi aata prediksi.

VII. KESIMPULAN

Berdasarkan penjelasan mengenai penelitian dan sistem, maka dapat disimpulkan bahwa :

1. Aplikasi *classifier* yang dikembangkan menggunakan bahasa pemrograman *R* dan *tool R Shiny* sebagai wadah untuk pembuatan *user interface*.
2. Metode klasifikasi yang digunakan pada aplikasi ini adalah algoritma *Naive Bayes*, karena *Naive Bayes* salah satu metode klasifikasi yang sederhana untuk pengolahan teks dan dapat menghasilkan performa yang baik
3. Aplikasi yang berbasis web ini memberikan kemudahan pada pengguna dalam aksesnya, karena tidak perlu mengunduh aplikasi seperti penggunaan aplikasi *WEKA* pada studi kasus sebelumnya yang masih berbasis *desktop*.
4. Aplikasi ini bertujuan untuk menggali informasi dari *dataset* pengguna. Dengan menggunakan algoritma *Naive Bayes* diharapkan pengguna bisa memperoleh informasi dan prediksi dari *dataset* yang mereka miliki.

VIII. DAFTAR PUSTAKA

- [1] H. B. Uno and N. Lamatenggo, *Teknologi Komunikasi dan Informasi Pembelajaran*, 1st ed. Jakarta: Bumi Aksara, 2010.
- [2] A. Roihan, P. A. Sunarya, and A. S. Rafika, "Pemanfaatan Machine Learning dalam Berbagai Bidang: Review paper," *IJCIT (Indonesian J. Comput. Inf. Technol.)*, vol. 5, no. 1, pp. 75–82, 2020, doi: 10.31294/ijcit.v5i1.7951.
- [3] S. Lorena, "Teknik Data Mining Menggunakan Metode Bayes Classifier Untuk Optimalisasi Pencarian Aplikasi Perpustakaan," *J. Tek. Komput.*, vol. 4, no. 2, pp. 17–20, 2016.
- [4] A. T. Utomo, A. S. Ahmar, and M. K. Aidid, "Pengembangan Paket R untuk Analisis Time Series Dengan Graphical User Interface (GUI)," pp. 1–6, 2017.
- [5] A. Saleh, "Implementasi Metode Klasifikasi Naïve Bayes Dalam Memprediksi Besarnya Penggunaan Listrik Rumah Tangga," *Creat. Inf. Technol. J.*, vol. 2, no. 3, pp. 207–217, 2015.
- [6] D. Animasi, S. Berbasis, and I. G. A. A. Yulistira, "Pemrograman R," vol. 1, pp. 2–7, 2013.
- [7] D. Purnamasari, J. Henharta, Y. P. Sasmita, F. Ihsani, and I. W. S. Wicaksana, "Machine Learning 'Get Easy Using WEKA,'" *Dapur Buku*, pp. 1–40, 2013, [Online]. Available: www.DapurBuku.com.