

Komparasi Algoritma Support Vector Machine dan Naïve Bayes untuk Analisa Sentimen Komentar Aplikasi

Dosen Pembimbing : Desi Ramayanti, S.Kom., MT.



UNIVERSITAS
MERCU BUANA

Disusun Oleh :

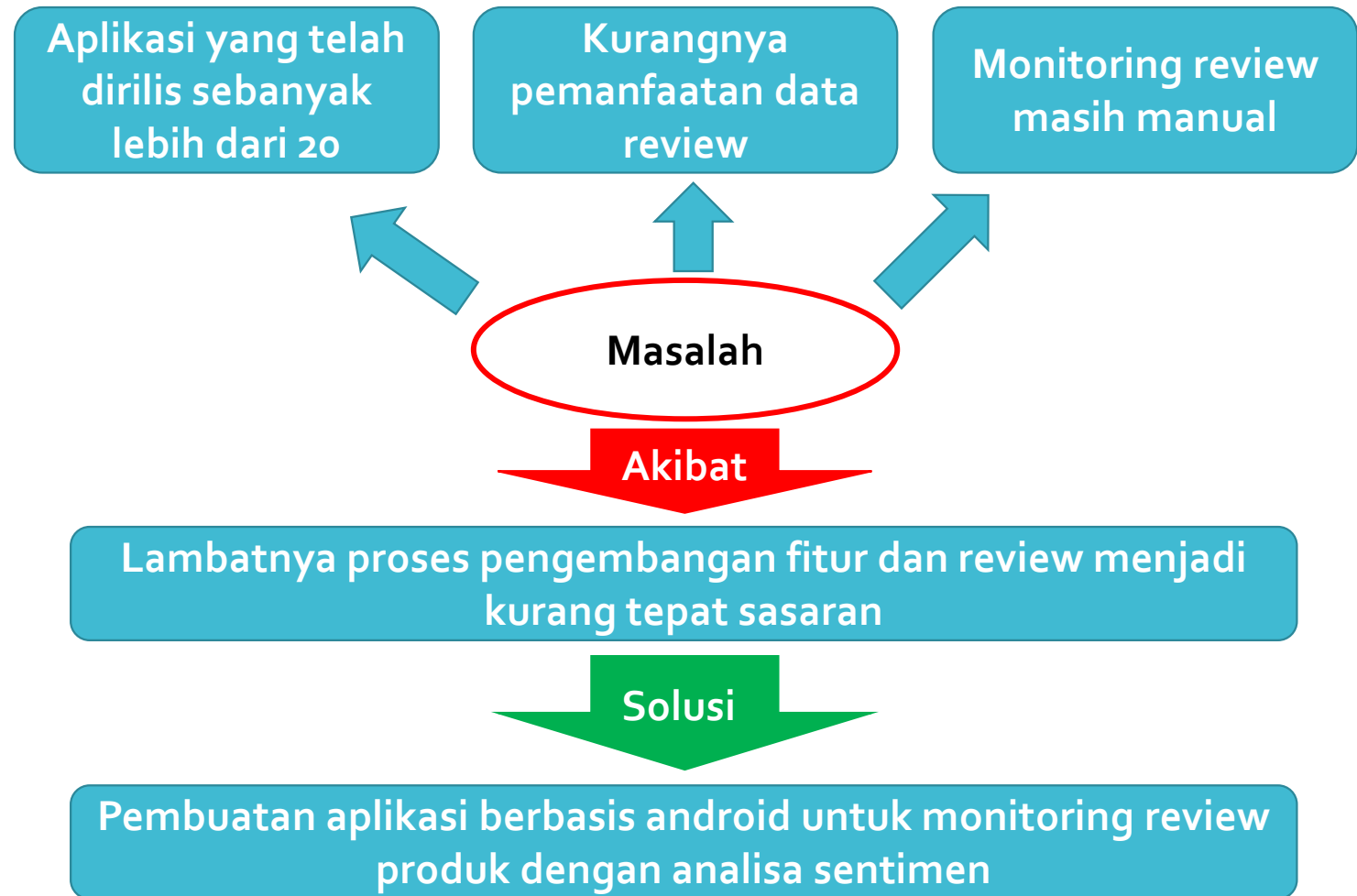
Jody

41517010040

Agenda



Latar Belakang



Rumusan Masalah

1

Bagaimana cara PT Sinergi Digital Teknologi dapat melihat review pengguna ?

2

Bagaimana cara menentukan algoritma deteksi emosi terbaik dari Support Vector Machine dan Naïve Bayes ?

3

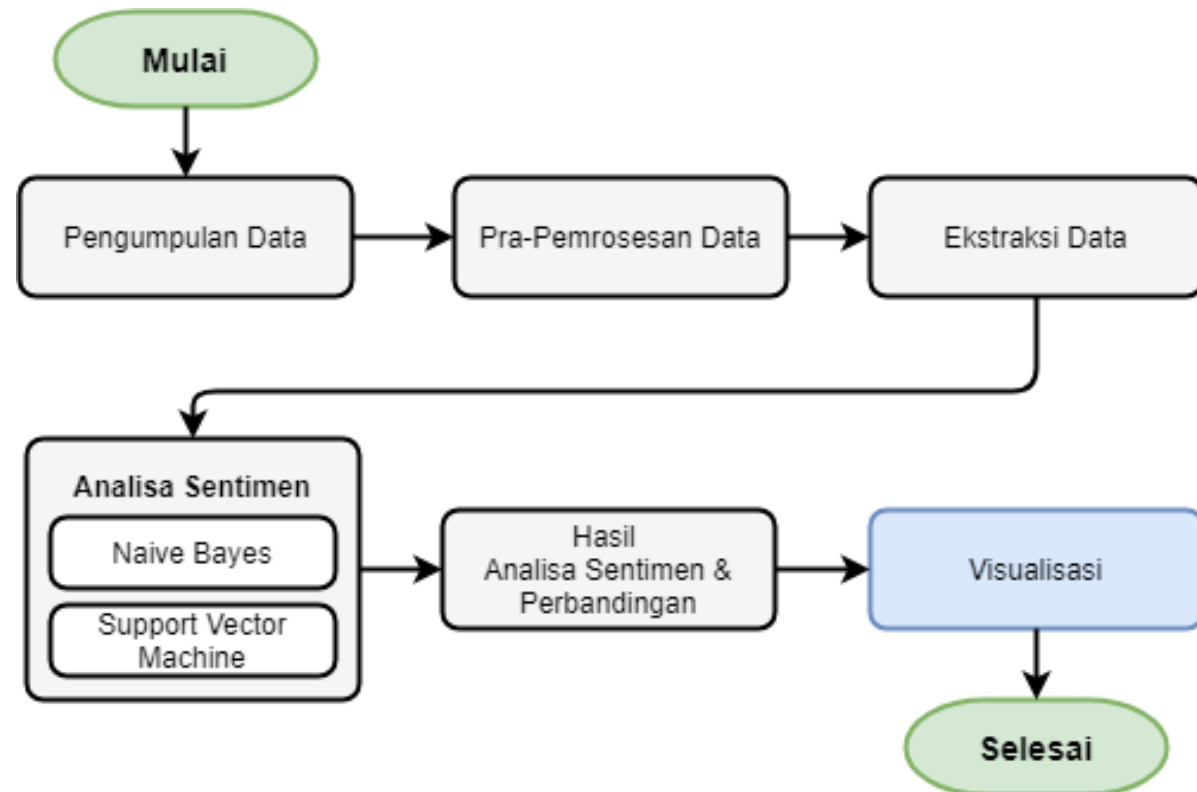
Bagaimana mengimplementasikan algoritma tersebut untuk melakukan deteksi emosi ?

4

Bagaimana membuat sebuah aplikasi monitoring review produk ?

Metodologi

Tahapan Penelitian



Tahap Pengumpulan Data

Data dari API Google Play dan App Store

Atribut	Keterangan
Username	Username pengguna yang memberikan komentar
User image	Link gambar profil pengguna
Tanggal review	Tanggal diberikannya komentar
Rating	Nilai yang diberikan pengguna dengan nilai terendah 1 dan tertinggi 5
Text	Isi komentar yang diberikan pengguna

Tahap Pengolahan Data Awal (1)

- **Case Folding**

Tahap mengubah kata yang memiliki huruf kapital menjadi huruf kecil

Potongan Source Code Case Folding

```
text = review.lower()
```

Tabel Case Folding

Teks	Hasil
Saya mau Tanya tentang transferan ke Bank lain yg belum sampaiz .Dan sampai skrng gak ada solusi.	saya mau tanya tentang transferan ke bank lain yg belum sampaiz .dan sampai skrng gak ada solusi.

Tahap Pengolahan Data Awal (2)

- Penghapusan Angka Dan Tanda Baca

Potongan *Source Code* Penghapusan Tanda Baca

```
tokenizer = RegexTokenizer(r'\w+')  
text = tokenizer.tokenize(text)  
text = " ".join(text)
```

Tabel Penghapusan Tanda Baca

Teks	Hasil
saya mau tanya tentang transferan ke bank lain yg belum sampai2 .dan sampai skrng gak ada solusi.	saya mau tanya tentang transferan ke bank lain yg belum sampai2 dan sampai skrng gak ada solusi

Tahap Pengolahan Data Awal (2.5)

Potongan *Source Code* Penghapusan Angka

```
text = re.sub(r"\d+", "", text)
```

Tabel Penghapusan Angka

Teks	Hasil
saya mau tanya tentang transferan ke bank lain yg belum sampai ² dan sampai skrng gak ada solusi	saya mau tanya tentang transferan ke bank lain yg belum sampai dan sampai skrng gak ada solusi

Tahap Pengolahan Data Awal (3)

- **Tokenisasi**

Tahap pemotongan teks komentar berdasarkan setiap kata yang menyusunnya.

Potongan *Source Code* Tokenisasi

```
text = text.split()
```

Tabel Tokenisasi

Teks	Hasil
saya mau tanya tentang transferan ke bank lain yg belum sampai dan sampai skrng gak ada solusi	['saya', 'mau', 'tanya', 'tentang', 'transferan', 'ke', 'bank', 'lain', 'yg', 'belum', 'sampai', 'dan', 'sampai', 'skrng', 'gak', 'ada', 'solusi']

Tahap Pengolahan Data Awal (4)

- **Normalisasi**

Tahap memperbaiki kata pada teks komentar karena menggunakan bahasa masa kini.

Potongan *Source Code* Normalisasi

```
reformed = [appos[word] if word in appos else word for  
             word in text]  
reformed = " ".join(reformed)
```

Tabel Normalisasi

Teks	Hasil
['saya', 'mau', 'tanya', 'tentang', 'transferan', 'ke', 'bank', 'lain', 'yg', 'belum', 'sampai', 'dan', 'sampai', 'skrng', 'gak', 'ada', 'solusi']	saya mau tanya tentang transferan ke bank lain yang belum sampai dan sampai sekarang tidak ada solusi

Tahap Pengolahan Data Awal (5)

- **Stemming**

Tahap mengubah kata pada teks komentar ke dalam bahasa bakunya.

Potongan *Source Code Stemming*

```
text = stemmer.stem(reformed)
```

Tabel *Stemming*

Teks	Hasil
saya mau tanya tentang transferan ke bank lain yang belum sampai dan sampai sekarang tidak ada solusi	saya mau tanya tentang transfer ke bank lain yang belum sampai dan sampai sekarang tidak ada solusi

Tahap Pengolahan Data Awal (6)

- **Stopword Removal**

Tahap menghapus kata henti pada teks komentar.

Potongan *Source Code* Stopword Removal

```
text = str.remove(text)
```

Tabel *Stopword Removal*

Teks	Hasil
saya mau tanya tentang transfer ke bank lain yang belum sampai dan sampai sekarang tidak ada solusi	mau tentang ke lain belum dan sekarang tidak ada solusi

Ekstraksi Data

Tahap ini mengubah teks komentar pengguna menjadi sebuah vektor agar mudah untuk dibaca dan dihitung oleh sistem. Ekstraksi Data ini menggunakan metode TF-IDF. Perhitungan TF-IDF menggunakan persamaan berikut.

$$tf_{t,d} = \frac{f_{t,d}}{n_d}$$

$$idf_t = \log \frac{N}{df_t}$$

$$W_{t,d} = tf_{t,d} \cdot idf_{t,d}$$

Potongan *Source Code* Ekstraksi Data

```
vectorizer = TfidfVectorizer(min_df=3, max_df=0.3, use_idf=True)
train_vectors = vectorizer.fit_transform(preprocessData1)
test_vectors = vectorizer.transform(preprocessData2)
```

Analisa Sentimen (1)

- **Algoritma Naïve Bayes**

Teorema Bayes ini sendiri merupakan teorema yang lebih fokus pada konsep probabilitas bersyarat berdasarkan pengalaman masa lalu. Secara umum teorema dari Bayes dirumuskan dalam persamaan berikut.

$$P(A|B) = \frac{P(A).P(B|A)}{P(B)}$$

Potongan *Source Code* Algoritma Naïve Bayes

```
# model algoritma multinominal naive bayes
Naive = naive_bayes.MultinomialNB()
# training data
Naive.fit(train_vectors, Y_train)
# testing data
predictions_NB = Naive.predict(test_vectors)
```

Analisa Sentimen (2)

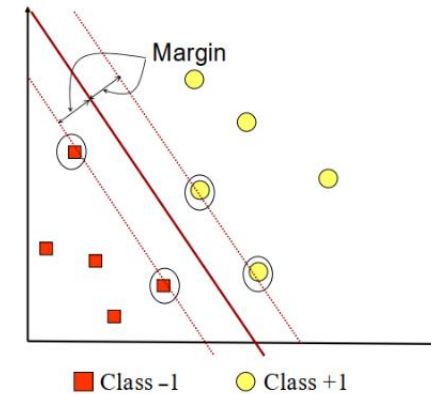
- **Algoritma Support Vector Machine**

Sedangkan SVM merupakan metode memprediksi dalam pengklasifikasian dan regresi. Dengan mencari *hyperplane* untuk memaksimalkan margin antar kelas data. *Hyperplane* berguna untuk memisahkan 2 kelompok kelas yaitu -1 dan +1. Persamaan Algoritma SVM terdapat pada persamaan berikut.

$$\vec{w} \cdot \vec{x} + b = 0 \text{ (Hyperplane)}$$

$$\vec{w} \cdot \vec{x} + b < 0 \text{ (Kelas -1)}$$

$$\vec{w} \cdot \vec{x} + b \geq 0 \text{ (Kelas +1)}$$



Potongan *Source Code* Algoritma Support Vector Machine

```
# model algoritma support vector machine
SVM = svm.SVC(kernel='linear')
# training data
SVM.fit(train_vectors, Y_train)
# testing data
predictions_SVM = SVM.predict(test_vectors)
```


Hasil Analisa Sentimen dan Perbandingan (1)

- Hasil Pengujian Algoritma Support Vector Machine

Jumlah Data Latih	Jumlah Data Uji	Sentimen	Precision SVM (%)	Recall SVM (%)	F1-Score SVM (%)	Accuracy SVM (%)
800	200	Pos	100	100	100	100
		Neg	100	100	100	
700	300	Pos	97	93	95	99
		Neg	99	100	99	
600	400	Pos	92	85	89	97.75
		Neg	98	99	99	
500	500	Pos	93	80	86	97.40
		Neg	98	99	99	

Hasil Analisa Sentimen dan Perbandingan (2)

- Hasil Pengujian Algoritma Naïve Bayes

Jumlah Data Latih	Jumlah Data Uji	Sentimen	Precision Naïve Bayes (%)	Recall Naïve Bayes (%)	F1-Score Naïve Bayes (%)	Accuracy Naïve Bayes (%)
800	200	Pos	100	67	80	97
		Neg	97	100	98	
700	300	Pos	100	67	80	97
		Neg	96	100	98	
600	400	Pos	93	63	75	95.75
		Neg	96	99	98	
500	500	Pos	97	55	70	95.20
		Neg	95	100	97	

Hasil Analisa Sentimen dan Perbandingan (2)

- Perbandingan Algoritma Support Vector Machine dan Naïve Bayes

Support Vector Machine memperoleh hasil :

- Akurasi rata-rata = 98.53%
- Rata-rata *precision* = 97.12%
- Rata-rata *Recall* = 94.5%
- Rata-rata *f1-score* = 95.87%
- Waktu eksekusi algoritma = 0.0015 second

Naïve Bayes memperoleh hasil :

- Akurasi rata-rata = 96.23%
- Rata-rata *precision* = 96.75%
- Rata-rata *Recall* = 81.37%
- Rata-rata *f1-score* = 87%
- Waktu eksekusi algoritma = 0.002 second








Kesimpulan : Algoritma Support Vector Machine lebih baik dari segi akurasi, *Precision*, *Recall*, *f1-score* dan waktu eksekusinya dibandingkan dengan Algoritma Naïve Bayes

Hasil Analisa Sentimen dari Sisi Aplikasi (1)

```
"values": [
  {
    "aplikasi": "Lenna Virtual Assistant",
    "id": 1620,
    "link_review": "https://play.google.com/store/apps/details?id=ai.lenna.assistant&reviewId=gp:A0qpTOHHtwrU1fZ_Sfb-vSo8U9oVthPN46ubwiUcmcQ_Y4xycGjWcig_LDCwh6IAhNAoiT1cQR564scd_th9cKk",
    "post": "Mantap",
    "rating": "5",
    "sentiment": 1,
    "sumber": "Google Play",
    "tanggal_post": "2020-10-13",
    "user_avatar": "https://play-lh.googleusercontent.com/a-/A0h14Gj0Iq6jd3Zoi9zW9IUoZrG-x3wJPRPvjH3FQHN9",
    "user_name": "Arfan Ian"
  },
  {
    "aplikasi": "Lenna Virtual Assistant",
    "id": 1621,
    "link_review": "https://play.google.com/store/apps/details?id=ai.lenna.assistant&reviewId=gp:A0qpTOHMQ7p-moX9pmod-MDQx_ZielK6hRsr3QxKDV95ddITF3KTH06OHGBIRoFdwOwoy3XmntbWK-0",
    "post": "ini apk sdh ga di urus ...sdh tdk ada petugas IT... Buang2 kuota... login sdh benar..di bilang salah..",
    "rating": "1",
    "sentiment": 0,
    "sumber": "Google Play",
    "tanggal_post": "2020-09-05",
    "user_avatar": "https://play-lh.googleusercontent.com/a-/A0h14GhXrFjIkrSZR0mQ1Tewu0GLU_qZH5p3oa4w1cY",
    "user_name": "FAMILY dlai24"
  }
],
```

Hasil Komentar dan Prediksi Sentimen dari Response API

Hasil Analisa Sentimen dari Sisi Aplikasi (1)

Account	Post	Date	Source	Rating	Sentiment
 YOHTAMA Yogi Wahyu Pratama	Apps yang bisa membantu dalam membuat asuransi semua barang berharga supaya aman dan hati tenang	27 Jun 2020	Google Play	5	positive
 asrul purnama	Bagus	23 Jun 2020	Google Play	5	positive
 Edgina Triana Dewi	Aplikasinya sangat mudah dimengerti dan sangat membantu	15 Jun 2020	Google Play	5	positive
 Ahmad Muzaqi	Mantul	14 Jun 2020	Google Play	5	positive
 dewi aulia	Mantappp sekali aplikasinya, sekarang sdah tidak susah lgi cari asuransi yg twpat. Semuanya lengkap d sni	12 Jun 2020	Google Play	5	positive
 dwi Sugesti	Memudahkan kita dalam berasuransi	09 Jun 2020	Google Play	5	positive
 fedora tehnik	Siiip...mantap..maju teruuus tripa	08 Jun 2020	Google Play	5	positive

Hasil Komentar dan Prediksi Sentimen dari Website Lenna Analytic