

BAB III

ANALISIS DAN PERANCANGAN

3.1. Arsitektur Sistem

Arsitektur sistem pada penelitian analisis sentimen ini terdiri dari serangkaian tahapan terstruktur yang dimulai dari pengumpulan data ulasan pengguna *Shopee* hingga proses evaluasi hasil klasifikasi. Sistem diawali dengan tahap input data berupa komentar mentah yang diperoleh dari *Google Play Store*, kemudian data tersebut masuk ke modul *Preprocessing* yang mencakup *Cleaning*, *case folding*, *Stopword removal*, *Stemming*, dan *Tokenizing* untuk memastikan teks berada dalam bentuk yang siap dianalisis. Setelah itu, data yang telah diproses diarahkan ke tahap pembobotan menggunakan metode *TF-IDF* untuk mengubah kata menjadi representasi numerik yang dapat dipahami oleh algoritma. Selanjutnya, fitur yang telah terbentuk dimasukkan ke dalam modul klasifikasi menggunakan algoritma *Naïve Bayes* untuk menentukan kategori sentimen dari setiap ulasan.

3.2. Langkah-Langkah Pengolahan Data

Pada tahapan pengolahan data ini, penulis melakukan pengolahan data dengan system manual. Tahapan utamanya yaitu menentukan data training dan untuk datanya yaitu sebagai berikut.

3.2.1. Pengumpulan Data

Pengumpulan data dilakukan dengan mengambil ulasan pengguna aplikasi *Shopee* yang tersedia pada *Google Play Store*, di mana setiap komentar diunduh untuk dijadikan bahan analisis sentimen. Data yang diambil berupa teks ulasan yang mencerminkan pengalaman, keluhan, maupun kepuasan pengguna terhadap

layanan aplikasi. Jumlah data disesuaikan dengan kebutuhan penelitian sehingga dapat mewakili variasi sentimen yang muncul. Proses ini menjadi tahap awal yang sangat penting karena kualitas data yang dikumpulkan akan berpengaruh langsung pada hasil analisis dan kinerja model yang digunakan.

Tabel 3. 1. Pengumpulan Data

| userName | content | score |
|---------------------------|--------------------------------------|--------------|
| Diego Sembiring | gagal terus | 1 |
| Sisilia Kioza | aplikasi membantu bisnis | 5 |
| Asep Hermawan | sangat puas belanja di <i>Shopee</i> | 5 |
| Muhammad Rachmat Ramdhani | lebih simpel | 1 |
| muhammad syarifuddin | sesuai pesanan | 3 |
| Maysarohh | good | 4 |
| Suhadak Suhadak | puass belanja d shopy | 5 |
| Made Suartana | jos | 5 |
| Asep Rohmat | ok | 5 |
| muhamad bunyamin | <i>Shopee</i> ok! | 5 |
| Triwidoto Widoto | sangat membantu | 5 |

Tabel di atas merupakan hasil pengumpulan data melalui proses *scraping* ulasan pengguna aplikasi *Shopee* dari *Google Play Store* menggunakan *Google Colab*. Data yang diperoleh berjumlah 30 ulasan tetapi yang dimasukkan hanya 12, masing-masing berisi informasi seperti username, isi komentar, rating, waktu ulasan, serta atribut pendukung lainnya. Proses *scraping* ini memungkinkan peneliti mendapatkan data secara otomatis dan efisien tanpa perlu mengambil ulasan secara manual.

3.2.2. Pre-processing Data

Pre-processing data pada penelitian ini dilakukan untuk membersihkan ulasan yang tidak layak digunakan, sehingga hanya teks yang relevan dan

berkualitas yang masuk ke tahap analisis. Proses ini mencakup penghapusan karakter yang tidak diperlukan seperti angka, emoji, simbol, tanda baca berlebihan, serta kata-kata yang tidak membawa makna penting. Selain itu, dilakukan juga normalisasi teks seperti *case folding*, *Stopword removal*, dan *Stemming* agar data menjadi lebih terstruktur dan konsisten. Dengan melakukan *Pre-processing* secara menyeluruh, ulasan yang digunakan menjadi lebih bersih dan siap diolah pada tahap selanjutnya untuk analisis sentimen.

3.2.3. Natural Language Processing (NLP)

Natural Language Processing (NLP) adalah cabang ilmu kecerdasan buatan yang berfokus pada bagaimana komputer dapat memahami, memproses, dan menganalisis bahasa manusia. Melalui NLP, teks dapat diubah menjadi bentuk yang dapat dibaca mesin sehingga memungkinkan komputer melakukan berbagai tugas seperti analisis sentimen, klasifikasi teks, penerjemahan, dan ekstraksi informasi.

1. Data Dasar Penelitian

Data dasar penelitian ini terdiri dari enam ulasan pengguna yang telah diberi label sentimen positif, netral, dan negatif berdasarkan skor serta isi komentar. Data tersebut digunakan sebagai dasar analisis sentimen dan perhitungan probabilitas awal pada metode *Naive Bayes*.

Tabel 3. 2. Data Dasar Penelitian

| No | Komentar | Skor | Label Sentimen |
|----|--------------------------------------|------|----------------|
| 1 | gagal terus | 1 | Negatif |
| 2 | aplikasi membantu bisnis | 5 | Positif |
| 3 | sangat puas belanja di <i>Shopee</i> | 5 | Positif |
| 4 | sesuai pesanan | 3 | Netral |
| 5 | good | 4 | Positif |
| 6 | puass belanja d shopy | 5 | Positif |

Tabel tersebut menunjukkan data dasar penelitian yang terdiri dari enam ulasan pengguna yang telah diberi label sentimen, yaitu positif, netral, dan negatif, berdasarkan skor penilaian dan isi komentar. Data ini digunakan sebagai dataset awal dalam analisis sentimen untuk menggambarkan persepsi pengguna.

2. Tahap *Cleaning*

Tahap *Cleaning* bertujuan untuk membersihkan data teks dari elemen non-teks seperti angka, simbol, tanda baca, dan karakter khusus yang tidak berpengaruh terhadap analisis sentimen. Proses ini dilakukan agar data teks menjadi lebih terstruktur dan siap digunakan pada tahapan pengolahan selanjutnya.

Tabel 3. 3. Tahap *Cleaning*

| Teks Awal | Teks Bersih |
|--------------------------------------|--------------------------------------|
| gagal terus | gagal terus |
| aplikasi membantu bisnis | aplikasi membantu bisnis |
| sangat puas belanja di <i>Shopee</i> | sangat puas belanja di <i>Shopee</i> |
| sesuai pesanan | sesuai pesanan |
| good | good |
| puass belanja d shopy | puass belanja d shopy |

Tabel tersebut menunjukkan hasil tahap *Cleaning*, di mana seluruh teks awal tidak mengalami perubahan karena data ulasan sudah terbebas dari elemen non-teks seperti simbol, angka, dan tanda baca. Hal ini menandakan bahwa data telah berada dalam kondisi bersih dan siap untuk dilanjutkan ke tahap *Preprocessing* berikutnya dalam analisis sentimen.

3. *Case folding*

Tahap *case folding* dilakukan dengan mengubah seluruh teks menjadi huruf kecil untuk menyeragamkan penulisan kata dan mencegah perbedaan makna akibat

penggunaan huruf kapital. Proses ini bertujuan agar setiap kata diperlakukan sama pada saat analisis dan perhitungan selanjutnya.

Tabel 3. 4. Case folding

| Teks Awal | Hasil Case folding |
|--------------------------------------|--------------------------------------|
| gagal terus | gagal terus |
| aplikasi membantu bisnis | aplikasi membantu bisnis |
| sangat puas belanja di <i>Shopee</i> | sangat puas belanja di <i>Shopee</i> |
| sesuai pesanan | sesuai pesanan |
| good | good |
| puass belanja d shopy | puass belanja d shopy |

Tabel hasil *case folding* menunjukkan bahwa seluruh teks ulasan tetap sama karena sejak awal sudah menggunakan huruf kecil. Hal ini menandakan bahwa data telah seragam secara penulisan sehingga siap digunakan pada tahap pengolahan teks berikutnya tanpa memunculkan perbedaan makna akibat variasi huruf kapital.

4. Tokenizing

Tokenizing merupakan proses memecah teks menjadi unit-unit terkecil berupa kata atau token agar dapat dianalisis secara terpisah. Tahap ini bertujuan untuk memudahkan pengolahan data teks dalam proses perhitungan dan analisis sentimen selanjutnya.

Tabel 3. 5. Tokenizing

| Komentar | Token Hasil Tokenizing |
|--------------------------------------|---|
| gagal terus | [gagal, terus] |
| aplikasi membantu bisnis | [aplikasi, membantu, bisnis] |
| sangat puas belanja di <i>Shopee</i> | [sangat, puas, belanja, di, <i>Shopee</i>] |
| sesuai pesanan | [sesuai, pesanan] |
| good | [good] |
| puass belanja d shopy | [puass, belanja, d, shopy] |

Tabel tersebut menunjukkan hasil tahap *Tokenizing*, di mana setiap komentar dipecah menjadi kata-kata penyusunnya agar dapat dianalisis secara lebih rinci.

Melalui proses ini, setiap kata diperlakukan sebagai satuan data yang berdiri sendiri sehingga memudahkan tahap pengolahan teks selanjutnya dalam analisis sentimen.

5. *Stopword removal*

Stopword removal merupakan tahap untuk menghapus kata-kata umum yang tidak memiliki pengaruh signifikan terhadap penentuan sentimen, seperti kata penghubung dan kata depan. Tahap ini bertujuan untuk meningkatkan kualitas fitur dengan mempertahankan kata-kata yang lebih informatif dan relevan.

Tabel 3. 6. *Stopword removal*

| Token Awal | Setelah <i>Stopword removal</i> |
|---|---|
| [gagal, terus] | [gagal, terus] |
| [aplikasi, membantu, bisnis] | [aplikasi, membantu, bisnis] |
| [sangat, puas, belanja, di, <i>Shopee</i>] | [sangat, puas, belanja, <i>Shopee</i>] |
| [sesuai, pesanan] | [sesuai, pesanan] |
| [good] | [good] |
| [puass, belanja, d, shopy] | [puass, belanja, shopy] |

Tabel tersebut menunjukkan hasil tahap *Stopword removal*, di mana kata-kata umum yang tidak memiliki makna penting dalam analisis sentimen, seperti “di” dan “d”, berhasil dihilangkan dari data. Proses ini membuat token yang dihasilkan menjadi lebih fokus pada kata-kata bermakna sehingga dapat meningkatkan akurasi pada tahap analisis selanjutnya.

6. *Stemming*

Stemming merupakan proses mengubah kata ke bentuk dasarnya untuk menyatukan berbagai variasi kata yang memiliki makna sama. Tahap ini bertujuan untuk meningkatkan konsistensi data teks sehingga analisis sentimen dapat dilakukan dengan lebih akurat.

Tabel 3. 7. Stemming

| Komentar Awal | Token Setelah Stemming |
|-----------------------------------|---|
| gagal terus | [gagal, terus] |
| aplikasi membantu bisnis | [aplikasi, bantu, bisnis] |
| sangat puas belanja <i>Shopee</i> | [sangat, puas, belanja, <i>Shopee</i>] |
| sesuai pesanan | [sesuai, pesan] |
| good | [good] |
| puass belanja d shopy | [puas, belanja, <i>Shopee</i>] |

Tabel tersebut menunjukkan hasil tahap *Stemming*, di mana kata-kata pada setiap komentar telah diubah ke bentuk dasarnya, seperti kata “membantu” menjadi “bantu”, “pesanan” menjadi “pesan”, serta perbaikan kata tidak baku seperti “puass” menjadi “puas” dan “shopy” menjadi “*Shopee*”. Proses ini bertujuan untuk menyeragamkan bentuk kata sehingga makna yang sama tidak dihitung sebagai kata berbeda pada tahap analisis selanjutnya.

7. Penyusunan Vocabulary

Penyusunan *Vocabulary* dilakukan dengan mengumpulkan seluruh kata unik yang muncul setelah tahap *Preprocessing*, termasuk *Cleaning*, *case folding*, *Tokenizing*, *Stopword removal*, dan *Stemming*. *Vocabulary* ini berfungsi sebagai daftar fitur yang akan digunakan dalam proses analisis sentimen, sehingga setiap kata dapat dihitung frekuensinya dan dianalisis kontribusinya terhadap masing-masing kelas sentimen.

Tabel 3. 8. Penyusunan Vocabulary

| No | Kata |
|-----------|-------------|
| 1 | gagal |
| 2 | terus |
| 3 | aplikasi |
| 4 | bantu |
| 5 | bisnis |
| 6 | sangat |

| | |
|---------------|---------------|
| 7 | puas |
| 8 | belanja |
| 9 | <i>Shopee</i> |
| 10 | sesuai |
| 11 | pesan |
| 12 | good |
| Jumlah | 12 |

Tabel penyusunan *Vocabulary* menunjukkan seluruh kata unik yang diperoleh setelah melalui tahapan *Preprocessing*, mulai dari *Cleaning* hingga *Stemming*. *Vocabulary* ini digunakan sebagai dasar dalam proses analisis sentimen untuk menghitung frekuensi kemunculan kata dan menentukan kontribusi setiap kata terhadap masing-masing kelas sentimen dalam metode *Naive Bayes*.

8. Term Frequency (TF) Berbasis Kelas

Term Frequency (TF) berbasis kelas merupakan metode perhitungan frekuensi kemunculan kata yang diakumulasikan berdasarkan masing-masing kelas sentimen, yaitu positif, netral, dan negatif. Pendekatan ini digunakan untuk mengetahui seberapa kuat suatu kata merepresentasikan suatu kelas sentimen dalam proses klasifikasi.

Tabel 3. 9. Term Frequency (TF) Berbasis Kelas

| Kata | D1 | D2 | D3 | D4 | D5 | D6 |
|---------------|-----------|-----------|-----------|-----------|-----------|-----------|
| gagal | 1 | 0 | 0 | 0 | 0 | 0 |
| terus | 1 | 0 | 0 | 0 | 0 | 0 |
| aplikasi | 0 | 1 | 0 | 0 | 0 | 0 |
| bantu | 0 | 1 | 0 | 0 | 0 | 0 |
| bisnis | 0 | 1 | 0 | 0 | 0 | 0 |
| sangat | 0 | 0 | 1 | 0 | 0 | 0 |
| puas | 0 | 0 | 1 | 0 | 0 | 1 |
| belanja | 0 | 0 | 1 | 0 | 0 | 1 |
| <i>Shopee</i> | 0 | 0 | 1 | 0 | 0 | 1 |
| sesuai | 0 | 0 | 0 | 1 | 0 | 0 |

| | | | | | | |
|-------|---|---|---|---|---|---|
| pesan | 0 | 0 | 0 | 1 | 0 | 0 |
| good | 0 | 0 | 0 | 0 | 1 | 0 |

Tabel *Term Frequency (TF)* berbasis kelas menunjukkan frekuensi kemunculan setiap kata yang diakumulasikan berdasarkan kelas sentimen positif, netral, dan negatif. Dari tabel tersebut terlihat bahwa kata-kata seperti puas, belanja, dan *Shopee* memiliki frekuensi tertinggi pada kelas positif, sedangkan kata gagal dan terus hanya muncul pada kelas negatif, serta kata sesuai dan pesan muncul pada kelas netral, sehingga masing-masing kata dapat merepresentasikan karakteristik dari kelas sentimen tertentu dalam proses klasifikasi.

9. Normalisasi *TF-IDF*

Normalisasi *TF-IDF* dilakukan untuk menyeragamkan skala nilai bobot setiap dokumen sehingga perbedaan panjang dokumen tidak memengaruhi hasil analisis. Dengan normalisasi, vektor *TF-IDF* menjadi lebih seimbang dan siap digunakan dalam proses klasifikasi atau pengukuran kemiripan dokumen.

Tabel 3. 10. Normalisasi *TF-IDF*

| Kata | D1 | D2 | D3 | D4 | D5 | D6 |
|---------------|-----|---------|------|-----|----|---------|
| gagal | 0.5 | 0 | 0 | 0 | 0 | 0 |
| terus | 0.5 | 0 | 0 | 0 | 0 | 0 |
| aplikasi | 0 | 0.33333 | 0 | 0 | 0 | 0 |
| bantu | 0 | 0.33333 | 0 | 0 | 0 | 0 |
| bisnis | 0 | 0.33333 | 0 | 0 | 0 | 0 |
| sangat | 0 | 0 | 0.25 | 0 | 0 | 0 |
| puas | 0 | 0 | 0.25 | 0 | 0 | 0.33333 |
| belanja | 0 | 0 | 0.25 | 0 | 0 | 0.33333 |
| <i>Shopee</i> | 0 | 0 | 0.25 | 0 | 0 | 0.33333 |
| sesuai | 0 | 0 | 0 | 0.5 | 0 | 0 |
| pesan | 0 | 0 | 0 | 0.5 | 0 | 0 |
| good | 0 | 0 | 0 | 0 | 1 | 0 |

Tabel di atas menunjukkan hasil normalisasi nilai *TF-IDF* pada setiap dokumen ulasan, di mana bobot kata telah disesuaikan agar berada pada skala yang seragam. Normalisasi menyebabkan total bobot pada setiap dokumen menjadi seimbang, sehingga kata-kata dalam dokumen yang lebih panjang tidak mendominasi perhitungan dibandingkan dokumen yang lebih pendek. Terlihat bahwa setiap dokumen memiliki distribusi bobot yang proporsional, misalnya pada dokumen D1 kata *gagal* dan *terus* masing-masing memiliki bobot 0,5, sedangkan pada dokumen D3 bobot kata terbagi rata sebesar 0,25 untuk setiap kata. Hasil normalisasi ini memastikan bahwa representasi vektor dokumen menjadi lebih adil dan stabil, sehingga dapat meningkatkan kinerja proses klasifikasi sentimen pada tahap selanjutnya menggunakan algoritma *Naïve Bayes*.

10. DF (*Document Frequency*)

Document Frequency (DF) adalah ukuran yang menunjukkan jumlah dokumen dalam suatu kumpulan data yang mengandung sebuah kata tertentu. Nilai DF digunakan untuk mengetahui seberapa umum atau jarangny suatu kata muncul di seluruh dokumen dan menjadi dasar dalam perhitungan *Inverse Document Frequency (IDF)*.

Tabel 3. 11. DF (*Document Frequency*)

| Kata | DF |
|---------------|----|
| gagal | 1 |
| terus | 1 |
| aplikasi | 1 |
| bantu | 1 |
| bisnis | 1 |
| sangat | 1 |
| puas | 2 |
| belanja | 2 |
| <i>Shopee</i> | 2 |

| | |
|--------|---|
| sesuai | 1 |
| pesan | 1 |
| good | 1 |

Berdasarkan tabel *Document Frequency (DF)* di atas, dapat diketahui bahwa sebagian besar kata hanya muncul pada satu dokumen, seperti gagal, terus, aplikasi, bantu, bisnis, sangat, sesuai, pesan, dan good, sehingga kata-kata tersebut bersifat spesifik dan memiliki tingkat keunikan yang tinggi. Sementara itu, kata puas, belanja, dan *Shopee* memiliki nilai DF sebesar 2, yang menunjukkan bahwa kata tersebut muncul pada lebih dari satu dokumen sehingga bersifat lebih umum dibandingkan kata lainnya. Perbedaan nilai DF ini menjadi dasar dalam perhitungan *IDF*, di mana kata dengan DF rendah akan memiliki bobot yang lebih besar karena dianggap lebih informatif dalam membedakan sentimen pada ulasan pengguna.

11. Hitung *IDF*

Tabel 3. 12. Data *IDF*

| <i>IDF</i> |
|------------|
| 0.77815 |
| 0.77815 |
| 0.77815 |
| 0.77815 |
| 0.77815 |
| 0.77815 |
| 0.47712 |
| 0.47712 |
| 0.47712 |
| 0.77815 |
| 0.77815 |
| 0.77815 |

Berdasarkan hasil perhitungan *Inverse Document Frequency (IDF)* di atas, terlihat bahwa sebagian besar kata memiliki nilai *IDF* sebesar 0,77815, yang menunjukkan bahwa kata-kata tersebut hanya muncul pada satu dokumen sehingga bersifat lebih spesifik dan memiliki daya pembeda yang lebih tinggi. Sementara itu, kata-kata yang memiliki nilai *IDF* lebih rendah yaitu 0,47712 muncul pada lebih dari satu dokumen, sehingga dianggap lebih umum dan memiliki tingkat keunikan yang lebih rendah. Perbedaan nilai *IDF* ini menunjukkan bahwa kata dengan frekuensi kemunculan yang jarang akan memperoleh bobot yang lebih besar dalam pembobotan *TF-IDF*, sedangkan kata yang sering muncul pada beberapa dokumen akan memperoleh bobot yang lebih kecil karena kontribusi informasinya relatif lebih rendah dalam proses analisis sentimen.

12. Hitung *TF-IDF*

Tabel 3. 13. Hitung *TF-IDF*

| Kata | <i>TF-IDF</i> | | | | | | Jumlah |
|----------|---------------|---------|---------|---------|---------|---------|---------|
| | D1 | D2 | D3 | D4 | D5 | D6 | |
| gagal | 0.38908 | 0 | 0 | 0 | 0 | 0 | 0.38908 |
| terus | 0.38908 | 0 | 0 | 0 | 0 | 0 | 0.38908 |
| aplikasi | 0 | 0.25938 | 0 | 0 | 0 | 0 | 0.25938 |
| bantu | 0 | 0.25938 | 0 | 0 | 0 | 0 | 0.25938 |
| bisnis | 0 | 0.25938 | 0 | 0 | 0 | 0 | 0.25938 |
| sangat | 0 | 0 | 0.19454 | 0 | 0 | 0 | 0.19454 |
| puas | 0 | 0 | 0.11928 | 0 | 0 | 0.15904 | 0.27832 |
| belanja | 0 | 0 | 0.11928 | 0 | 0 | 0.15904 | 0.27832 |
| Shopee | 0 | 0 | 0.11928 | 0 | 0 | 0.15904 | 0.27832 |
| sesuai | 0 | 0 | 0 | 0.38908 | 0 | 0 | 0.38908 |
| pesan | 0 | 0 | 0 | 0.38908 | 0 | 0 | 0.38908 |
| good | 0 | 0 | 0 | 0 | 0.77815 | 0 | 0.77815 |

Tabel di atas menunjukkan hasil akhir pembobotan *TF-IDF* pada setiap dokumen ulasan setelah melalui proses normalisasi, di mana setiap nilai

mencerminkan tingkat kepentingan suatu kata dalam dokumen tertentu. Terlihat bahwa kata *gagal* dan *terus* memiliki bobot *TF-IDF* sebesar 0,38908 pada dokumen D1 yang menunjukkan karakteristik sentimen negatif, sedangkan kata *aplikasi*, *bantu*, dan *bisnis* memiliki bobot 0,25938 pada D2 yang merepresentasikan ulasan positif. Pada dokumen D3 dan D6, kata *puas*, *belanja*, dan *Shopee* memiliki bobot yang relatif lebih kecil karena kata tersebut muncul pada lebih dari satu dokumen sehingga nilai *IDF*-nya lebih rendah. Sementara itu, kata *good* memiliki bobot *TF-IDF* tertinggi yaitu 0,77815 pada D5 karena hanya muncul pada satu dokumen dan memiliki tingkat keunikan yang tinggi. Hasil pembobotan ini menunjukkan bahwa kata dengan bobot *TF-IDF* besar memiliki peran penting dalam merepresentasikan isi dokumen dan menjadi fitur utama dalam proses klasifikasi sentimen menggunakan algoritma *Naïve Bayes*.

13. Hitung *Prior Probability Naïve Bayes*

Prior Probability merupakan probabilitas awal suatu kelas sentimen yang dihitung berdasarkan proporsi jumlah data pada setiap kelas terhadap total keseluruhan data. Nilai ini menggambarkan kecenderungan awal data sebelum mempertimbangkan pengaruh kata-kata pada proses klasifikasi menggunakan metode *Naive Bayes*.

$$P(c) = \frac{N_c}{N}$$

Keterangan:

- a) $P(c)$: *Prior Probability* dari kelas sentimen c
- b) N_c : jumlah dokumen yang termasuk ke dalam kelas sentiment c
- c) N : jumlah seluruh dokumen pada data penelitian

Perhitungan *Prior Probability*:

a) Kelas Positif:

$$P(\text{Positif}) = \frac{4}{6} = 0,667$$

b) Kelas Netral:

$$P(\text{Netral}) = \frac{1}{6} = 0,167$$

c) Kelas Negatif:

$$P(\text{Negatif}) = \frac{1}{6} = 0,167$$

Hasil *Prior*

| Kelas | Prior |
|--------------|--------------|
| Positif | 0.667 |
| Netral | 0.167 |
| Negatif | 0.167 |

Berdasarkan hasil perhitungan *Prior Probability* pada tabel di atas, dapat diketahui bahwa kelas positif memiliki nilai *prior* tertinggi sebesar 0,667 dibandingkan dengan kelas netral dan negatif yang masing-masing bernilai 0,167. Hal ini menunjukkan bahwa sebelum mempertimbangkan kemunculan kata-kata dalam proses klasifikasi, data penelitian cenderung didominasi oleh sentimen positif, sehingga peluang awal suatu data termasuk ke dalam kelas positif lebih besar dibandingkan kelas lainnya.

14. Likelihood dengan Laplace Smoothing

Likelihood dengan *Laplace Smoothing* digunakan untuk menghitung probabilitas kemunculan suatu kata pada kelas sentimen tertentu dengan

menambahkan nilai konstanta agar tidak terjadi probabilitas nol. Metode ini bertujuan untuk menjaga kestabilan perhitungan dan meningkatkan keandalan hasil klasifikasi pada metode *Naive Bayes*.

Tabel 3. 14. Variabel Gagal

| Variabel | Partisi | Positif | Netral | Negatif | P(Positif) | P(Neutral) | P(Negatif) |
|----------|-----------|---------|--------|---------|------------|------------|--------------|
| Gagal | Ada | 0 | 0 | 1 | $0/6 = 0$ | $0/6 = 0$ | $1/6 = 0,16$ |
| | Tidak Ada | 0 | 0 | 0 | $0/6 = 0$ | $0/6 = 0$ | $0/6 = 0$ |
| | Total | 0 | 0 | 1 | 100% | 100% | 100% |

Tabel 3. 15. Variabel Sesuai

| Variabel | Partisi | Positif | Netral | Negatif | P(Positif) | P(Neutral) | P(Negatif) |
|----------|-----------|---------|--------|---------|------------|--------------|------------|
| Sesuai | Ada | 0 | 1 | 0 | $0/6 = 0$ | $1/6 = 0,16$ | $0/6 = 0$ |
| | Tidak Ada | 0 | 0 | 0 | $0/6 = 0$ | $0/6 = 0$ | $0/6 = 0$ |
| | Total | 0 | 1 | 0 | 100% | 100% | 100% |

Tabel 3. 16. Variabel Good

| Variabel | Partisi | Positif | Netral | Negatif | P(Positif) | P(Neutral) | P(Negatif) |
|----------|-----------|---------|--------|---------|--------------|------------|------------|
| Good | Ada | 1 | 0 | 0 | $1/6 = 0,16$ | $0/6 = 0$ | $0/6 = 0$ |
| | Tidak Ada | 0 | 0 | 0 | $0/6 = 0$ | $0/6 = 0$ | $0/6 = 0$ |
| | Total | 1 | 0 | 0 | 100% | 100% | 100% |

15. Data Testing

Tahapan terakhir pada proses klasifikasi menggunakan algoritma *Naive Bayes* adalah perhitungan *Posterior probability*, yaitu tahap di mana seluruh nilai probabilitas awal (*prior*) dan peluang kemunculan fitur kata pada masing-masing kelas (*Likelihood*) digabungkan untuk menghasilkan probabilitas akhir suatu dokumen termasuk pada kelas tertentu. Perhitungan ini mengikuti prinsip Bayes yang menyatakan bahwa kelas terbaik adalah kelas dengan nilai posterior paling tinggi, sehingga setiap komentar yang telah melalui proses *Preprocessing* dan pembobotan akan dievaluasi probabilitasnya terhadap kelas Positif, Netral, dan Negatif. Pada tahap ini, setiap kata dalam komentar diuji kontribusinya terhadap

masing-masing kelas, kemudian nilai posterior tersebut dibandingkan sehingga diperoleh keputusan klasifikasi. Dengan demikian, proses *posterior probability* menjadi titik akhir yang menentukan hasil sentimen, karena nilai terbesar menjadi acuan penetapan kategori ulasan pengguna terhadap aplikasi, apakah bersifat positif, netral, atau negative.

Tabel 3. 17. Data Testing

| Komentar Awal | Token Setelah <i>Stemming</i> |
|-----------------------------------|---|
| gagal terus | [gagal, terus] |
| aplikasi membantu bisnis | [aplikasi, bantu, bisnis] |
| sangat puas belanja <i>Shopee</i> | [sangat, puas, belanja, <i>Shopee</i>] |
| sesuai pesanan | [sesuai, pesan] |
| good | [good] |
| puass belanja d shopy | [puas, belanja, <i>Shopee</i>] |

Setelah diperoleh untuk data testing, kemudian dilakukan *Cleaning* untuk membersihkan teks yang dibutuhkan ataupun symbol.

Tabel 3. 18. Data *Preprocessing*

| Komentar Awal | Token Setelah <i>Stemming</i> |
|-----------------------------------|---|
| gagal terus | [gagal, terus] |
| aplikasi membantu bisnis | [aplikasi, bantu, bisnis] |
| sangat puas belanja <i>Shopee</i> | [sangat, puas, belanja, <i>Shopee</i>] |
| sesuai pesanan | [sesuai, pesan] |
| good | [good] |
| puass belanja d shopy | [puas, belanja, <i>Shopee</i>] |

Setelah data di *Cleaning*, kemudian masuk pada proses perhitungan data. Untuk proses ini penulis akan mengambil data nomor 1. Untuk proses perhitungannya yaitu sebagai berikut.

- 1. Perhitungan pertama pada kalimat “gagal terus”, yang akan dihitung yaitu pada kata ”gagal”.**

$$\begin{aligned}
 P(\text{Uji}|\text{Positif}) &= P(\text{Positif}) \times P(\text{gagal}|\text{Positif}) \\
 &= 0,667 \times 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Uji}|\text{Netral}) &= P(\text{Netral}) \times P(\text{gagal}|\text{Netral}) \\
 &= 0,167 \times 0 \\
 &= 0
 \end{aligned}$$

$$\begin{aligned}
 P(\text{Uji}|\text{Negatif}) &= P(\text{Negatif}) \times P(\text{gagal}|\text{Negatif}) \\
 &= 0,167 \times 0,16 \\
 &= 0,02
 \end{aligned}$$

Dari hasil perhitungan data uji, maka diperoleh hasil bahwa dari 3 nilai label yang ada bahwa nilai negatif yang paling tinggi, maka hasil uji yaitu data tersebut masuk pada kategori negatif.

| Komentar Awal | Token Setelah <i>Stemming</i> | Token Setelah <i>Stemming</i> |
|-----------------------------------|---|-------------------------------|
| gagal terus | [gagal, terus] | Negatif |
| aplikasi membantu bisnis | [aplikasi, bantu, bisnis] | Positif |
| sangat puas belanja <i>Shopee</i> | [sangat, puas, belanja, <i>Shopee</i>] | Positif |
| sesuai pesanan | [sesuai, pesan] | Netral |
| good | [good] | Positif |
| puass belanja d shopy | [puas, belanja, <i>Shopee</i>] | Positif |

16. Confusion Matrix

| | | Predicted | | | Σ |
|----------|---------|-----------|--------|---------|----------|
| | | Positif | Netral | Negatif | |
| Actual | Positif | 4 | 0 | 0 | 4 |
| | Netral | 0 | 1 | 0 | 1 |
| | Negatif | 0 | 0 | 1 | 1 |
| Σ | | 4 | 1 | 1 | 6 |

TP : Nilai sebenarnya dan nilai prediksi harus sama. Jadi untuk kelas ini, nilai sel 1 adalah nilai TP.

FN : Jumlah nilai kolom yang bersesuaian kecuali nilai TP

$$\begin{aligned} \text{FN} &= \text{sel 2} + \text{sel 3} \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

FP : Jumlah nilai baris terkait kecuali nilai TP

$$\begin{aligned} \text{FP} &= \text{sel 4} + \text{sel 7} \\ &= 0 + 0 \\ &= 0 \end{aligned}$$

TN : Jumlah nilai semua kolom dan baris kecuali nilai kelas yang kita hitung nilainya.

$$\begin{aligned} \text{TN} &= \text{sel 5} + \text{sel 6} + \text{sel 8} + \text{sel 9} \\ &= 1 + 0 + 0 + 1 \\ &= 2 \end{aligned}$$

Perhitungan Akurasi, Presisi, dan Recall:

$$\text{Accuracy} = \frac{4+2}{4+2+0+0} \times 100\% \quad \text{Then the Accuracy value} = 100\%$$

$$\text{Presisi} = \frac{4}{4+0} \times 100\% \quad \text{Then the Precision value} = 100\%$$

$$\text{Recall} = \frac{4}{4+0} \times 100\% \quad \text{Then the Recall value} = 100\%$$

Berdasarkan hasil evaluasi menggunakan *Confusion Matrix* di atas, dapat diketahui bahwa model klasifikasi mampu mengklasifikasikan data sentimen dengan sangat baik. Seluruh data pada masing-masing kelas (positif, netral, dan

negatif) berhasil diprediksi dengan benar tanpa adanya kesalahan klasifikasi, yang ditunjukkan oleh nilai *False Positive* (FP) dan *False Negative* (FN) yang bernilai nol. Hal ini menghasilkan nilai akurasi, presisi, dan recall sebesar 100%, yang berarti model memiliki tingkat ketepatan prediksi yang sempurna pada data uji yang digunakan. Dengan demikian, metode yang diterapkan dinilai sangat efektif dalam mengklasifikasikan sentimen pada dataset tersebut, meskipun hasil ini juga perlu diperhatikan secara kritis karena kemungkinan dipengaruhi oleh jumlah data yang terbatas.