

## Model Naïve Bayes dan Decision Tree untuk Prediksi Penerimaan Spesifikasi Mobil

Retno Nugroho Whidhiasi<sup>1\*</sup>, Malikus Sumadyo<sup>2</sup>, Inna Ekawati<sup>3</sup>

<sup>1,2,3</sup> Program Studi Teknik Komputer Universitas Islam 45

retno.nw@unismabekasi.ac.id<sup>1</sup>, malikus.sumadyo@unismabekasi.ac.id<sup>2</sup>, inna.ekawati@unismabekasi.ac.id<sup>3</sup>

### Abstrak

Mobil merupakan kebutuhan setiap orang sebagai alat transportasi. Seseorang yang telah berkeluarga akan mempunyai kebutuhan yang lebih terhadap sebuah mobil, terlebih jika mempunyai keluarga besar. Jika mempunyai keluarga besar, memilih sebuah mobil yang tepat untuk keluarga sangatlah penting, dan banyak hal yang harus diperhatikan dalam memilih sebuah mobil. Ketepatan memilih mobil untuk suatu keluarga besar akan berpengaruh terhadap pemenuhan semua kebutuhan transportasi dari setiap anggota keluarga. Dalam penelitian ini dilakukan evaluasi terhadap variabel-variabel yang dipertimbangkan dalam memilih sebuah mobil, yaitu berdasarkan harga beli, biaya maintenance, faktor kenyamanan yang meliputi jumlah pintu, kapasitas orang yang bisa diangkut, ukuran bagasi dan perkiraan keamanan mobil dari data car evaluation. Evaluasi penerimaan dilakukan dengan melakukan prediksi penerimaan menggunakan algoritma naïve bayes dan decision tree classifier. Output dari evaluasi ini adalah penerimaan terhadap variabel-variabel yang dipertimbangkan dalam memilih sebuah mobil tersebut, yang dikategorikan kedalam empat kelas kategori, yaitu unacc, acc, good dan vgood. Kedua algoritma classifier tersebut menghasilkan kinerja yang berbeda, Naïve bayes menghasilkan akurasi 84,1% dan decision tree menghasilkan akurasi yang lebih tinggi yaitu 91,91%.

**Kata Kunci:** prediksi, pengenalan, naïve bayes, pengenalan penerimaan mobil, decision tree

### Abstract

*Cars are a necessity for everyone as a means of transportation. Having a car is a must for someone who has a family, especially one with a big family. If you have a large number of family member, there are many things to consider when choosing the right car for your family. The accuracy of choosing a car for a big family will affect the fulfillment of all transportation needs of each family member. In this study, an evaluation of the variables considered in choosing a car is carried out, namely based on the purchase price, maintenance costs, comfort factors which include the number of doors, the capacity of people that can be carried, luggage size and car safety estimates from car evaluation data. Acceptance evaluation is done by predicting acceptance using the Naive Bayes algorithm and decision tree classifier. The output of this evaluation is acceptance of the variables considered in choosing a car, which is put into four class categories, namely unacc, acc, good and vgood. Those two the classifier algorithm produces different performance, Naïve Bayes produces an accuracy of 84.1% and the decision tree produces a higher accuracy of 91.91%.*

**Keywords:** prediction, recognition, naïve bayes, car evaluation, car specification acceptance, decision tree

## PENDAHULUAN

Mobil merupakan kebutuhan setiap orang sebagai alat transportasi. Seseorang yang telah berkeluarga akan mempunyai kebutuhan yang lebih terhadap sebuah mobil. Dalam sebuah keluarga perlu dipertimbangkan jumlah penumpang yang ikut dalam bepergian menggunakan mobil, termasuk jumlah anak kecil dan jumlah orang dewasa. Terdapat berbagai jenis mobil yang bisa dipilih berdasarkan kapasitasnya. Mobil yang berkapasitas kecil, yaitu 5 kebawah dapat memilih jenis sedan, hatchback, SUV atau MPV. Namun MPV ini merupakan mobil sejenis minibus, mempunyai berkapasitas besar, yaitu dapat memuat 6-7 orang penumpang, bahkan ada yang berkapasitas hingga 9 orang penumpang. Hal yang penting juga adalah bagasi untuk menyimpan berbagai barang keperluan saat bepergian.

Selain hal tersebut perlu dipastikan bahwa mobil mempunyai fitur keamanan dan fitur kenyamanan yang memadai. Fitur keamanan meliputi *automatic lock door & child lock* yang sangat bermanfaat untuk keamanan anak kecil, sabuk pengaman, ABS dan *airbag*. Fitur kenyamanan meliputi suspensi yang nyaman, kabin yang tidak masuk suara-suara dari luar, AC yang bisa mencapai sampai bangku paling belakang dan audio visual yang sangat digemari saat ini. Pertimbangan yang tidak kalah penting adalah mengenai budget dan biaya, baik untuk pembelian maupun perawatan bulanan.

Dengan banyaknya variabel yang merupakan penentu penerimaan sebuah mobil pastilah terdapat banyak kombinasi dari penerimaan setiap variabelnya yang pada akhirnya masuk ke dalam kelas kategori tidak diterima, diterima, bagus atau sangat bagus. Informasi tersebut sangat diperlukan oleh produsen sebagai pihak yang memproduksi untuk dapat memprediksi tingkat penerimaan konsumen juga oleh konsumen sebagai dasar untuk memutuskan memilih mobil yang akan dibeli. Kesulitan yang sering terjadi adalah pada saat konsumen telah mengetahui spesifikasi dan biaya-biaya dari suatu mobil yang diminati adalah kurang cepatnya dalam berpikir untuk menyimpulkan apakah mobil tersebut cocok untuk kebutuhannya dan layak untuk dibeli atau tidak. Untuk membantu konsumen agar lebih cepat dan efisien dalam memprediksi penerimaan terhadap spesifikasi yang ada pada sebuah mobil diperlukan alat bantu berupa sebuah model untuk memprediksi penerimaan suatu mobil berdasarkan variabel-variabel tersebut menggunakan algoritma tertentu dalam machine learning.

Terdapat banyak macam algoritma yang dapat digunakan untuk melakukan suatu prediksi, identifikasi maupun klasifikasi, diantaranya adalah *naïve bayes* dan *decision tree*. Model hasil *learning Naïve bayes* adalah membentuk model berupa probabilitas, sedangkan model hasil *learning decision tree* adalah model pohon keputusan. Menurut riset-riset yang telah dilakukan, kedua model tersebut dapat digunakan dengan akurasi yang cukup bagus.

Klasifikasi metagenom menggunakan *naïve bayes* dengan ekstraksi ciri perhitungan frekuensi *k-mer* untuk fragmen panjang (10 kbp) memberikan akurasi 82,23% dengan ekstraksi ciri *3-mer* dan 85,89% dengan ekstraksi ciri *4-mer* [1]. Deteksi spam pada twitter menggunakan *naïve bayes* dengan tahap *tokenizing*, *filtering*, normalisasi kata dan *stemming* menghasilkan akurasi 95,57% [2]. Pendekatan *naïve bayes* mampu mengidentifikasi dengan tepat masukan berdasarkan *rule*, jenis penyakit kedelai di dataran tinggi sebesar 49% dan jenis penyakit kedelai di dataran rendah sebesar 75% [3]. Klasifikasi peminatan siswa SMA telah dilakukan dengan model hasil *learning* algoritma *Naïve Bayes* yang dimulai dari tahap persiapan hingga pengujian memiliki nilai akurasi yang sangat baik yaitu sebesar 99,47% [4]. Klasifikasi pasien hepatocellular carcinoma (HCC) dari dataset RS Islam Bandung menggunakan *naïve bayes* menghasilkan akurasi 98,25% dan *decision tree* menghasilkan akurasi 100% [5].

*Decision tree* C.45 dapat diterapkan untuk memprediksi lahan kritis di daerah budidaya pertanian dengan akurasi 92,47% [6]. Model evaluasi beasiswa pendidikan tinggi yang dibangun menggunakan pohon keputusan C.45 dapat mewujudkan suatu sistem evaluasi beasiswa yang efisien dan adil [7]. Untuk mendiagnosis penyakit diabetes meletus lebih dianjurkan menggunakan algoritma C.45 dengan ketelitian sebesar 100% dibandingkan dengan algoritma *naïve bayes* yang mempunyai ketelitian sebesar 90% [8]. Klasifikasi menggunakan C.45 untuk menentukan status gizi memberikan akurasi sebesar 90,16%, 76,64% dan 83,83% berdasarkan berat badan/umur, tinggi badan/umur dan berat badan / tinggi badan [9]. Pemakaian pohon keputusan untuk menentukan apakah seorang calon mahasiswa dapat diterima sebagai penerima beasiswa atau tidak memberikan akurasi 87% [10].

Dalam penelitian ini dilakukan prediksi penerimaan terhadap variabel-variabel yang dipertimbangkan dalam memilih sebuah mobil, yaitu berdasarkan perhitungan biaya dan faktor kenyamanan menggunakan algoritma *naïve bayes* dan *decision tree*, sehingga diperoleh prediksi kategori penerimaan berdasar variabel-variabel yang dipertimbangkan dalam memilih sebuah mobil, yaitu *unacc*, *acc*, *good* dan *vgood*.

## METODE PENELITIAN

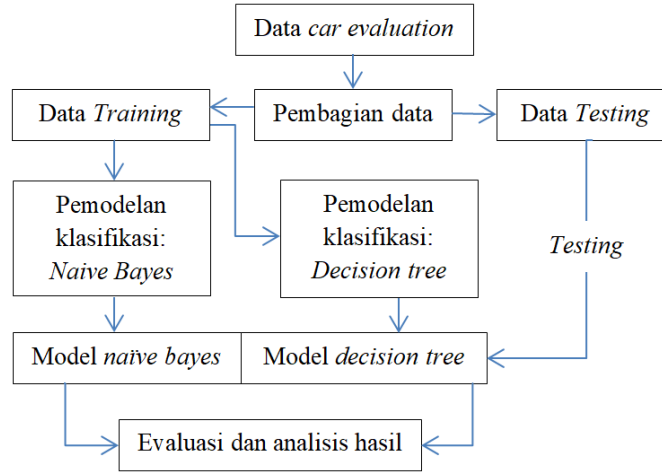
Pengenalan variabel-variabel yang dipertimbangkan dalam memilih mobil ini terbagi menjadi empat tahapan, yaitu pengumpulan data, pembagian data, pemodelan klasifikasi, *testing* dan evaluasi dan analisis hasil, disajikan pada Gambar 1. Pengumpulan data dilakukan untuk mendapatkan data yang akan digunakan sebagai *knowledge base* yang digunakan untuk data pelatihan (*training*). Pembagian data dilakukan untuk mempersiapkan data agar siap untuk diproses lebih lanjut menggunakan *classifier*. Proses *training* akan dilakukan menggunakan *naïve bayes* dan *decision tree*.

### 1. Data Car Evaluation

Data yang digunakan merupakan data sekunder diambil dari *UCI machine learning repository*. Data set yang digunakan berupa data *car evaluation*, yang berupa penerimaan terhadap spesifikasi suatu mobil. Variabel penerimaan tersebut meliputi harga keseluruhan dan karakteristik teknis. Harga keseluruhan mencakup harga beli dan harga pemeliharaan. Karakteristik teknis mencakup kenyamanan dan perkiraan keamanan mobil. Faktor kenyamanan meliputi jumlah pintu mobil, kapasitas orang yang

dapat diangkut dan ukuran bagasi.

Kategori kelas dalam data tersebut terdiri dari empat kelas kategori dan 6 atribut. Kelas kategori terdiri dari *unacc*, *acc*, *good* dan *vgood*. Variabel terdiri dari *buying*, *maint*, *doors*, *persons*, *lug\_boot* dan *safety*. Nilai-nilai dari tiap varibel disajikan pada Tabel 1.



Gambar 1. Tahapan penelitian

Tabel 1. Variabel dalam dataset

<i>buying</i>	<i>maint</i>	<i>doors</i>	<i>persons</i>	<i>lug_boot</i>	<i>safety</i>
<i>vhigh</i>	<i>vhigh</i>	2	2	<i>small</i>	<i>low</i>
<i>high</i>	<i>high</i>	3	4	<i>med</i>	<i>med</i>
<i>med</i>	<i>med</i>	4	<i>more</i>	<i>big</i>	<i>high</i>
<i>low</i>	<i>low</i>	<i>5more</i>			

## 2. Pembagian Data

Pada tahap pembagian data, dilakukan pembagian dataset menjadi dua kelompok data yang saling asing, yaitu data *training* dan data *testing*. Proporsi pembagian tersebut sebesar 80% untuk data *training* dan 20% untuk data *testing* yang dilakukan secara acak. Data *training* merupakan data yang mempunyai label atau kelas kategori sedangkan data *testing* merupakan data yang label atau kelas kategorinya telah dipisahkan untuk digunakan untuk *testing*, kemudian baru dicocokkan dengan label atau kelas kategori untuk mengetahui ketepatan prediksinya.

## 3. Pemodelan klasifikasi

Pemodelan dilakukan menggunakan Rapidminer Studio Versi 9.10 yang dijalankan pada laptop Core i5-10210U CPU @1.60GHz 2.11GHz RAM 16GB dengan sistem operasi windows 11Pro. Pemodelan klasifikasi dilakukan menggunakan algoritma *naïve bayes* dan *decision tree* yang akan menghasilkan dua model yang berbeda, yaitu model probabilitas dan model pohon keputusan. Proses training pada kedua algoritma tersebut menggunakan metode *supervised learning*, yaitu sistem melakukan *training* dengan panduan berupa label atau kelas kategori dalam menentukan proses.

### 3.1 Pemodelan Naive Bayes

Untuk menentukan nilai probabilitas dalam algoritma *naive bayes* dihitung menggunakan persamaan (1):

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}, \dots\dots\dots (1)$$

Keterangan = H adalah suatu kelas spesifik, X adalah suatu fitur {X<sub>1</sub>, X<sub>2</sub>,..., X<sub>n</sub>}, P(H|X) adalah posterior, yaitu Probabilitas berdasar kondisi atau probabilitas bersyarat, P(H) adalah *prior probability*, yaitu peluang kemunculan sebuah kelas dari seluruh pengamatan yang dilakukan, P(X|H) adalah likelihood, yaitu peluang kemunculan nilai-nilai fitur pada kelas tertentu, dan P(X) adalah evidence, yaitu

peluang kemunculan nilai-nilai fitur dari semua pengamatan [11].

Dikatakan prior karena peluang ini dijadikan pengetahuan awal atau pengetahuan dasar atau prior believe atau kepercayaan dasar, sebelum memperhatikan faktor lainnya. Evidence seringkali tidak perlu dihitung karena dalam permasalahan klasifikasi yang dicari adalah nilai peluang terbesar, sedangkan evidence pada rumus bayes merupakan penyebut yang nilainya sama untuk setiap kelas.

**3.2 Pemodelan Decision Tree**

Algoritma decision tree merupakan diagram dengan struktur yang terdiri dari node-node, tiap node merepresentasikan data dalam atribut. Klasifikasi decision tree berasumsi pada pengamatan label pada objek yang mirip satu sama lain pada variabel yang berbeda. Kumpulan data yang digunakan pada decision tree terdiri dari atribut dan data record yang dinyatakan dalam bentuk tabel. Parameter yang digunakan sebagai kriteria dalam pembuatan hirarki adalah atribut [11]. Untuk menentukan atribut terpilih, terdapat banyak pendekatan yang dapat digunakan, salah satunya adalah gain ratio. Membangun pohon keputusan menggunakan kriteria gain ratio dilakukan dengan algoritma berikut:

1. Nilai gain tertinggi pada masing-masing atribut akan dipilih sebagai akar, dihitung dengan persamaan (2):

$$Gain(S,A) = Entropy(S) - \sum_{i=1}^n \frac{|S_i|}{S} * Entropy(S_i) \dots\dots\dots (2)$$

Keterangan = S adalah himpunan suatu kasus, A adalah fitur atau atribut, n adalah jumlah partisi pada atribut A, |Si| adalah jumlah kasus pada partisi ke-i dan |S| adalah jumlah kasus dalam S. Berikut persamaan (3) untuk menghitung nilai entropy:

$$Entropy(S) = \sum_{i=1}^n -p_i * \log_2 p_i \dots\dots\dots (3)$$

Keterangan = S adalah himpunan suatu kasus, A adalah fitur atau atribut, n adalah jumlah partisi S, dan Pi adalah proporsi dari Si terhadap S.

2. Membuat cabang pada masing-masing nilai
3. Membagi kasus dalam cabang
4. Mengulangi proses untuk setiap cabang sampai semua kasus pada cabang memiliki kelas yang sama.

**5. Testing (pengujian)**

Model hasil *learning* dari algoritma *naïve bayes* dan *decision tree* digunakan untuk melakukan *testing* atau pengujian terhadap sejumlah data *testing* yang dipisahkan label atau kelas kategorinya. Kemudian hasilnya akan dicocokkan dengan label atau kelas kategorinya untuk mengetahui kemampuan pengenalan kedua model tersebut. Hasil testing dipetakan dalam sebuah matriks konfusi, sehingga akan diketahui prediksi benar dan prediksi yang salah dilakukan. Dari matriks konfusi kemudian akan dihitung kemampuan pengenalannya atau akurasi, yang dihitung dengan persamaan (4):

$$Akurasi = \frac{\sum \text{data uji benar}}{\sum \text{data uji}} * 100\% \dots\dots\dots (4)$$

**6. Evaluasi dan analisis hasil**

Model *naive bayes* dan *decision tree* beserta hasil pengujian dilakukan evaluasi dan analisis terkait kemampuannya dalam memprediksi atau mengenali kategori penerimaan *car evaluation* pada penelitian ini.

**HASIL DAN PEMBAHASAN**

Dalam penelitian ini dilakukan pengenalan menggunakan algoritma *naïve bayes* dan *decision tree*. Kedua algoritma ini akan melakukan proses *training* atau pembelajaran untuk mengenali suatu pola dari suatu dataset sehingga terbentuk model *knowledge*, yang kemudian akan digunakan untuk mengenali suatu kondisi berdasarkan variabel-variabel.

Data *car evaluation* yang merupakan variabel-variabel yang dipertimbangkan dalam pemilihan mobil diolah, kemudian *ditraining* dan dilakukan *testing*. Berdasarkan hal tersebut maka penelitian ini dibagi dalam empat tahap, yaitu pengumpulan data, pra proses data, *training* dan *testing*.

**1. Data Car Evaluation**

Langkah pertama adalah mengumpulkan data yang digunakan sebagai data *training* dan *testing*.

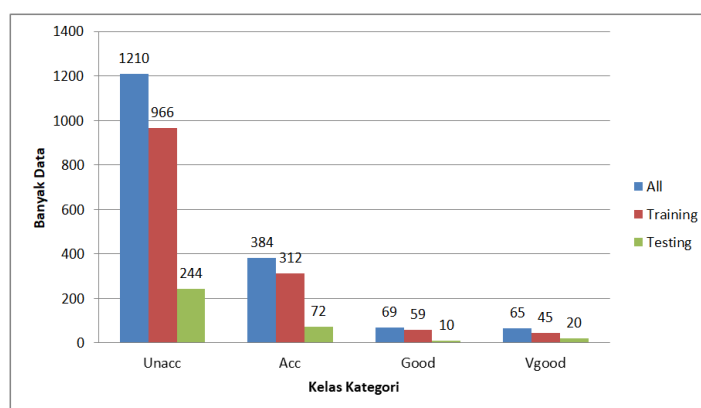
Data *training* digunakan oleh algoritma *decision tree* dan *naïve bayes* untuk dipelajari, sehingga mendapatkan *knowledge* atau model yang akan digunakan selanjutnya jika model tersebut sudah cukup pintar.

Data yang digunakan adalah dataset *car evaluation* dari *UCI machine learning repository* sejumlah 1728 data yang terdiri dari empat kelas kategori, yaitu *unacc*, *acc*, *good* dan *vgood*. Kelas kategori *unacc* sejumlah 1210 data, kategori *acc* sejumlah 384 data, kategori *good* sejumlah 69 dan kategori *vgood* sejumlah 65 data. Prosentase kategori *unacc* sebanyak 70%, kategori *acc* sejumlah 22%, kategori *good* sejumlah 3,99% dan kategori *vgood* sejumlah 3,76%.

## 2. Pembagian Data

Data set dibagi menjadi data *training* dan data *testing* dengan proporsi 80% data *training* dan 20% data *testing*, sehingga jumlah data *training* adalah 1382 data dan data *testing* sejumlah 346 data.

Data *training* terdiri dari 966 data kategori *unacc*, 312 data kategori *acc*, 59 data kategori *good* dan 45 data kategori *vgood*. Data *testing* terdiri dari 244 data kategori *unacc*, 72 data kategori *acc*, 10 data kategori *good* dan 20 data kategori *vgood*. Komposisi data keseluruhan, data *training* dan *testing* tersebut disajikan pada Gambar 2.



Gambar 2. Komposisi data

## 3. Pemodelan Klasifikasi

Objek utama dari *machine learning* adalah data. Data digunakan untuk proses belajar sehingga memiliki *knowledge*. Data digunakan sebagai data *training* yang dilatih dan menghasilkan pengetahuan yang disebut *machine learning model*. Proses *training* (pelatihan) ini pada dasarnya adalah melakukan proses belajar berdasarkan pasangan data input dan output. Pelatihan dilakukan untuk membentuk model dari pembelajaran terhadap data *training*, sehingga jika diberikan data input model ini akan mampu memprediksi nilai outputnya. Dari model yang terbentuk akan dapat diketahui apakah model tersebut sudah cukup bagus atau perlu diperbaiki kembali.

Sistem *machine learning* dalam penelitian ini digunakan untuk mengenali rekomendasi penerimaan suatu mobil berdasarkan variabel-variabel yang dipertimbangkan kedalam empat kategori, yaitu *unacc*, *acc*, *good* dan *vgood*. Terdapat 6 variabel yang dipertimbangkan dalam mengenali rekomendasi penerimaan mobil, yaitu *buying* (harga beli), *maint* (biaya maintenance), faktor kenyamanan yang meliputi *doors* (jumlah pintu), *persons* (kapasitas orang yang bisa diangkut), *lug\_boot* (ukuran bagasi) dan *safety* (perkiraan keamanan).

Data yang telah disiapkan kemudian dilakukan pelatihan atau proses belajar sehingga terbentuk *knowledge* atau model. Model yang dihasilkan berbeda antara *naïve bayes* dan *decision tree*. *Naïve bayes* menggunakan teori probabilitas atau peluang, dengan konsep. Oleh karena itu pelatihan menggunakan algoritma *naïve bayes* akan menghasilkan model berupa probabilitas, sedangkan proses pelatihan menggunakan algoritma *decision tree* akan menghasilkan model berupa pohon keputusan.

### 3.1 Model naïve bayes

Algoritma *naïve bayes classifier* merupakan algoritma *supervised learning* sehingga membutuhkan label atau supervisor yang bersifat mengarahkan dalam pembelajaran, yang berfungsi

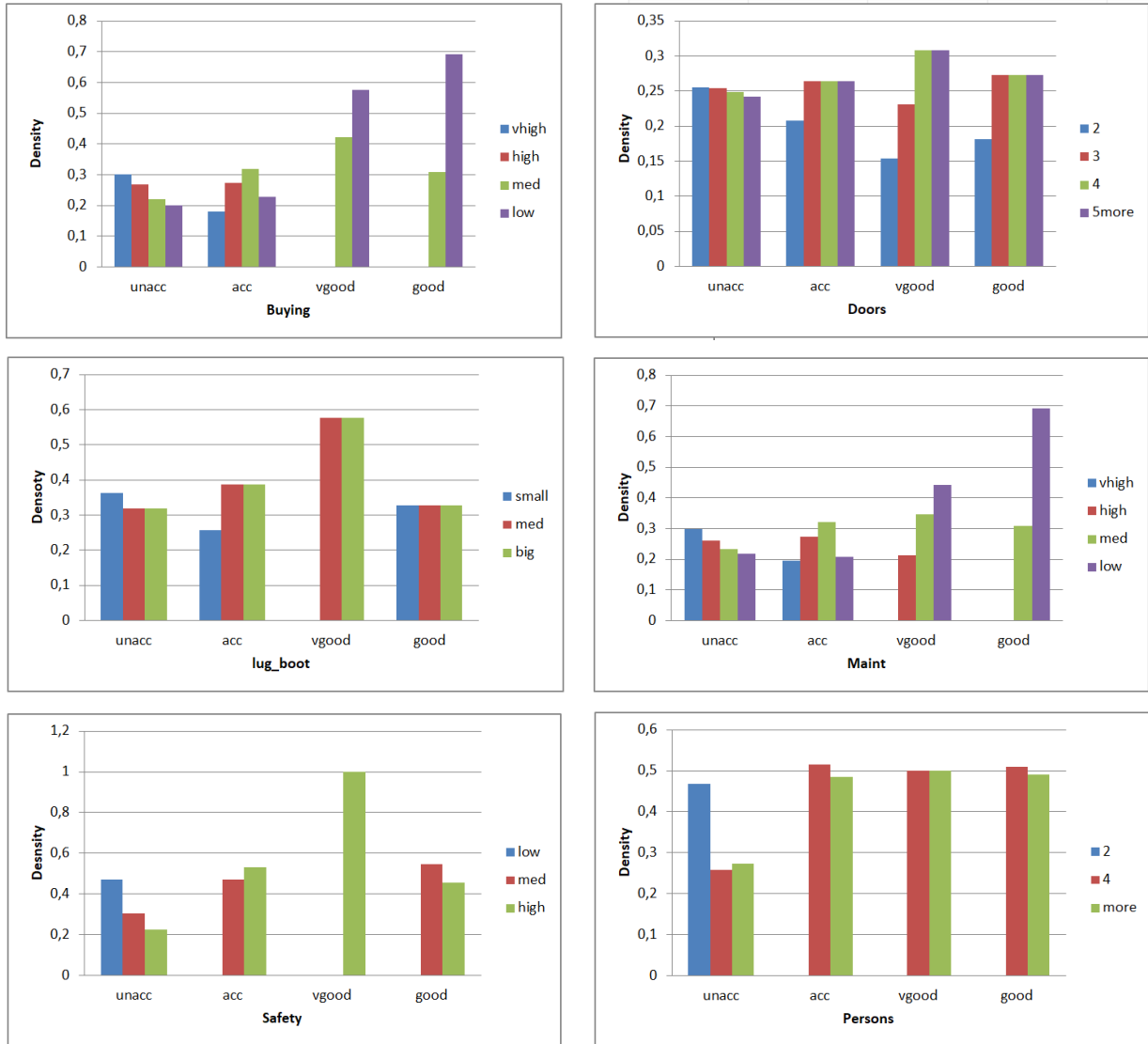
untuk mengoreksi jika terdapat kesalahan. Label merupakan kelompok dari masing-masing data. Dalam klasifikasi *car evaluation* ini yang berlaku sebagai label atau supervisor adalah *unacc*, *acc*, *good* dan *vgood*.

Algoritma *naïve bayes* menggunakan teori probabilitas atau peluang, dengan konsep dasar menghitung peluang dari setiap kondisi. Model *machine learning* yang yang dihasilkan dari proses *training* menggunakan *naïve bayes classifier* adalah model yang berupa nilai probabilitas dari masing-masing label atau kategori, yang dilakukan dengan mempelajari pola data *training* dan menghitung kemungkinan kemunculan tiap nilai yang muncul berdasarkan kategori, sehingga akan terbentuk nilai-nilai probabilitas dan nilai probabilitas bersyarat dari hasil *learning dataset*.

Probabilitas tersebut dihitung dari jumlah tiap kategori dari (per) total seluruh data dan probabilitas bersyarat untuk tiap kolom atribut. Probabilitas bersyarat dihitung dari probabilitas pada tiap kolom atribut berdasarkan label. Model hasil *learning* yang dihasilkan oleh *naïve bayes classifier* yang berupa probabilitas disajikan pada Tabel 2 dan Gambar 3.

**Tabel 2. Model hasil *learning naïve bayes classifier***

Atribut	Parameter	unacc	acc	vgood	good
buying	value=vhigh	0,302	0,179	0	0
buying	value=high	0,268	0,274	0	0
buying	value=med	0,221	0,319	0,423	0,309
buying	value=low	0,2	0,228	0,577	0,691
buying	value=unknown	0	0	0	0
doors	value=2	0,255	0,208	0,154	0,182
doors	value=3	0,254	0,264	0,231	0,273
doors	value=4	0,249	0,264	0,308	0,273
doors	value=5more	0,242	0,264	0,308	0,273
doors	value=unknown	0	0	0	0
lug_boot	value=small	0,362	0,257	0	0,327
lug_boot	value=med	0,319	0,388	0,577	0,327
lug_boot	value=big	0,319	0,388	0,577	0,327
lug_boot	value=unknown	0	0	0	0
maint	value=vhigh	0,3	0,195	0	0
maint	value=high	0,26	0,274	0,212	0
maint	value=more	0,233	0,322	0,346	0,309
maint	value=low	0,217	0,208	0,442	0,691
maint	value=unknown	0	0	0	0
persons	value=2	0,468	0	0	0
persons	value=4	0,259	0,515	0,5	0,509
persons	value=more	0,273	0,485	0,5	0,491
persons	value=unknown	0	0	0	0
safety	value=low	0,471	0	0	0
safety	value=med	0,305	0,469	0	0,545
safety	value=high	0,224	0,531	1	0,455
safety	value=unknown	0	0	0	0



Gambar 3. Model *learning naïve bayes classifier*

**Model Decision Tree**

Data dilatih menggunakan *decision tree* menggunakan *pruning* dan *prepruning*. *Pruning* adalah pemangkasan setelah pohon keputusan dihasilkan, sedangkan *prepruning* adalah pemangkasan pada saat pohon keputusan dibangun. Pelatihan dilakukan dengan empat kriteria yaitu *gain ratio*, *information gain* dan *gini index*, dengan *pruning* dan *prepruning*. Kriteria tersebut digunakan untuk memutuskan atribut mana yang akan ditempatkan pada akar atau pada level berbeda dari pohon sebagai simpul internal.

*Gain ratio* adalah rasio perolehan informasi yang didapat dari informasi intrinsik, dilakukan untuk mengurangi bias terhadap atribut multi-nilai dengan mempertimbangkan jumlah dan ukuran cabang saat memilih atribut. *Information Gain* merupakan suatu metode yang digunakan untuk melakukan pembobotan terhadap suatu fitur menggunakan teknik scoring, yaitu menggunakan nilai entropy yang paling besar. Fitur yang mempunyai nilai *Information Gain* lebih besar atau sama dengan nilai *threshold* tertentu adalah fitur yang dipilih. *Gini index* adalah salah satu kriteria yang dapat digunakan untuk menentukan titik pemecah terbaik (*optimal splitting point*).

Pelatihan dengan menggunakan empat kriteria tersebut menghasilkan model yang berbeda. Berikut adalah model hasil dari pelatihan *decision tree* menggunakan *gain ratio*:

**Tree**

```

safety = high
| persons = 2: unacc {unacc=146, acc=0, vgood=0, good=0}
| persons = 4
| | buying = high
| | | maint = high: acc {unacc=0, acc=10, vgood=0, good=0}
| | | | maint = low: acc {unacc=0, acc=10, vgood=0, good=0}
| | | | maint = med: acc {unacc=0, acc=7, vgood=0, good=0}
| | | | maint = vhigh: unacc {unacc=10, acc=0, vgood=0, good=0}
| | | buying = low
| | | | maint = high
| | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=4, good=0}
| | | | | lug_boot = med: acc {unacc=0, acc=2, vgood=2, good=0}
| | | | | lug_boot = small: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | maint = low
| | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=4, good=0}
| | | | | | lug_boot = med: good {unacc=0, acc=0, vgood=1, good=2}
| | | | | | lug_boot = small: good {unacc=0, acc=0, vgood=0, good=4}
| | | | | maint = med
| | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=2, good=0}
| | | | | | lug_boot = med: vgood {unacc=0, acc=0, vgood=2, good=2}
| | | | | | lug_boot = small: good {unacc=0, acc=0, vgood=0, good=4}
| | | | | maint = vhigh: acc {unacc=0, acc=10, vgood=0, good=0}
| | | buying = med
| | | | maint = high: acc {unacc=0, acc=10, vgood=0, good=0}
| | | | | maint = low
| | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=3, good=0}
| | | | | | lug_boot = med: vgood {unacc=0, acc=0, vgood=2, good=1}
| | | | | | lug_boot = small: good {unacc=0, acc=0, vgood=0, good=3}
| | | | | | maint = med
| | | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=4, good=0}
| | | | | | | lug_boot = med: acc {unacc=0, acc=2, vgood=2, good=0}
| | | | | | | lug_boot = small: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | lug_boot = small: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | maint = vhigh: acc {unacc=0, acc=10, vgood=0, good=0}
| | | buying = vhigh
| | | | maint = high: unacc {unacc=10, acc=0, vgood=0, good=0}
| | | | | maint = low: acc {unacc=0, acc=5, vgood=0, good=0}
| | | | | maint = med: acc {unacc=0, acc=11, vgood=0, good=0}
| | | | | maint = vhigh: unacc {unacc=10, acc=0, vgood=0, good=0}
| persons = more
| | buying = high
| | | maint = high
| | | | | lug_boot = big: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | lug_boot = med: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | lug_boot = small: unacc {unacc=1, acc=1, vgood=0, good=0}
| | | | | | maint = low
| | | | | | | doors = 2: unacc {unacc=1, acc=1, vgood=0, good=0}
| | | | | | | doors = 4: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | doors = 5more: acc {unacc=0, acc=2, vgood=0, good=0}
| | | | | | | maint = med: acc {unacc=1, acc=10, vgood=0, good=0}
| | | | | | | maint = vhigh: unacc {unacc=10, acc=0, vgood=0, good=0}
| | | buying = low
| | | | maint = high
| | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=3, good=0}
| | | | | | | lug_boot = med: vgood {unacc=0, acc=1, vgood=2, good=0}
| | | | | | | | lug_boot = small: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | | | | maint = low
| | | | | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=4, good=0}
| | | | | | | | | | lug_boot = med: vgood {unacc=0, acc=0, vgood=3, good=0}
| | | | | | | | | | | lug_boot = small: good {unacc=1, acc=0, vgood=0, good=3}
| | | | | | | | | | | | maint = med: good {unacc=1, acc=0, vgood=3, good=4}
| | | | | | | | | | | | maint = vhigh: acc {unacc=0, acc=9, vgood=0, good=0}
| | | buying = med
| | | | | maint = high: acc {unacc=1, acc=9, vgood=0, good=0}
| | | | | | maint = low
| | | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=3, good=0}
| | | | | | | | lug_boot = med: vgood {unacc=0, acc=0, vgood=3, good=1}
| | | | | | | | | lug_boot = small: unacc {unacc=1, acc=0, vgood=0, good=1}
| | | | | | | | | | maint = med
| | | | | | | | | | | | lug_boot = big: vgood {unacc=0, acc=0, vgood=2, good=0}
| | | | | | | | | | | | | lug_boot = med: vgood {unacc=0, acc=1, vgood=3, good=0}
| | | | | | | | | | | | | | lug_boot = small: acc {unacc=1, acc=3, vgood=0, good=0}
| | | | | | | | | | | | | | | maint = vhigh: acc {unacc=1, acc=10, vgood=0, good=0}
| | | buying = vhigh
| | | | | maint = high: unacc {unacc=10, acc=0, vgood=0, good=0}
| | | | | | maint = low: acc {unacc=1, acc=10, vgood=0, good=0}
| | | | | | | maint = med

```



```

| | | | doors = 2: unacc {unacc=1, acc=1, vgood=0, good=0}
| | | | doors = 3: acc {unacc=0, acc=2, vgood=0, good=0}
| | | | doors = 4: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | doors = 5more: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | maint = vhigh: unacc {unacc=9, acc=0, vgood=0, good=0}
safety = low: unacc {unacc=456, acc=0, vgood=0, good=0}
safety = med
| | | | persons = 2: unacc {unacc=156, acc=0, vgood=0, good=0}
| | | | persons = 4
| | | | | buying = high
| | | | | | lug_boot = big
| | | | | | | maint = high: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | maint = low: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | maint = med: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | maint = vhigh: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | | | | lug_boot = med: unacc {unacc=8, acc=3, vgood=0, good=0}
| | | | | | | | lug_boot = small: unacc {unacc=13, acc=0, vgood=0, good=0}
| | | | | buying = low
| | | | | | maint = high: acc {unacc=0, acc=10, vgood=0, good=0}
| | | | | | | maint = low
| | | | | | | | lug_boot = big: good {unacc=0, acc=0, vgood=0, good=3}
| | | | | | | | | lug_boot = med: good {unacc=0, acc=1, vgood=0, good=2}
| | | | | | | | | lug_boot = small: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | maint = med
| | | | | | | | | lug_boot = big: good {unacc=0, acc=0, vgood=0, good=2}
| | | | | | | | | | lug_boot = med: acc {unacc=0, acc=2, vgood=0, good=1}
| | | | | | | | | | lug_boot = small: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | maint = vhigh
| | | | | | | | | lug_boot = big: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | | | lug_boot = med: unacc {unacc=2, acc=1, vgood=0, good=0}
| | | | | | | | | | | lug_boot = small: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | buying = med
| | | | | | maint = high
| | | | | | | lug_boot = big: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | lug_boot = med: unacc {unacc=2, acc=1, vgood=0, good=0}
| | | | | | | | | lug_boot = small: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | | | | | maint = low
| | | | | | | | | lug_boot = big: good {unacc=0, acc=0, vgood=0, good=3}
| | | | | | | | | | lug_boot = med: acc {unacc=0, acc=1, vgood=0, good=1}
| | | | | | | | | | | lug_boot = small: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | | maint = med: acc {unacc=0, acc=11, vgood=0, good=0}
| | | | | | | | | maint = vhigh
| | | | | | | | | | lug_boot = big: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | | | | lug_boot = med: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | | | | | | | | lug_boot = small: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | | buying = vhigh
| | | | | | maint = high: unacc {unacc=11, acc=0, vgood=0, good=0}
| | | | | | | maint = low
| | | | | | | | lug_boot = big: acc {unacc=0, acc=2, vgood=0, good=0}
| | | | | | | | | lug_boot = med: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | | | | | | lug_boot = small: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | | | | | | maint = med
| | | | | | | | | | lug_boot = big: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | | | | lug_boot = med: unacc {unacc=2, acc=2, vgood=0, good=0}
| | | | | | | | | | | | lug_boot = small: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | | | | | | | | | maint = vhigh: unacc {unacc=11, acc=0, vgood=0, good=0}
| | | | | persons = more
| | | | | | lug_boot = big
| | | | | | | maint = high
| | | | | | | | buying = high: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | | buying = low: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | | | buying = med: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | | | | buying = vhigh: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | | | | | | | | maint = low
| | | | | | | | | | | | | buying = high: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | | | | | | | buying = low: good {unacc=0, acc=0, vgood=0, good=4}
| | | | | | | | | | | | | | | buying = med: good {unacc=0, acc=0, vgood=0, good=4}
| | | | | | | | | | | | | | | | buying = vhigh: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | | | | | | | | | | | | | maint = med: acc {unacc=0, acc=9, vgood=0, good=2}
| | | | | | | | | | | | | | | | | | maint = vhigh
| | | | | | | | | | | | | | | | | | | buying = high: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | | | | | | | | | | | | | | | | | buying = low: acc {unacc=0, acc=2, vgood=0, good=0}
| | | | | | | | | | | | | | | | | | | | | buying = med: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | | | | | | | | | | | | | | | | | | buying = vhigh: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | | | | | | | | | | | | | | | | | | | lug_boot = med
| | | | | | | | | | | | | | | | | | | | | | | | | maint = high
| | | | | | | | | | | | | | | | | | | | | | | | | | | doors = 2: unacc {unacc=2, acc=1, vgood=0, good=0}

```

```

| | | | doors = 3: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | doors = 4: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | doors = 5more: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | maint = low
| | | | | buying = high: acc {unacc=0, acc=3, vgood=0, good=0}
| | | | | buying = low: good {unacc=0, acc=0, vgood=0, good=3}
| | | | | buying = med: good {unacc=0, acc=1, vgood=0, good=3}
| | | | | buying = vhigh: acc {unacc=1, acc=3, vgood=0, good=0}
| | | | maint = med
| | | | | buying = high: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | buying = low: good {unacc=0, acc=1, vgood=0, good=2}
| | | | | buying = med: acc {unacc=0, acc=4, vgood=0, good=0}
| | | | | buying = vhigh: acc {unacc=1, acc=3, vgood=0, good=0}
| | | | maint = vhigh
| | | | | buying = high: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | buying = low: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | buying = med: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | buying = vhigh: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | lug_boot = small
| | | | | buying = high: unacc {unacc=12, acc=0, vgood=0, good=0}
| | | | | buying = low
| | | | | | doors = 2: unacc {unacc=3, acc=0, vgood=0, good=0}
| | | | | | doors = 3: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | | doors = 4: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | | doors = 5more: acc {unacc=0, acc=2, vgood=0, good=0}
| | | | | buying = med
| | | | | | maint = high: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | | | maint = low: acc {unacc=1, acc=2, vgood=0, good=0}
| | | | | | maint = med: acc {unacc=1, acc=3, vgood=0, good=0}
| | | | | | maint = vhigh: unacc {unacc=4, acc=0, vgood=0, good=0}
| | | | | buying = vhigh: unacc {unacc=13, acc=0, vgood=0, good=0}

```

**Pengujian (testing)**

**Pengujian model naïve bayes**

Model hasil pelatihan menggunakan *naïve bayes* kemudian dilakukan pengujian menggunakan data uji atau data testing yang saling asing dengan data training. Model *naïve bayes* akan memilih yang memiliki probabilitas yang paling tinggi untuk menentukan suatu data masuk ke dalam suatu kelas atau kategori.

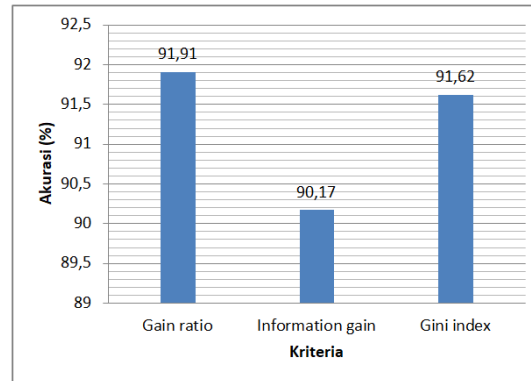
Model *naïve bayes* dapat mengenali dengan tepat kelas kategori unacc sebanyak 235 data atau 90,04% dan 26 terprediksi sebagai acc. Kelas kategori acc dikenali dengan tepat sebanyak 46 data atau 66,67%, 7 data terprediksi sebagai unacc, 4 data terprediksi sebagai vgood dan 12 data terprediksi sebagai good. Kelas kategori vgood dikenali dengan tepat sebanyak 9 data atau 90% dan 1 data terprediksi sebagai good. Kelas kategori good dikenali dengan tepat sebanyak 1 atau 16,67% dan 5 data terprediksi sebagai acc, sehingga akurasi yang didapatkan sebesar 84,10%. Pengenalan tersebut disajikan dalam matriks konfusi pada Tabel 3.

**Tabel 3. Matriks konfusi prediksi naïve bayes**

	<i>unacc</i>	<i>acc</i>	<i>vgood</i>	<i>good</i>
<i>unacc</i>	235	26	0	0
<i>acc</i>	7	46	4	12
<i>vgood</i>	0	0	9	1
<i>good</i>	0	5	0	1

**Pengujian model *decision tree***

Model hasil pelatihan menggunakan *decision tree* dengan tiga kriteria tersebut kemudian dilakukan pengujian. Pengujian pada model *decision tree* dengan tiga kriteria tersebut memberikan tingkat pengenalan atau akurasi yang berbeda, yaitu 91,91% pada kriteria *gain ratio*, 90,17% pada kriteria *information gain* dan 91,62% pada kriteria *gini index*, seperti tersaji pada Gambar 4.



Gambar 4. Akurasi *decision tree*

Dari hasil pengujian tersebut dipilih *decision tree* yang mempunyai tingkat pengenalan atau akurasi yang tertinggi, yaitu *decision tree* yang menggunakan kriteria *gain ratio*. Model *decision tree* tersebut mampu mengenali atau memprediksi dengan tepat kelas kategori *unacc* sebesar 234 data atau 95,90% dan 10 data diprediksi sebagai *acc*. Kelas kategori *acc* yang mampu dikenali 63 data atau 87,50%, 6 data terprediksi sebagai *unacc*, 3 data terprediksi sebagai *good*. Kelas kategori *vgood* yang mampu dikenali dengan tepat sebanyak 13 data atau 65%, 3 data terprediksi sebagai *good*. Kelas kategori *good* yang dikenali dengan tepat sebanyak 8 data atau 80% dan 2 data terprediksi sebagai *unacc*. Hasil tersebut dipetakan dalam matriks konfusi yang disajikan pada Tabel 6.

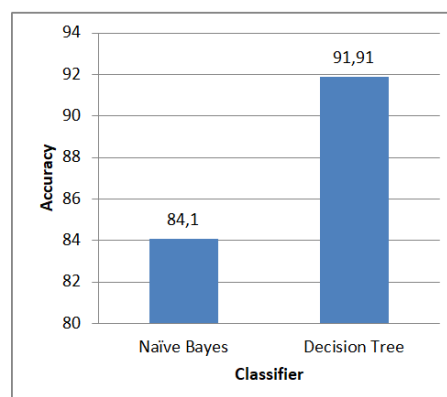
Tabel 6. Matriks konfusi prediksi *decision tree*

TRUE	<i>unacc</i>	<i>acc</i>	<i>vgood</i>	<i>good</i>
<i>unacc</i>	234	10	0	0
<i>acc</i>	6	63	0	3
<i>vgood</i>	0	4	13	3
<i>good</i>	2	0	0	8

### Analisis model naïve bayes dan decision tree

Prediksi data *car evaluation* menggunakan dua algoritma machine learning, yaitu naïve bayes dan decision tree classifier dilakukan menggunakan dataset yang sama, dengan proporsi pembagian data training dan data testing yang sama dengan pembagian secara acak.

Kedua algoritma *classifier* tersebut menghasilkan model dan kinerja yang berbeda. Algoritma *naïve bayes* menghasilkan model berupa nilai probabilitas dan menghasilkan kinerja dengan tingkat ketepatan atau akurasi sebesar 84,1%. *Decision tree* menghasilkan model berupa pohon keputusan dan menghasilkan kinerja dengan tingkat ketepatan atau akurasi yang lebih tinggi yaitu sebesar 91,91%. Perbandingan akurasi dapat dilihat pada Gambar 5. Berdasarkan nilai akurasi dapat dilihat bahwa performa *decision tree* lebih baik daripada naïve bayes pada data *car evaluation* ini.



Gambar 5. Perbandingan akurasi *naïve bayes* dan *decision tree*

## KESIMPULAN

Prediksi terhadap variabel-variabel yang dipertimbangkan dalam memilih suatu mobil berdasarkan *buying*, *maint*, faktor kenyamanan yang meliputi *doors*, *persons*, *lug\_boot* dan *safety* menggunakan algoritma *naïve bayes* dan *decision tree* menghasilkan model yang berbeda dan menunjukkan kemampuan pengenalan atau ketepatan prediksi yang berbeda pula. Algoritma *naïve bayes* yang menghasilkan model *learning* berupa nilai probabilitas memberikan akurasi atau ketepatan prediksi sebesar 84,1%, sedangkan algoritma *decision tree* yang menghasilkan model *learning* berupa pohon keputusan memberikan akurasi atau ketepatan prediksi sebesar 91,91%. Berdasarkan ketepatan prediksi pada data *car evaluation* tersebut, algoritma *decision tree* lebih tepat digunakan daripada algoritma *naïve bayes* karena *decision tree* mempunyai ketepatan pengenalan yang lebih tinggi atau lebih baik.

Untuk mendapatkan ketepatan pengenalan yang lebih tinggi daripada algoritma *decision tree* yang telah dilakukan tuning dengan empat criteria ini, perlu dicoba untuk menggunakan algoritma *random foresh*. Algoritma *random foresh* kemungkinan akan memberikan ketepatan pengenalan yang lebih tinggi, karena dengan prinsip kerjanya yang membagi data training menjadi beberapa bagian dan menghasilkan banyak pohon keputusan tersebut akan memberikan nilai voting yang lebih tinggi untuk akurasinya.

## REFERENSI

- [1] D. K. Utami, W. A. Kusuma, and A. Buono, “Klasifikasi Metagenom dengan Metode Naïve Bayes Classifier Metagenome Classification Using Naïve Bayes Classifier Method,” vol. 3.
- [2] A. Wahyuningtyas and I. S. Sitanggang, “Deteksi Spam pada Twitter Menggunakan Algoritme Naïve Bayes Spam Detection on Twitter using Naïve Bayes Algorithm,” vol. 7, no. Kemenkominfo 2013, pp. 31–40.
- [3] I. P. Astuti, I. Hermadi, A. Buono, and K. H. Mutaqin, “Penyakit Kedelai Dengan Pendekatan Naïve Bayes,” vol. 14, no. 2, 2005.
- [4] H. Naparin, “Klasifikasi Peminatan Siswa Sma Menggunakan,” vol. 02, no. 01, pp. 25–32, 2016.
- [5] Q. S. Setiawan, “Comparison of Naive Bayes and Decision Tree for Classifying Hepatocellular Carcinoma ( HCC ),” pp. 1–5, 2020.
- [6] D. Istiawan, L. Khikmah, S. M. Semarang, A. Info, C. Land, and D. Mining, “Implementation of C4 . 5 Algorithm for Critical Land Prediction in Agricultural Cultivation Areas in Pemali Jratun Watershed,” vol. 2, no. 2, 2019.
- [7] X. Wang, C. Zhou, X. Wang, C. Zhou, and X. Xu, “ScienceDirect ScienceDirect Application of C4 . 5 decision tree for scholarship evaluations Application of C4 . 5 decision tree for scholarship evaluations,” vol. 00, no. 2018, 2019, doi: 10.1016/j.procs.2019.04.027.
- [8] C. Series, “Comparison of C4 . 5 algorithm with naive Bayesian method in classification of Diabetes Mellitus ( A case study at Hasanuddin University hospital Makassar ) Comparison of C4 . 5 algorithm with naive Bayesian method in classification of Diabetes Mellitus ( A case study at Hasanuddin University hospital Makassar ),” 2019, doi: 10.1088/1742-6596/1341/9/092009.
- [9] K. Pinaryanto, R. A. Nugroho, and Y. Basilius, “Classification of Toddler Nutrition Using C4 . 5 Decision Tree Method,” vol. 3, no. 1, pp. 131–142, 2021.
- [10] A. A. Aldino and H. Sulistiani, “Decision Tree C4 . 5 Algorithm For Tuition Aid Grant Program Classification ( Case Study: Department Of Information System , Universitas Teknokrat Indonesia ),” vol. 7, no. 1, pp. 40–50, 2020.
- [11] F. Gorunescu, *Data Mining : Concepts, Models and Techniques*, 12th ed. Springer, 2011.