

BAB 4

HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini merepresentasikan pengalaman pengguna aplikasi BTN Mobile melalui pendekatan analisis sentimen dan klasifikasi berdasarkan lima dimensi *HEART framework*, yaitu *happiness*, *engagement*, *adoption*, *retention*, dan *task success*. Data ulasan dikumpulkan menggunakan metode *web scraping* dengan bantuan *package google-play-scraper*, menghasilkan 13.333 ulasan dari total 23.897 yang tersedia. Setelah dilakukan penghapusan duplikasi dan validasi manual, sebanyak 10.973 ulasan dinyatakan valid untuk dianalisis. Seluruh data kemudian melalui proses *preprocessing text* yang terdiri atas lima tahap, yaitu *cleaning*, *normalisasi*, *tokenizing*, *filtering*, dan *stemming*, guna memastikan data dalam kondisi optimal untuk dianalisis lebih lanjut.

Penentuan label terhadap ulasan dilakukan berdasarkan sistem rating, di mana ulasan dengan rating 4 dan 5 dikategorikan sebagai positif, sementara rating 1 hingga 3 dikategorikan sebagai negatif. Dari 8.000 data yang diberi label, diperoleh 3.851 ulasan positif dan 4.149 ulasan negatif. Proses klasifikasi sentimen menggunakan algoritma SVM dengan representasi TF-IDF dan pembagian data 80% untuk pelatihan dan 20% untuk pengujian. Hasilnya, model menunjukkan akurasi dan *f1-score* sebesar 94%, mengindikasikan performa klasifikasi yang sangat baik dan seimbang. Untuk menggali lebih dalam pengalaman pengguna, dilakukan klasifikasi dimensi HEART menggunakan pendekatan *cosine similarity*. Sebanyak 2.500 ulasan diberi label secara manual sebagai data latih dengan distribusi seimbang pada lima dimensi. Model ini mencapai akurasi dan *f1-score* rata-rata sebesar 88%, dengan dimensi *happiness* mencatatkan performa tertinggi dalam hal *recall* dan *f1-score* sedangkan dimensi *engagement* memiliki nilai *recall* dan *f1-score* yang sedikit lebih rendah dibandingkan dimensi lainnya.

Distribusi hasil klasifikasi HEART terhadap 10.973 ulasan menunjukkan bahwa *happiness* merupakan dimensi dengan jumlah sentimen positif terbanyak

dengan 1.549 ulasan, mencerminkan kepuasan pengguna. Sementara itu, *Task Success* memiliki jumlah sentimen negatif terbanyak dengan dengan 1.647 ulasan, menunjukkan banyaknya kendala teknis yang dialami pengguna. Hasil ini menegaskan bahwa meskipun aplikasi telah memberikan pengalaman yang baik dalam aspek kenyamanan, seperti tampilan yang menarik, namun masih terdapat hambatan signifikan pada beberapa fitur utama pada aplikasi seperti hambatan dalam transfer dana dan pembayaran tagihan. Oleh karena itu, diperlukan peningkatan keandalan sistem untuk mendukung keberhasilan tugas pengguna sehingga dapat meningkatkan kualitas pengalaman pengguna di masa mendatang secara lebih optimal.

4.2 PENGUMPULAN DATA

Berdasarkan hasil pengumpulan data yang dilakukan menggunakan metode *web scraping* ulasan dengan *package google-play-scraper*, diperoleh sebanyak 13.333 ulasan pengguna aplikasi *BTN Mobile* dari Google Play Store. Proses pengambilan data dilakukan pada tanggal 19 April 2025 menggunakan platform *google colab* dan bahasa pemrograman *python*. Berikut Kode Program 4.1 untuk menjalankan proses *web scraping*.

Kode Program 4.1 Scraping Ulasan

```
1. from google_play_scraper import reviews, Sort
2. app_package = "id.co.btn.mobilebanking.android"
3. ulasan, _ = reviews(
4.     app_package,
5.     lang='id',
6.     country='id',
7.     count=23897,
8.     sort=Sort.NEWEST
9. )
```

Parameter yang digunakan dalam Kode Program 4.1 salah satunya adalah 'sort=Sort.NEWEST'. Parameter ini digunakan untuk mengatur agar data ulasan yang diambil merupakan ulasan terbaru yang ditulis oleh pengguna. Dengan memilih opsi ini, peneliti dapat memperoleh informasi terkini mengenai pengalaman dan pendapat pengguna terhadap aplikasi, sehingga analisis yang

dilakukan menjadi lebih relevan dengan kondisi aplikasi saat ini. Hasil dari proses *web scraping* dapat dilihat pada Gambar 4.1.

	reviewId	content	score	datetime
0	068a4e0f-d50c-4474-b8a0-ccf1a1ca2b0	serba praktis pokoknya	5	2025-04-19 14:46:58
1	c7978808-404f-4319-987a-da00b7c2035	wow BTN bekin dari gua 320 ign jadi maling k...	1	2025-04-19 14:35:38
2	c0ae5765-0999-4589-b187-cd04ec278d53	kenapa pengajuan pinjaman di aplikasi tidak da...	5	2025-04-19 11:59:17
3	#940d37a-a82f-443c-ad33-f6e6e7d1ap462	semangat update bala malah jd makin ribet aku	1	2025-04-19 07:25:36
4	7f2b58bc-d4eb-4b65-8e08-b71480cbe9b	Aplikasi BTN mobile bagus. Iu saja sudah cuku...	5	2025-04-19 03:53:03
...				
13328	35d6a792-28b9-4858-041e-b3bc7c4d5391	Lebih friendly dan mudah dipakai	5	2023-01-31 10:28:25
13329	189fdd1-a527-4cbe-8f1e-bac167a78a8	Keren	5	2023-01-31 10:18:54
13330	ecc01535-1cda-4e42-9483-1b8757661e47	Aplikasinya keren	5	2023-01-31 09:12:41
13331	323c9bc1-f180-4fe4-9289-5327f7ebd310	LEBIH MUDAH MURAH DAN FLEKSIBEL	5	2023-01-31 08:15:41
13332	62a6b0el-ecb2-4e71-84de-dbbe34bd9624	Lebih nyaman dan tampilan okey	5	2023-01-31 07:45:21

Gambar 4.1 Hasil Proses *Web Scraping*

Setelah Gambar 4.1 disimpan, selanjutnya dilakukan proses penghapusan data duplikasi baik berupa ulasan dengan isi yang identik maupun pengulangan dari akun yang sama untuk memastikan keunikan setiap data ulasan. Hasil dari proses ini menyisakan 12.060 ulasan. Setelah itu, dilakukan pemeriksaan manual terhadap setiap ulasan untuk mengevaluasi kesesuaian antara isi komentar dan rating yang diberikan oleh pengguna, seperti mengeliminasi ulasan yang positif namun memiliki rating rendah atau sebaliknya. Proses verifikasi ini bertujuan untuk memastikan validitas dan konsistensi data sebelum dilakukan analisis lebih lanjut. Setelah tahap verifikasi, diperoleh sebanyak 10.973 ulasan yang dianggap layak dan siap digunakan dalam tahap preprocessing. Proses pengumpulan data dirangkum secara sistematis dalam Tabel 4.1 di bawah ini:

Tabel 4.1 Proses Pengumpulan Data Ulasan

Tahap	Deskripsi	Jumlah Ulasan
Data awal di Google Play Store	Jumlah ulasan aplikasi BTN <i>Mobile</i> yang ditampilkan secara publik	23.897
Pengambilan data	Ulasan yang berhasil diambil menggunakan package <i>google-play-scraper</i>	13.333
Penghapusan duplikat	Penghapusan ulasan identik dan pengulangan dari akun yang sama	12.060
Pemeriksaan manual	Menyesuaikan antara isi ulasan dan rating serta menghapus data tidak valid	10.973

Tabel 4.1 menunjukkan alur dari proses pengumpulan data ulasan sehingga dapat menghasilkan data ulasan yang siap digunakan dalam tahap berikutnya. Hasil dari pengumpulan data ulasan dapat dilihat pada Gambar 4.2 berikut:

	content	score
0	serba praktis pokoknya.	5
1	woy BTN balikin duit gua 320 jgn jadi maling k...	1
2	sementak update bale malah jd makin ribet. aku...	1
3	Aplikasi BTN mobile bagus. itu saja sudah cuku...	5
4	BTN sangat tidak aman sekali jikalau transaksi...	1
...	...	...
10968	Cakep nih M Bankingnya Semua jadi lebih mudah	5
10969	saya kasih bintang 3 dulu, soalnya pas mau log...	3
10970	Lebih friendly dan mudah dipakai. . . Maju le...	5
10971	LEBIH MUDAH, MURAH DAN FLEKSIBEL	5
10972	Lebih nyaman dan tampilan okey	5

10973 rows x 2 columns

Gambar 4.2 Hasil Pengumpulan Data

4.3 PREPROCESSING TEXT

Preprocessing text dilakukan untuk membersihkan dan menyiapkan data teks agar dapat diolah dengan lebih efektif pada tahapan analisis selanjutnya. Alur proses ini dilakukan secara bertahap diantaranya sebagai berikut:

4.3.1 Cleaning

Tahapan awal yang dilakukan pada *preprocessing text* yaitu proses membersihkan data teks dari kesalahan yang terdiri dari mengubah format penulisan menjadi lowercase atau huruf kecil, kemudian menghapus karakter seperti simbol tertentu, angka dan juga tanda baca. Berikut Kode Program 4.2 untuk menjalankan proses *cleaning*.

Kode Program 4.2 Cleaning Text

```

1. def clean_text(text):
2.     text = text.lower()
3.     text = re.sub(r'http\S+|www\S+|https\S+', '', text)
4.     text = re.sub(r'@\w+|#\w+', '', text)
5.     text = re.sub(r'^\x00-\x7F+', '', text)
6.     text = re.sub(r'\d+', '', text)
7.     text = text.translate(str.maketrans('', '', string.punctuation))
8.     text = re.sub(r'\s+', ' ', text)
9.     return text.strip()

```

Berdasarkan Kode Program 4.2 tahapan *cleaning* yang dilakukan dalam *function* tersebut yaitu:

- `text.lower()` : mengubah seluruh huruf dalam teks menjadi huruf kecil.
- `re.sub(r'http\S+|www\S+|https\S+', '', text)` : menghapus URL atau tautan.
- `re.sub(r'@\w+|#\w+', '', text)` : menghapus mention (@) dan hashtag (#).
- `re.sub(r'[^\x00-\x7F]+' , '', text)` : menghapus emoji atau karakter non-ASCII.
- `text = re.sub(r'\d+', '', text)` : menghapus angka.
- `text = text.translate(str.maketrans('', '', string.punctuation))` : menghapus tanda baca.
- `text = re.sub(r'\s+', ' ', text)` : menghapus spasi ganda.
- `return text.strip()` : menghapus spasi di awal dan akhir teks.

Hasil dari tahapan *cleaning text* dapat dilihat pada Gambar 4.3 berikut:

	content	score	cleaned_content
10777	Keren, fitur layanannya semakin terbaik 🌟	5	keren fitur layanannya semakin terbaik
7410	Sering eror, kadang GX bisa di buka	1	sering eror kadang gx bisa di buka
1920	Dari sekian banyak nya pilihan aplikasi mobile...	5	dari sekian banyak nya pilihan aplikasi mobile...
9155	Transfer gagal tapi saldo langsung kepotong ...	2	transfer gagal tapi saldo langsung kepotong bu...
8100	Suka banget sama aplikasi BTN Mobile ini karen...	5	suka banget sama aplikasi btn mobile ini karen...

Gambar 4.3 Sampel Hasil *Cleaning Text*

4.3.2 Normalisasi

Tahapan *normalisasi* merupakan salah satu tahapan penting dalam *preprocessing text* yang bertujuan untuk menyamakan kata agar lebih konsisten dan dapat dikenali oleh sistem dalam tahap analisis selanjutnya. Proses ini penting karena dalam data ulasan pengguna sering ditemukan kata-kata yang tidak baku, seperti singkatan, bahasa informal, atau ejaan yang menyimpang dari kaidah Bahasa Indonesia. Proses normalisasi ini dilakukan menggunakan pendekatan berbasis kamus (*dictionary-based normalization*), di mana setiap kata yang tidak baku akan dicocokkan dengan daftar kata baku yang telah disusun oleh penulis dalam bentuk kamus normalisasi. Kamus normalisasi tersebut dapat dilihat pada **Lampiran 1**. dan Kode Program 4.3 yang digunakan untuk menjalankan proses *normalisasi* dapat dilihat sebagai berikut:

Kode Program 4.3 Normalisasi Text

```
1. def normalisasi_text(text):
2.     words = text.split()
3.     return ' '.join([kamus_normalisasi.get(word, word) for word
                        in words])
```

Proses dari Kode Program 4.3 tersebut dimulai dengan memecah teks menjadi kata-kata menggunakan metode ‘split()’. Setiap kata kemudian dibandingkan dengan daftar kamus normalisasi; jika kata tersebut merupakan kata tidak baku, maka akan digantikan dengan kata bakunya sesuai kamus yang telah disusun. Kamus normalisasi ini dibuat secara manual berdasarkan pengamatan terhadap pola penulisan tidak baku yang sering muncul dalam ulasan pengguna. Fungsi ‘get()’ pada kamus digunakan untuk mencari padanan baku dari kata tidak baku, dengan parameter kedua sebagai nilai default yang akan mengembalikan kata asli jika tidak ditemukan dalam kamus. Hasil dari tahapan *normalisasi text* dapat dilihat pada Gambar 4.4 berikut:

	cleaned_content	normalized_content
10777	keren fitur layanannya semakin terbaik	keren fitur layanannya semakin terbaik
7410	sering eror kadang gx bisa di buka	sering eror kadang tidak bisa di buka
1920	dari sekian banyak nya pilihan aplikasi mobile...	dari sekian banyak pilihan aplikasi mobile me...
9155	transfer gagal tapi saldo langsung kepotong bu...	transfer gagal tapi saldo langsung kepotong bu...
8100	suka banget sama aplikasi btn mobile ini karen...	suka sekali sama aplikasi btn mobile ini karen...

Gambar 4.4 Sampel Hasil *Normalisasi Text*

4.3.3 Tokenizing

Tokenizing merupakan tahapan dalam *preprocessing text* yang bertujuan untuk memecah kalimat atau paragraf menjadi unit-unit kata yang lebih kecil, yang disebut sebagai token. Dalam penelitian ini, proses *tokenizing* dilakukan dengan menggunakan *function* ‘word tokenize()’ dari pustaka NLTK dalam bahasa pemrograman *python*. Fungsi ini mampu menangani pemisahan kata secara lebih akurat dibanding metode pemisahan berdasarkan spasi, karena dapat mengenali tanda baca, kontraksi, dan format kalimat lainnya dalam Bahasa Indonesia maupun

Inggris. Berikut adalah Kode Program 4.4 yang digunakan untuk menjalankan proses *tokenizing*.

Kode Program 4.4 *Tokenizing Text*

```
1. def tokenize_text(text):
2.     return word_tokenize(text)
```

Proses dari Kode Program 4.4 menggunakan *function* ‘word_tokenize()’ bertujuan memecah teks menjadi token-token individual. Fungsi ini tidak hanya memisahkan kata berdasarkan spasi, tetapi juga mampu memisahkan tanda baca dari kata, sehingga menghasilkan pemisahan yang lebih akurat. Tokenisasi ini akan mengembalikan list token berupa kata-kata individual dari input teks agar siap digunakan dalam proses berikutnya. Hasil dari tahapan *tokenizing text* dapat dilihat pada Gambar 4.5 berikut:

	normalized_content	tokenized_content
10777	keren fitur layanannya semakin terbaik	[keren, fitur, layanannya, semakin, terbaik]
7410	sering eror kadang tidak bisa di buka	[sering, eror, kadang, tidak, bisa, di, buka]
1920	dari sekian banyak pilihan aplikasi mobile me...	[dari, sekian, banyak, pilihan, aplikasi, mobi...
9155	transfer gagal tapi saldo langsung kepotong bu...	[transfer, gagal, tapi, saldo, langsung, kepot...
8100	suka sekali sama aplikasi btn mobile ini karen...	[suka, sekali, sama, aplikasi, btn, mobile, in...

Gambar 4.5 Sampel Hasil *Tokenizing Text*

4.3.4 Filtering

Filtering merupakan tahapan *preprocessing text* yang bertujuan untuk menyaring kata-kata yang tidak memiliki makna penting. Kata-kata tersebut umumnya berupa kata hubung, kata depan, kata ganti, dan kata bantu yang sering muncul dalam teks namun tidak memberikan kontribusi bermakna terhadap interpretasi atau klasifikasi data. Dalam penelitian ini, *filtering* dilakukan dengan menggunakan daftar *stopwords* Bahasa Indonesia dari pustaka NLTK. Proses *filtering* mengeliminasi token yang termasuk dalam daftar *stopwords*, sehingga hanya kata-kata penting yang relevan dengan makna kalimat yang dipertahankan untuk proses analisis selanjutnya. Berikut adalah Kode Program 4.5 yang digunakan untuk melakukan *filtering*.

Kode Program 4.5 Filtering Text

```
1. stop_words = set(stopwords.words('indonesian'))
2.   def filter_stopwords(tokens):
3.     return [word for word in tokens if word not in stop_words]
```

Proses Kode Program 4.5 di atas dimulai dengan memuat daftar *stopwords* Bahasa Indonesia ke dalam variabel ‘stop_words’. *Function* ‘filter_stopwords()’ kemudian akan menerima input berupa daftar token dan mengembalikan token yang sudah disaring. Dengan demikian, data teks yang dihasilkan dari tahap ini menjadi lebih ringkas dan relevan untuk dianalisis lebih lanjut. Hasil dari tahapan *filtering text* dapat dilihat pada Gambar 4.6 berikut:

	tokenized_content	filtered_content
10777	[keren, fitur, layanannya, semakin, terbaik]	[keren, fitur, layanannya, terbaik]
7410	[sering, eror, kadang, tidak, bisa, di, buka]	[error, kadang, buka]
1920	[dari, sekian, banyak, pilihan, aplikasi, mobi...	[sekian, pilihan, aplikasi, mobile, membantu, ...
9155	[transfer, gagal, tapi, saldo, langsung, kepot...	[transfer, gagal, saldo, langsung, kepotong, b...
8100	[suka, sekali, sama, aplikasi, btn, mobile, in...	[suka, aplikasi, btn, mobile, keuangan, kontro...

Gambar 4.6 Sampel Hasil *Filtering Text*

4.3.5 Stemming

Tahapan akhir dalam *preprocessing text* pada penelitian ini yaitu proses mengubah kata menjadi bentuk dasarnya dengan cara menghapus imbuhan dan juga akhiran kata. Tujuan *stemming* adalah untuk menyederhanakan teks sehingga kata-kata dapat dianalisis secara konsisten. Dengan demikian, *stemming* membantu dalam menyatukan kata-kata yang memiliki makna serupa namun bentuk berbeda, sehingga meningkatkan efektivitas dalam proses analisis data teks. Dalam penelitian ini, proses *stemming* dilakukan dengan menggunakan pustaka *sastrawi*, yaitu pustaka pemrosesan bahasa alami untuk Bahasa Indonesia yang menyediakan *stemmer* berbasis aturan. Berikut adalah Kode Program 4.6 yang digunakan dalam proses *stemming*.

Kode Program 4.6 Stemming Text

```

1. stemmer = StemmerFactory().create_stemmer()
2. def clean_and_stem(tokens):
3.     if not tokens or not isinstance(tokens, list):
4.         return ""
5.     sentence = ' '.join(tokens)
6.     stemmed_sentence = stemmer.stem(sentence)
7.     return stemmed_sentence

```

Kode Program 4.6 tersebut menggunakan *function* ‘clean_and_stem()’ untuk memeriksa apakah input berupa list token yang valid, lalu menggabungkan token-token tersebut menjadi satu kalimat dan menerapkan *stemming* untuk mengubah kata-kata menjadi bentuk dasarnya dengan menghilangkan imbuhan dan akhiran. Hasil akhir dari proses ini adalah kalimat yang terdiri dari kata-kata dasar yang siap digunakan. Fungsi lain dari tahapan ini yaitu dapat meningkatkan akurasi analisis teks karena mengurangi variasi kata yang memiliki makna sama namun berbeda bentuk akibat penggunaan imbuhan. Hasil dari tahapan *stemming text* dapat dilihat pada Gambar 4.7 berikut:

	filtered_content	stemmed_content
10777	[keren, fitur, layanannya, terbaik]	keren fitur layan baik
7410	[eror, kadang, buka]	eror kadang buka
1920	[sekian, pilihan, aplikasi, mobile, membantu, ...]	sekian pilih aplikasi mobile bantu butuh perso...
9155	[transfer, gagal, saldo, langsung, kepotong, b...]	transfer gagal saldo langsung potong buka noti...
8100	[suka, aplikasi, btn, mobile, keuangan, kontro...]	suka aplikasi btn mobile uang kontrol handphon...

Gambar 4.7 Sampel Hasil *Stemming Text*

4.4 LABELING

Berdasarkan metode *labeling* yang telah ditetapkan, yaitu dengan memberikan label positif pada ulasan dengan rating 4 dan 5, serta negatif pada ulasan dengan rating 1, 2, dan 3 (Sidauruk et al., 2023), dilakukan proses pelabelan terhadap 8.000 data ulasan yang telah melewati tahap *preprocessing text*. Pemanfaatan rating sebagai dasar pelabelan dipilih karena rating dianggap sebagai indikator langsung dari kepuasan atau ketidakpuasan pengguna terhadap aplikasi (Pamungkas & Sulianta, 2024). Pendekatan ini digunakan untuk memastikan bahwa

setiap ulasan memiliki label yang merepresentasikan sentimen pengguna secara objektif berdasarkan penilaian pengguna terhadap aplikasi. Dengan demikian, data yang telah diberi label dapat digunakan sebagai dasar yang kuat dalam pelatihan model analisis sentimen, sehingga diharapkan mampu mengidentifikasi kecenderungan sentimen pada ulasan lain secara lebih akurat. Kode Program 4.7 di bawah ini menunjukkan implementasi proses pelabelan berdasarkan nilai rating.

Kode Program 4.7 *Labeling Text*

```
1. def ulasan_score(score):
2.     if score in [1, 2, 3]:
3.         return 'Negatif'
4.     elif score in [4, 5]:
5.         return 'Positif'
```

Dari Kode Program 4.7 tersebut, diperoleh hasil bahwa sebanyak 3.851 ulasan dikategorikan sebagai ulasan positif, dan 4.149 ulasan dikategorikan sebagai ulasan negatif. Hasil ini menunjukkan bahwa mayoritas pengguna cenderung memberikan ulasan dengan sentimen negatif terhadap aplikasi *BTN Mobile*. Distribusi *labeling* ini akan digunakan dalam tahap pelatihan dan pengujian model klasifikasi sentimen pada penelitian selanjutnya. Sampel hasil dari proses *labeling* dapat dilihat pada Gambar 4.8.

	content	sentimen
0	Keren, fitur layanannya semakin terbaik.. 🙌	Positif
1	Sering error, kadang GX bisa di buka	Negatif
2	Dari sekian banyak nya pilihan aplikasi mobile...	Positif
3	Transfer gagal tapi saldo langsung kepotong ...	Negatif
4	Suka banget sama aplikasi BTN Mobile ini karen...	Positif
...	...	...
7995	Maaf saya mau tanya ..Saya baru buka rekening B...	Negatif
7996	Ko makin ribet	Negatif
7997	Kenapa tidak bisa login padahal sudah daftar? ...	Negatif
7998	Error mulu parah	Negatif
7999	Bpk/Ibu YTH. Sinarmas MSIG Life sbg mitra Bank...	Negatif

8000 rows × 2 columns

Gambar 4.8 Sampel Hasil *Labeling*

4.5 KLASIFIKASI SVM

Setelah tahap pelabelan terhadap 8.000 data ulasan dilakukan, langkah selanjutnya adalah melakukan klasifikasi sentimen menggunakan algoritma SVM. Sebelum masuk ke tahap klasifikasi, seluruh data ulasan yang telah diberi label terlebih dahulu diubah menjadi *representasi numerik* menggunakan teknik TF-IDF. Teknik ini digunakan untuk menghitung bobot setiap kata dalam dokumen berdasarkan frekuensi kemunculannya dan tingkat kekhususan kata tersebut di seluruh dokumen. Data ulasan yang telah diberi label kemudian dibagi menjadi dua bagian, yaitu 80% untuk data *training* dan 20% untuk data *testing*. Berikut Kode Program 4.8 yang menunjukkan *representasi numerik* menggunakan teknik TF-IDF dan pembagian data.

Kode Program 4.8 TF-IDF dan Pembagian data

```
1. # TF-IDF Vectorizer
2. tfidf = TfidfVectorizer()
3. X_tfidf = tfidf.fit_transform(X)

4. # Split data menjadi train dan test
5. X_train, X_test, y_train, y_test = train_test_split(X_tfidf,
    y, test_size=0.2, random_state=42)
```

Objek ‘TfidfVectorizer()’ digunakan untuk mentransformasikan data teks ke dalam bentuk representasi numerik berbasis bobot kata, sehingga setiap ulasan dapat diolah oleh algoritma *machine learning*. Hasil dari transformasi ini disimpan dalam variabel ‘X_tfidf’, yang berisi matriks vektor dari seluruh data teks ulasan. Setelah itu, data yang telah ditransformasi dibagi menjadi dua bagian yaitu data latih ‘X_train, y_train’ dan data uji ‘X_test, y_test’ dengan rasio pembagian 80:20. Pembagian ini dilakukan secara acak namun konsisten menggunakan parameter ‘random_state=42’, untuk memastikan hasil eksperimen tetap sama setiap kali kode dijalankan. Pembagian ini bertujuan agar model dapat dilatih pada sebagian data, lalu dievaluasi performanya secara objektif terhadap data yang belum pernah dilihat sebelumnya.

Maka dari Kode Program 4.8 tersebut, diperoleh hasil bahwa sebanyak 6.400 data sebagai data *training*, dan 1.600 data sebagai data *testing*. Dari data

tersebut model SVM dilatih menggunakan data latih untuk mengenali pola-pola kata yang berasosiasi dengan masing-masing label, kemudian diuji menggunakan data uji untuk mengevaluasi performa klasifikasi. Model SVM yang digunakan dapat dilihat pada Kode Program 4.9 di bawah ini:

Kode Program 4.9 Model SVM

```
1. svm_model = SVC()
2. svm_model.fit(X_train, y_train)
3. y_pred_train = svm_model.predict(X_train)
4. y_pred_test = svm_model.predict(X_test)
```

Objek 'SVC()' dari *library scikit-learn* digunakan untuk membentuk model SVM. Kemudian, model tersebut dilatih menggunakan data latih melalui fungsi 'fit()'. Setelah pelatihan selesai, dilakukan prediksi terhadap dua subset data. 'y_pred_train = svm_model.predict(X_train)' berfungsi melakukan prediksi terhadap data latih untuk mengevaluasi seberapa baik model belajar dari data tersebut. Selanjutnya, 'y_pred_test = svm_model.predict(X_test)' memprediksi data uji untuk mengukur kemampuan generalisasi model terhadap data yang belum pernah dilihat sebelumnya. Berdasarkan Kode Program 4.9 tersebut, diperoleh hasil dari kedua prediksi yang akan digunakan dalam tahap evaluasi performa model pada Gambar 4.4 di bawah ini

Akurasi Data Latih : 0.99078125
Akurasi Data Uji : 0.93875

Classification Report (data uji):				
	precision	recall	f1-score	support
Negatif	0.92	0.96	0.94	814
Positif	0.96	0.92	0.94	786
accuracy			0.94	1600
macro avg	0.94	0.94	0.94	1600
weighted avg	0.94	0.94	0.94	1600

Confusion Matrix (data uji):
[[780 34]
[64 722]]

Gambar 4.9 Hasil Evaluasi Model SVM

Menurut hasil evaluasi model pada Gambar 4.4, klasifikasi sentimen menggunakan algoritma SVM terhadap 1.600 data uji, diperoleh performa model yang sangat baik dengan akurasi mencapai 94%. Model menunjukkan kemampuan

klasifikasi yang seimbang antara kelas sentimen positif dan negatif. Pada kelas negatif, model memiliki nilai *precision* sebesar 92%, *recall* sebesar 96%, dan *f1-score* sebesar 94%. Sementara itu, pada kelas positif, *precision* mencapai 96%, *recall* sebesar 92%, dan *f1-score* juga sebesar 94%. Nilai *macro average* dan *weighted average* untuk *precision*, *recall*, dan *f1-score* masing-masing sebesar 94%, yang menandakan bahwa performa model konsisten dan tidak berat sebelah terhadap salah satu kelas. Hasil ini menunjukkan bahwa pendekatan klasifikasi menggunakan SVM dan pembobotan kata dengan TF-IDF dalam penelitian ini mampu menghasilkan model yang efektif dan akurat dalam mengklasifikasikan sentimen pengguna terhadap aplikasi BTN Mobile.

Jika dibandingkan dengan penelitian terdahulu yang juga menggunakan algoritma SVM, hasil klasifikasi sentimen dalam penelitian ini menunjukkan performa yang lebih unggul. Dalam penelitian (Katherine Ivana Ruslim et al., 2019), kombinasi SVM dengan *lexicon-based features* menghasilkan akurasi sebesar 84,6%, dengan *f1-score* sebesar 85,5%. Sementara itu, (Aldren Marpaung et al., 2024) menunjukkan bahwa algoritma SVM menghasilkan akurasi sebesar 83,2%, lebih tinggi dibandingkan *Naïve Bayes* dalam analisis sentimen terhadap ulasan pengguna aplikasi BCA Mobile. Penelitian (Dwijayanti, 2022) yang menggunakan SVM untuk klasifikasi sentimen pada aplikasi dompet digital menghasilkan akurasi sebesar 89% pada data latih dan 90% pada data uji. Dibandingkan dengan ketiga penelitian tersebut, model klasifikasi dalam penelitian ini mencatat akurasi tertinggi, yaitu 94% pada data uji, dengan *precision*, *recall*, dan *f1-score* yang seimbang antar kelas. Hasil ini memperkuat efektivitas penggunaan SVM yang dipadukan dengan pembobotan TF-IDF dalam membangun model klasifikasi sentimen yang lebih akurat dan stabil dalam konteks aplikasi mobile banking, khususnya BTN Mobile.

Setelah model SVM dilatih menggunakan data pelatihan dan dievaluasi pada data pengujian, model tersebut disimpan dalam bentuk file .pkl untuk memudahkan penggunaan kembali tanpa perlu melatih ulang model dari awal. Selain itu, hasil pembobotan kata menggunakan metode TF-IDF juga disimpan ke dalam file .pkl, yang berfungsi sebagai transformator data teks ke dalam bentuk

vektor numerik sesuai dengan bobot yang telah dihitung selama pelatihan. Penyimpanan kedua file ini bertujuan agar model dapat digunakan secara efisien pada tahap prediksi data ulasan baru. Kode program untuk menyimpan model SVM dan hasil TF-IDF dapat dilihat pada Kode Program 4.10.

Kode Program 4.10 Penyimpanan Model dan Vectorizer

```
1. # Simpan model SVM
2. with open('svm_model.pkl', 'wb') as f:
3.     pickle.dump(svm_model, f)

4. # Simpan TF-IDF vectorizer
5. with open('tfidf_vectorizer.pkl', 'wb') as f:
6.     pickle.dump(tfidf, f)
```

Kode Program 4.10 di atas menyimpan model SVM dan TF-IDF *vectorizer* menggunakan modul *pickle*. Parameter 'wb' pada fungsi 'open()' berarti *write binary*, yaitu membuka file dalam mode tulis biner agar objek *python* dapat disimpan dalam bentuk biner. Variabel 'f' adalah *file handler* yang digunakan sebagai referensi untuk menulis objek ke file tersebut.

Selanjutnya, proses prediksi terhadap data ulasan terbaru diawali dengan mentransformasikan data ulasan yang telah melalui tahap *preprocessing text* ke dalam bentuk vektor TF-IDF dengan menggunakan transformator yang telah disimpan sebelumnya. Setelah data ulasan tersebut berhasil diubah ke dalam bentuk representasi TF-IDF, proses selanjutnya adalah memasukkan data tersebut ke dalam model SVM yang telah dilatih. Berikut Kode Program 4.11 yang menunjukkan penerapan model SVM untuk prediksi sentimen terhadap seluruh ulasan.

Kode Program 4.11 Penerapan Model SVM untuk Prediksi Sentimen

```
1. X_all = df_all['stemmed_content']
2. X_all_tfidf = tfidf.transform(X_all)
3. predictions = svm_model.predict(X_all_tfidf)
```

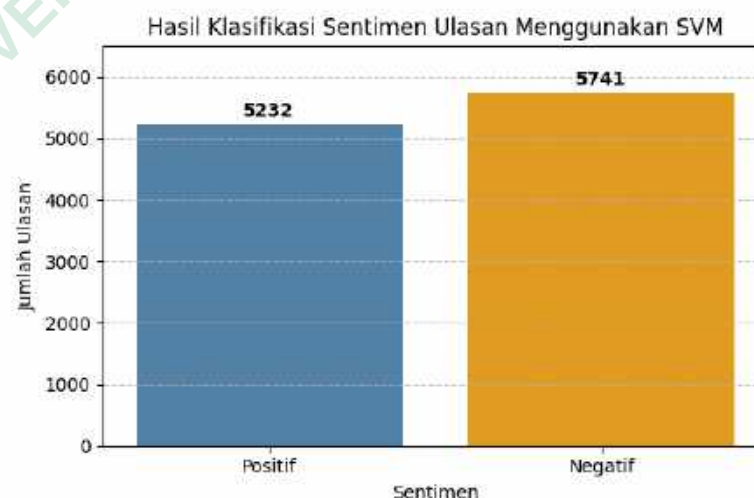
Kode Program 4.11 diatas digunakan untuk melakukan proses klasifikasi terhadap data ulasan yang telah melalui seluruh tahapan *preprocessing text*. Pertama, data teks diambil dari kolom 'stemmed_content'. Selanjutnya, data tersebut ditransformasikan menjadi representasi numerik menggunakan TF-IDF *vectorizer*

yang sebelumnya telah dilatih. Setelah itu, model SVM yang telah disimpan dan dilatih sebelumnya digunakan untuk memprediksi sentimen dari setiap ulasan. Proses ini memungkinkan sistem untuk secara otomatis mengklasifikasikan ulasan pengguna berdasarkan pola-pola bahasa yang telah dipelajari dari data pelatihan sebelumnya. Sampel hasil dari proses klasifikasi SVM dapat dilihat pada Gambar 4.5 berikut:

	ulasan	hasil_SVM
10777	keren fitur layan baik	Positif
7410	eror kadang buka	Negatif
1920	sekian pilih aplikasi mobile bantu butuh perso...	Positif
9155	transfer gagal saldo langsung potong buka noti...	Negatif
8100	suka aplikasi btn mobile uang kontrol handphon...	Positif

Gambar 4.10 Sampel hasil Klasifikasi SVM

Berdasarkan Gambar 4.5, model SVM yang telah dilatih berhasil mengelompokkan ulasan ke dalam dua kategori sentimen, yaitu positif sebanyak 5.232 ulasan dan negatif sebanyak 5.741 ulasan. Hasil ini menunjukkan bahwa mayoritas ulasan pengguna mengandung sentimen negatif, meskipun selisihnya tidak terlalu besar dibandingkan dengan ulasan positif. Proporsi hasil klasifikasi ini divisualisasikan melalui Gambar 4.6 berikut untuk memberikan gambaran penyebaran jumlah ulasan sentimen positif dan negatif terhadap total ulasan pengguna aplikasi BTN Mobile.



Gambar 4.11 Visualisasi Hasil Klasifikasi Sentimen

4.6 IDENTIFIKASI KATA KUNCI DIMENSI HEART

Proses identifikasi kata kunci dimensi HEART dalam penelitian ini diawali dengan menggunakan daftar kata kunci yang telah disusun pada penelitian (Dwijayanti, 2022) yang berjudul “*Analisis Pengalaman Pengguna Berbasis Ulasan Pengguna Aplikasi Dompot Digital Menggunakan Text Mining*”. Setelah itu, dilakukan penggabungan antara kata kunci dari penelitian tersebut dengan kata kunci yang diperoleh dari berbagai referensi dan literatur yang relevan untuk masing-masing dimensi, yaitu *happiness*, *engagement*, *adoption*, *retention*, dan *task success*. Daftar awal kata kunci ini disusun untuk merepresentasikan indikator pengalaman pengguna digital sebagaimana dijelaskan dalam berbagai studi sebelumnya. Hasil akhir identifikasi kata kunci yang digunakan dalam penelitian ini disajikan pada Tabel 4.2.

Tabel 4.2 Daftar Kata Kunci dari Referensi dan Literatur

Dimensi	Kata Kunci	Referensi
<i>Happiness</i>	['puas', 'mudah', 'guna', 'nyaman', 'suka', 'bantu', 'buruk', 'kecewa', 'senang', 'susah', 'sangat', 'rekomendasi', 'baik', 'bgus', 'payah', 'ribet', 'enak', 'lengkap', 'gampang', 'rumit', 'ringan', 'repot', 'sulit', 'nyesel', 'nikmat', 'hebat', 'jelek', 'mantap', 'bingung', 'jelas', 'nyesal', 'paham', 'praktis', 'kesel', 'keren', 'berat', 'makasih', 'terimakasih', 'alhamdulillah', 'yuk', 'ayo', 'rekomended', 'layak', 'sesuai', 'kurang', 'ringkas', 'simpler']	(Dwijayanti, 2022)
	['puas', 'senang', 'nyaman', 'menyenangkan', 'mudah digunakan', 'menarik', 'lengkap', 'terbaik', 'suka', 'memuaskan'.]	(Kristi et al., 2022; Rozikin & Asnawi, 2024; Subiksa et al., 2023)
<i>Engagement</i>	['dapat', 'butuh', 'perlu', 'waktu', 'setiap', 'rutin', 'pernah', 'belum', 'biasa', 'harus', 'layan', 'pakai', 'notif', 'jamin', 'notifikasi', 'pemberitahuan', 'blm', 'wajib', 'pake', 'perlu', 'hrs', 'pakek', 'promo', 'ragu', 'diskon',	(Dwijayanti, 2022)

	'giveaway', 'saran', 'mager', 'make', 'uninstal', 'uninstall', 'hapus']	
	['aktif', 'sering digunakan', 'berulang', 'betah', 'tertarik', 'membuka setiap hari', 'sering pakai', 'rutin', 'setiap hari',].	(Kristi et al., 2022; Rozikin & Asnawi, 2024; Subiksa et al., 2023)
<i>Adoption</i>	['tahu', 'penuh', 'tingkat', 'manfaat', 'baru', 'lama', 'fitur', 'percaya', 'fasilitas', 'unduh', 'pengguna', 'dukung', 'daftar', 'kirim', 'uang', 'transaksi', 'ragu', 'bayar', 'install', 'instal', 'download', 'aman', 'belanja', 'berkat', 'solusi', 'andal', 'saldo', 'beli']	(Dwijayanti, 2022)
	['ingin mencoba', 'baru daftar', 'baru pakai', 'gampang registrasi', 'percobaan pertama', 'mulai gunakan', 'registrasi cepat', 'aktivasi', 'cepat instal'].	(Kristi et al., 2022; Rozikin & Asnawi, 2024; Subiksa et al., 2023)
<i>Retention</i>	['sering', 'terus', 'selama', 'kualitas', 'selalu', 'saat', 'bagi', 'harap', 'ingin', 'lagi', 'masuk', 'update', 'versi', 'login', 'premium', 'riwayat', 'jarang', 'keluar', 'logout', 'kadang', 'aktivasi', 'kapok', 'langgan', 'betah', 'mau', 'kemarin', 'hari', 'perbarui', 'upgrade', 'akun', 'verifikasi']	(Dwijayanti, 2022)
	['kembali terus', 'uninstall', , 'tetap pakai', 'jadi kebiasaan', 'tidak tergantikan', 'rutin digunakan', 'setia', 'tetap menggunakan', 'tidak pindah', 'upgrade'].	(Kristi et al., 2022; Rozikin & Asnawi, 2024; Subiksa et al., 2023)
<i>Task Success</i>	['dapat', 'selesai', 'waktu', 'cepat', 'lebih', 'ketika', 'efektif', 'efisien', 'salah', 'error', 'gagal', 'tolak', 'hilang', 'masalah', 'henti', 'perbaiki', 'sukses', 'bisa', 'kendala', 'komplain', 'lelet', 'pending', 'sesuai', 'lancar', 'cepat', 'ganggu', 'tunggu', 'sibuk', 'nunggu', 'ulang', 'loading', 'lag', 'bug', 'kacau', 'lemot', 'lambat', 'nunggu', 'batal', 'beres', 'gmna', 'proses', 'keluh', 'error', 'komplen', 'tunda', 'macet', 'stabil']	(Dwijayanti, 2022)

	['berhasil', 'cepat', 'efisien', 'lancar', 'mudah', 'navigasi jelas', 'minim error', 'panduan jelas', 'user-friendly', 'responsif'].	(Kristi et al., 2022; Rozikin & Asnawi, 2024; Subiksa et al., 2023)
--	--	---

Namun, setelah kata kunci pada Tabel 4.2 tersebut diterapkan pada data ulasan ditemukan bahwa masih terdapat banyak ulasan yang belum dapat terklasifikasi ke dalam dimensi HEART. Oleh karena itu, proses identifikasi dilanjutkan penulis dengan eksplorasi terhadap kata-kata yang sering muncul dalam ulasan pengguna aplikasi *BTN Mobile*. Kata-kata ini kemudian dipetakan ke dalam dimensi HEART dengan mempertimbangkan konteks dan nuansa kalimat secara manual. Pendekatan ini dilakukan untuk memperkaya cakupan kata kunci agar lebih representatif terhadap bahasa alami yang digunakan oleh pengguna. Kata Kunci yang didapatkan dapat dilihat pada Tabel 4.3 berikut.

Tabel 4.3 Daftar Kata Kunci dari Eksplorasi Penulis

Dimensi	Kata Kunci
<i>Happiness</i>	['good', 'kompli', 'benci', 'tipu', 'intuitif', 'estetis', 'tenang', 'kerenn', 'sayang', 'joss', 'sejahtera', 'okee', 'helpful', 'parah', 'tolong', 'best', 'ramah', 'mantab', 'baguss', 'nice', 'gajelas', 'kocak', 'lumayan', 'terima', 'kasih', 'kesan', 'apresiasi', 'bintang', 'jaya', 'friendly', 'super', 'fresh', 'mantapp', 'recommended', 'cepat respons', 'pelayanan ramah', 'bismillah', 'perfect', 'cinta', 'berkah', 'seep', 'mantul', 'jos', 'joss', 'gokil', 'top', 'ok', 'tolol', 'istimewa', 'okew', 'bagussdd', 'bahagia', 'lesat', 'bgusss', 'muas', 'gatel', 'amazing', 'satisfying', 'kece', 'baguz', 'menjengkelkan', 'jelekkk', 'maling', 'pusing', 'gilaaa', 'thanks', 'semangat', 'sesal', 'kagum', 'mumet', 'bangsat', 'sipp', 'sabar', 'wokey', 'love', 'lucu', 'busuk', 'mewah', 'elegant', 'wow', 'debest', 'membosankan'].
<i>Engagement</i>	['shope', 'customer', 'sistem', 'fiturnya', 'ewallet', 'wallet', 'transfer', 'pulsa', 'topup', 'ngecek', 'cek', 'mutasi', 'dana', 'nomor', 'data', 'shopeepay', 'qris', 'kuis', 'simpan', 'kantor', 'shoppepay', 'menjelajah', 'klik banyak fitur', 'lama waktu tinggal', 'suka eksplorasi', 'tidak ingin berhenti', 'ketagihan', 'menu', 'perangkat', 'bulan', 'setiap bulan', 'usaha', 'muncul', 'mendownload', 'lihat', 'apps', 'pesan', 'rooting', 'call', 'selalu buka', 'aktif digunakan', 'intens', 'banyak fitur', 'interaksi', 'pakai terus', 'akses terus', 'tiap hari', 'mtoken', 'produk', 'fee', 'sms', 'sandi', 'handphone', 'code', 'seru', 'interaktif', 'tambah fitur', 'pasif', 'jarang pakai', 'sales', 'make', 'beli', 'bayar', 'akses']

<i>Adoption</i>	['registrasi', 'aktifasi', 'password', 'user', 'bikin', 'langsung', 'simple', 'android', 'aktivasinya', 'password', 'operasi', 'modal', 'syariah', 'launching', 'diinstal', 'diroot', 'digital', 'menarik sejak awal', 'first impression bagus', 'fitur terlihat menarik', 'promosi efektif', 'ajak teman', 'onboarding lancar', 'baru mencoba', 'baru kenal', 'baru login', 'mendaftar', 'pertamakali', 'baru cobain', 'kali', 'proses rumit', 'gagal daftar', 'tidak paham fungsi awal', 'tidak tertarik mencobanya', 'desain membingungkan', 'terlalu banyak informasi di awal', 'malas mulai', 'hambatan teknis', 'tidak ada panduan']
<i>Retention</i>	['jadi kebiasaan', 'tergantikan', 'nasabah', 'lanjut', 'ngulang', 'slalu', 'masuk', 'dipake', 'pesat', 'semenjak', 'kembali lagi', 'setia', 'tetap menggunakan', 'rekomendasi ulang', 'harus memakai', 'teruss', 'balik', 'pakai lagi', 'buka lagi', 'masih digunakan', 'sudah lama', 'masih setia', 'balik lagi', 'kembali login', 'terus update', 'modern', 'kerja', 'terbiasa', 'pindah ke aplikasi lain', 'nyangkut', 'berguna', 'prioritas', 'akurat', 'bank', 'tidak punya daya tarik lagi', 'kembang', 'tahan', 'tidak ada pembaruan', 'berulang', 'pindah', 'solusi', 'rekomendasi', 'bantu', 'butuh', 'biasa', 'praktis', 'aplikasi']
<i>Task Success</i>	['minim error', 'panduan jelas', 'user-friendly', 'kendala', 'error', 'lambat', 'tersendat', 'pembelian', 'pembayaran', 'kode', 'blokir', 'ngebug', 'aneh', 'maju', 'canggih', 'kartu', 'potong', 'tabung', 'tunai', 'transfer', 'cair', 'admin', 'deteksi', 'token', 'setor', 'delay', 'cocok', 'merespon', 'rusak', 'stuck', 'detail', 'ruwet', 'macet', 'hang', 'fitur rusak', 'gonta ganti', 'gabisa', 'skip', 'sensitif', 'laju', 'improve', 'cacat', 'trobek', 'rempang', 'quality', 'lemah', 'terblokir', 'angel', 'apik', 'depan', 'tampil', 'sinyal', 'acc', 'diacc', 'struk', 'ganti', 'baharu', 'fleksibel', 'trouble', 'refund', 'rekening', 'tolak', 'hilang', 'henti', 'perbaiki', 'komplain', 'cepat', 'sibuk', 'nunggu', 'loading', 'lemot', 'batal', 'beres', 'klik', 'banyak', 'fitur', 'lama', 'bisa', 'buka', 'salah', 'pending', 'gampang', 'fungsi', 'membingungkan', 'terlalu', 'informasi', 'panduan', 'masalah', 'ulang', 'lupa', 'tarik', 'jelas', 'transaksi', 'saldo', 'aman', 'sulit', 'uang', 'berat', 'kualitas', 'kurang', 'bingung', 'verifikasi']

Berdasarkan hasil dari Tabel 4.2 dan Tabel 4.3 maka diperoleh hasil akhir dari proses ini yaitu daftar kata kunci yang lebih komprehensif dan mampu menangkap opini pengguna secara lebih akurat untuk setiap dimensi HEART. Berikut hasil akhir daftar kata kunci yang dapat dilihat pada Tabel 4.5 berikut.

Tabel 4.4 Daftar Kata Kunci Dimensi HEART

Dimensi	Kata Kunci
<i>Happiness</i>	['senang', 'nyaman', 'menyenangkan', 'lengkap', 'terbaik', 'good', 'kompli', 'benci', 'tipu', 'intuitif', 'estetis', 'tenang', 'kerenn', 'hebat', 'sayang', 'josss', 'mantap', 'sejahtera', 'okee', 'memuaskan', 'helpful', 'jelek', 'buruk', 'parah', 'susah', 'tolong', 'best', 'ramah', 'mantab', 'baguss', 'nice', 'gajelas', 'kocak', 'lumayan', 'terima', 'kasih', 'terimakasih', 'kesan', 'apresiasi', 'bintang', 'jaya', 'enak', 'alhamdulillah', 'bagusss', 'friendly', 'super', 'fresh', 'mantapp', 'recommended', 'cepat respons', 'pelayanan ramah', 'bismillah', 'perfect', 'cinta', 'berkah', 'seep', 'baik', 'mantul', 'jos', 'jossh', 'gokil', 'top', 'ok', 'tolol', 'ribet', , 'istimewa', 'okew', 'bagussdd', 'bahagia', 'lesat', 'bgusss', 'muas', 'gatel', 'amazing', 'satisfying', 'kece', 'baguz', 'kecewa', 'frustasi', 'menjengkelkan', 'jelek', 'maling', 'pusing', 'gilaaa', 'thanks', 'semangat', 'sesal', 'kagum', 'mumet', 'bangsat', 'sipp', 'sabar', 'wokey', 'love', 'lucu', 'busuk', 'mewah', 'elegant', 'wow', 'debest', 'membosankan', 'puas', 'sangat', 'ringan', 'nyesel', 'nikmat', 'kesel', 'keren', 'makasih', 'yuk', 'ayo', 'rekomended', 'layak', 'ringkas'].
<i>Engagement</i>	['layan', 'shope', 'customer', 'sistem', 'fiturnya', 'ewallet', 'wallet', 'transfer', 'pulsa', 'topup', 'ngecek', 'cek', 'mutasi', 'dana', 'nomor', 'data', 'shopeepay', 'qris', 'aktif', 'sering digunakan', 'kuis', 'simpan', 'kantor', 'shoppepay', 'menjelajah', 'klik banyak fitur', 'lama waktu tinggal', 'membuka setiap hari', 'suka eksplorasi', 'tidak ingin berhenti', 'sering pakai', 'setiap hari', 'ketagihan', 'menu', 'promo', 'notif', 'perangkat', 'bulan', 'setiap bulan', 'usaha', 'muncul', 'mendownload', 'lihat', 'apps', 'pesan', 'rooting', 'call', 'tidak bisa lepas', 'selalu buka', 'aktif digunakan', 'intens', 'banyak fitur', 'interaksi', 'pakai terus', 'akses terus', 'tiap hari', 'mtoken', 'produk', 'fee', 'sms', 'sandi', 'handphone', 'code', 'dipakai terus', 'seru', 'interaktif', 'tambah fitur', 'pasif', 'jarang pakai', 'sales', 'merasa bosan', 'pernah', 'belum', 'jamin', 'notifikasi', 'pemberitahuan', 'blm', 'wajib', 'hrs', 'pakek', 'diskon', 'giveaway', 'saran', 'mager', 'make', 'uninstal', 'suka', 'beli', 'bayar', 'akses', 'tertarik']
<i>Adoption</i>	['registrasi', 'aktifasi', 'password', 'user', 'bikin', 'langsung', 'simple', 'instal', 'android', 'aktivasinya', 'password', 'operasi', 'modal', 'unduh', 'syariah', 'cepat instal', 'ingin mencoba', 'tabungan', 'daftar online', 'launching', 'diinstal', 'diroot', 'digital', 'menarik sejak awal', 'first impression bagus', 'fitur terlihat menarik', 'promosi efektif', 'ajak teman', 'onboarding lancar', 'baru daftar', 'gampang registrasi', 'percobaan pertama', 'baru mencoba', 'baru kenal', 'baru login', 'mulai gunakan', 'registrasi cepat', 'baru', 'pertama kali', 'mobile', 'banking', 'bangking', 'coba',

	'awal', 'mendaftar', 'pertamakali', 'baru cobain', 'kali', 'proses rumit', 'gagal daftar', 'tidak paham fungsi awal', 'tidak tertarik mencobanya', 'desain membingungkan', 'terlalu banyak informasi di awal', 'malas mulai', 'hambatan teknis', 'tidak ada panduan', 'download', 'install', 'online', 'tahu', 'penuh', 'fasilitas', 'pengguna', 'dukung', 'daftar', 'belanja', 'berkat', 'andal', 'manfaat', 'menarik', 'bagus', 'paham']
<i>Retention</i>	['kembali terus', 'uninstall', 'jadi kebiasaan', 'tergantikan', 'rutin digunakan', 'nasabah', 'lanjut', 'ngulang', 'slalu', 'masuk', 'dipake', 'pesat', 'keluar', 'semenjak', 'tetap pakai', 'kembali lagi', 'setia', 'tetap menggunakan', 'upgrade', 'masih bertahan', 'pengalaman berulang', 'rekomendasi ulang', 'harus memakai', 'lagi', 'teruss', 'balik', 'pakai lagi', 'buka lagi', 'tetap', 'masih digunakan', 'sudah lama', 'masih setia', 'balik lagi', 'kembali login', 'terus update', 'update', 'membantu', 'memudahkan', 'mempermudah', 'gratis', 'modern', 'kerja', 'terbiasa', 'pindah ke aplikasi lain', 'nyangkut', 'berguna', 'prioritas', 'akurat', 'bank', 'kembang', 'tahan', 'jarang digunakan', 'berulang', 'pindah', 'selama', 'saat', 'bagi', 'harap', 'premium', 'riwayat', 'logout', 'kadang', 'aktivasi', 'langgan', 'betah', 'mau', 'kemarin', 'perbarui', 'digunakan', 'percaya', 'kapok', 'sering', 'setiap', 'hari', 'ingin', 'pakai', 'akun', 'versi', 'selalu', 'terus', 'jarang', 'malas', 'kembali', 'lain', 'masuk', 'ada', 'uninstall', 'rutin', 'pake', 'solusi', 'rekomendasi', 'harus', 'bantu', 'butuh', 'biasa', 'praktis', 'aplikasi']
<i>Task Success</i>	['berhasil', 'efisien', 'minim error', 'panduan jelas', 'user-friendly', 'kendala', 'error', 'lambat', 'tersendat', 'pembelian', 'pembayaran', 'eror', 'kode', 'blokir', 'bug', 'ngebug', 'ganggu', 'respon', 'aneh', 'lelet', 'maju', 'canggih', 'kartu', 'potong', 'tabung', 'tunai', 'transfer', 'cair', 'admin', 'deteksi', 'token', 'setor', 'delay', 'cocok', 'merespon', 'rusak', 'stuck', 'detail', 'ruwet', 'macet', 'hang', 'fitur rusak', 'gonta ganti', 'gabisa', 'skip', 'sensitif', 'laju', 'improve', 'cacat', 'trobel', 'rempong', 'quality', 'lemah', 'lag', 'terblokir', 'angel', 'apik', 'depan', 'tampil', 'sinyal', 'acc', 'diacc', 'struk', 'ganti', 'baharu', 'fleksibel', 'trouble', 'kacau', 'refund', 'rekening', 'selesai', 'lebih', 'ketika', 'tolak', 'hilang', 'henti', 'perbaiki', 'komplain', 'cepat', 'sibuk', 'nunggu', 'loading', 'lemot', 'batal', 'beres', 'mudah', 'sukses', 'cepat', 'respons', 'klik', 'banyak', 'fitur', 'lama', 'waktu', 'bisa', 'buka', 'salah', 'pending', 'efektif', 'lancar', 'gampang', 'login', 'proses', 'rumit', 'gagal', 'fungsi', 'membingungkan', 'terlalu', 'informasi', 'panduan', 'masalah', 'ulang', 'lupa', 'tarik', 'jelas', 'transaksi', 'saldo', 'aman', 'sulit', 'simpler', 'uang', 'berat', 'kualitas', 'kurang', 'bingung', 'verifikasi', 'sesuai', 'dapat', 'ragu']

4.7 KLASIFIKASI HEART FRAMEWORK

Klasifikasi *HEART framework* dalam penelitian ini bertujuan untuk memahami pengalaman pengguna aplikasi *BTN Mobile* berdasarkan lima dimensi yaitu *happiness*, *engagement*, *adoption*, *retention*, dan *task success*. Proses klasifikasi *HEART framework* dimulai dengan tahap persiapan data untuk klasifikasi manual. Dari total 10.973 ulasan yang telah melalui *preprocessing* dan klasifikasi sentimen menggunakan algoritma SVM, dipilih secara acak sebanyak 2.500 ulasan untuk dijadikan dataset pelatihan. Pemilihan acak ini dilakukan untuk memastikan kesetaraan data dan menghindari bias dalam proses pembelajaran model. Klasifikasi manual dilakukan oleh peneliti dengan menyesuaikan kata kunci yang telah ditetapkan untuk setiap dimensi HEART. Proses ini melibatkan pembacaan manual terhadap setiap ulasan dan penentuan dimensi yang paling sesuai berdasarkan konteks dan makna ulasan tersebut. Tabel 4.5 menunjukkan distribusi hasil klasifikasi manual untuk 2.500 ulasan.

Tabel 4.5 Distribusi Klasifikasi Manual *HEART Framework*

Dimensi HEART	Jumlah Ulasan
<i>Happiness</i>	500
<i>Engagement</i>	500
<i>Adoption</i>	500
<i>Retention</i>	500
<i>Task Success</i>	500
Total	2.500

Berdasarkan Tabel 4.5, dapat dilihat bahwa distribusi klasifikasi manual dilakukan secara seimbang untuk masing-masing dimensi HEART, yaitu sebanyak 500 ulasan untuk setiap dimensi. 2.500 ulasan yang sudah terklasifikasi manual tersebut, kemudian digunakan sebagai data pelatihan untuk membangun model klasifikasi otomatis berbasis *cosine similarity*. Model klasifikasi dibangun dengan merepresentasikan setiap ulasan dalam bentuk vektor menggunakan pendekatan TF-IDF. Metode TF-IDF dipilih karena kemampuannya dalam memberikan bobot yang tepat untuk setiap kata berdasarkan frekuensi kemunculan dan signifikansinya dalam corpus.

Proses pengembangan model dimulai dengan menggabungkan dataset pelatihan 2.500 ulasan yang sudah berlabel dimensi HEART dengan data set uji yaitu 10.973 ulasan hasil klasifikasi sentimen. Seluruh ulasan kemudian di vektorisasi menggunakan TF-IDF untuk menghasilkan representasi numerik. Setelah vektor terbentuk, data dibagi menjadi dua bagian yaitu vektor untuk data berlabel sebagai referensi, dan vektor untuk data yang belum diklasifikasikan. Implementasi proses ini ditunjukkan pada Kode Program 4.12.

Kode Program 4.12 Vektorisasi TF IDF Data Ulasan Berlabel dan Tak Berlabel

```

1. # Load file
2. df_labeled = pd.read_csv("2500_Ulasan_HEART.csv")
3. df_all = pd.read_csv("Hasil_Sentimen_10973.csv")

4. # Gabungkan teks untuk vektorisasi
5. combined_texts = pd.concat([df_labeled["ulasan"],
    df_all["ulasan"]], ignore_index=True)

6. # Lakukan vektorisasi menggunakan TF-IDF
7. vectorizer = TfidfVectorizer()
8. tfidf_matrix = vectorizer.fit_transform(combined_texts)

9. # Memisahkan vektor: 2500 ulasan berlabel dan 10.973 ulasan
    tak berlabel
10. labeled_vectors = tfidf_matrix[:len(df_labeled)]
11. all_vectors = tfidf_matrix[len(df_labeled):]

```

Setelah proses vektorisasi dari Kode Program 4.12 selesai, klasifikasi untuk seluruh ulasan dilakukan dengan menghitung nilai kemiripan kosinus antara setiap vektor ulasan tak berlabel dengan seluruh vektor dari ulasan berlabel. Untuk setiap ulasan, sistem akan mengambil tiga ulasan berlabel yang memiliki tingkat kemiripan tertinggi. Label dari ketiga ulasan ini kemudian digunakan untuk melakukan voting mayoritas guna menentukan dimensi HEART yang paling sesuai bagi ulasan tersebut. Proses ini diimplementasikan dalam Kode program 4.13.

Kode Program 4.13 Klasifikasi Dimensi HEART Menggunakan Cosine Similarity

```

1. k = 3
2. predicted_labels = []
3. for i in range(all_vectors.shape[0]):
4.     sims = cosine_similarity(all_vectors[i], labeled_vectors)[0]
5.     top_k_idx = np.argsort(sims)[-k:][::-1]
6.     top_k_labels = [labeled_labels[j] for j in top_k_idx]
7.     majority = Counter(top_k_labels).most_common(1)[0][0]
8.     predicted_labels.append(majority)

```

Kode Program 4.13 menunjukkan implementasi proses klasifikasi HEART dengan menggunakan metrik *cosine similarity* yang dilakukan secara otomatis dan efisien untuk seluruh dataset ulasan hasil klasifikasi SVM. Berikut penjelasannya:

a. 'k = 3'

Menentukan nilai k sebanyak 3, yang berarti sistem akan mempertimbangkan tiga tetangga ulasan terdekat dengan nilai kemiripan tertinggi dalam proses klasifikasi.

b. 'for i in range(all_vectors.shape[0]):'

Melakukan iterasi untuk setiap vektor dalam dataset ulasan yang belum berlabel. Variabel i menunjukkan indeks dari setiap ulasan dalam data tak berlabel.

c. 'sims = cosine_similarity(all_vectors[i], labeled_vectors)[0]'

Menghitung nilai *cosine similarity* antara satu ulasan tak berlabel 'all_vectors[i]' dengan seluruh ulasan berlabel 'labeled_vectors'. Hasilnya adalah array satu dimensi sims yang berisi skor kemiripan terhadap semua data berlabel.

d. 'top_k_idx = np.argsort(sims)[-k:][::-1]'

Mengurutkan skor kemiripan dari yang terkecil ke terbesar menggunakan 'np.argsort(sims)', lalu mengambil k nilai tertinggi dengan [-k:], dan membalik urutannya menjadi dari tertinggi ke terendah dengan [::-1].

- e. `'top_k_labels = [labeled_labels[j] for j in top_k_idx]'`

Mengambil label dari ketiga tetangga terdekat berdasarkan indeks `top_k_idx`.

Label ini akan digunakan untuk melakukan voting.

- f. `'majority = Counter(top_k_labels).most_common(1)[0][0]'`

Menghitung frekuensi kemunculan masing-masing label di antara tetangga terdekat menggunakan 'Counter', lalu mengambil label yang paling banyak muncul 'most common' sebagai hasil prediksi akhir.

Dengan selesainya proses klasifikasi, langkah selanjutnya adalah melakukan evaluasi terhadap performa model. Evaluasi ini bertujuan untuk mengukur sejauh mana model klasifikasi mampu mengidentifikasi dan memetakan dimensi HEART secara akurat berdasarkan data uji. Hasil evaluasi disajikan dalam bentuk *classification report* yang mencakup metrik *precision*, *recall*, *f1-score*, dan *accuracy*, serta *confusion matrix* untuk melihat distribusi prediksi pada masing-masing kelas. Gambar 4.7 berikut menyajikan hasil evaluasi dari model klasifikasi *cosine similarity* yang dibangun.

Classification Report:				
	precision	recall	f1-score	support
Adoption	0.90	0.86	0.88	610
Engagement	0.90	0.85	0.87	544
Happiness	0.83	0.93	0.88	624
Retention	0.88	0.90	0.89	544
Task Success	0.89	0.85	0.87	580
accuracy			0.88	2902
macro avg	0.88	0.88	0.88	2902
weighted avg	0.88	0.88	0.88	2902
Confusion Matrix:				
[[526 10 32 21 21]				
[18 460 26 18 22]				
[11 17 578 11 7]				
[11 7 24 489 13]				
[17 18 33 17 495]]				

Gambar 4.12 Hasil Evaluasi Model *Cosine Similarity*

Berdasarkan hasil evaluasi pada Gambar 4.7, model klasifikasi menunjukkan kinerja yang memuaskan dengan nilai *accuracy* sebesar 88%. Hal ini menunjukkan bahwa model mampu mengklasifikasikan ulasan pengguna ke dalam lima dimensi HEART dengan tingkat ketepatan yang cukup tinggi. Secara keseluruhan, metrik evaluasi seperti *precision*, *recall*, dan *f1-score* juga menunjukkan performa yang seimbang di setiap kelas, dengan nilai *macro avg* dan *weighted avg* masing-masing sebesar 88%. Dimensi *happiness* memiliki performa tertinggi dalam hal *recall*, yaitu sebesar 93%, dan nilai *f1-score* sebesar 88%, yang menunjukkan bahwa model sangat baik dalam mengenali ulasan yang berkaitan dengan kebahagiaan pengguna. Sementara itu, dimensi *engagement* dan *task success* memiliki nilai *recall* dan *f1-score* yang sama, yaitu masing-masing sebesar 85% dan 87%, meskipun sedikit lebih rendah dibandingkan dimensi lainnya seperti *retention* dan *adoption*, namun dimensi *engagement* dan *task success* tetap berada dalam kategori performa yang baik.

Pada bagian *confusion matrix*, distribusi kesalahan antar kelas dapat diamati lebih rinci. Misalnya, pada baris pertama yaitu *adoption*, sebagian prediksi salah diklasifikasikan ke dimensi *engagement* dan *happiness*, yang menunjukkan bahwa beberapa ulasan memiliki kemiripan konteks antar dimensi. Meski demikian, angka prediksi benar tetap dominan pada setiap dimensi, mencerminkan efektivitas model dalam membedakan karakteristik tiap dimensi HEART. Secara keseluruhan, hasil evaluasi ini mengindikasikan bahwa pendekatan klasifikasi berbasis *cosine similarity* mampu menghasilkan prediksi yang cukup akurat dan dapat diandalkan dalam analisis dimensi HEART berdasarkan ulasan pengguna. Sampel Hasil dari proses klasifikasi *HEART framework* dapat dilihat pada Gambar 4.8 berikut:

	ulasan	hasil_SVM	kategori
10777	keren fitur layan baik	Positif	Happiness
7410	eror kadang buka	Negatif	Retention
1920	sekian pilih aplikasi mobile bantu butuh perso...	Positif	Adoption
9155	transfer gagal saldo langsung potong buka noti...	Negatif	Retention
8100	suka aplikasi btn mobile uang kontrol handphon...	Positif	Task Success

Gambar 4.13 Sampel Hasil Klasifikasi *HEART Framework*

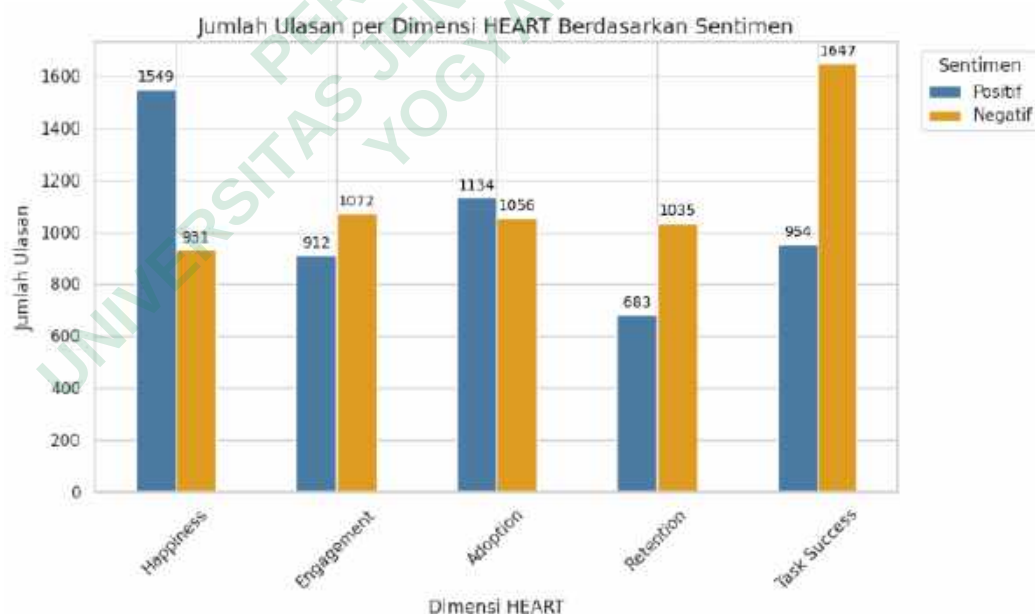
Jika dibandingkan dengan penelitian terdahulu yang juga menggunakan pendekatan *HEART framework*, model klasifikasi dalam penelitian ini menunjukkan keunggulan yang signifikan. Keunggulan penelitian ini terletak pada pendekatan klasifikasi dimensi HEART yang lebih spesifik, yakni dengan memfokuskan pada pengklasifikasian langsung kata kunci ke dalam satu dimensi utama berdasarkan dominansi kemunculan, sehingga mengurangi ambiguitas dalam penentuan dimensi. Keunggulan tersebut berdampak pada akurasi yang menyebabkan penelitian ini lebih unggul dari pada penelitian sebelumnya. Misalnya pada penelitian yang dilakukan oleh (Dwijayanti, 2022), penelitian tersebut memiliki kelemahan yang dapat diidentifikasi. Salah satu kelemahan yang ditemukan terletak pada metode pengklasifikasian dimensi HEART yang digunakan, yaitu melalui pendekatan LDA. Pendekatan ini menghasilkan pengklasifikasian kata kunci ke dalam beberapa dimensi sekaligus, yang dapat menimbulkan ambiguitas dan mengurangi ketepatan klasifikasi. Selain itu, meskipun akurasi klasifikasi sentimen dalam penelitian tersebut cukup tinggi yaitu 90%, akurasi pengklasifikasian ke dimensi HEART hanya mencapai 85%, menunjukkan adanya ruang untuk peningkatan terutama dalam aspek spesifik pemetaan dimensi. Penelitian ini mencoba menjawab kelemahan tersebut dengan pendekatan yang lebih terarah, sehingga menghasilkan hasil klasifikasi dimensi yang lebih presisi.

4.8 ANALISIS HASIL

Pada tahap ini dilakukan analisis terhadap ulasan pengguna aplikasi *BTN Mobile* yang diperoleh dari Google Play Store. Analisis dilakukan dengan pendekatan dua tahap, yaitu klasifikasi sentimen menggunakan metode SVM, kemudian pengelompokan berdasarkan dimensi HEART untuk mengevaluasi pengalaman pengguna secara lebih komprehensif. Klasifikasi sentimen dilakukan terhadap 10.973 ulasan pengguna. Dengan menggunakan algoritma SVM, ulasan dikategorikan ke dalam dua jenis sentimen, yaitu positif dan negatif. Hasilnya menunjukkan bahwa sebanyak 5.232 ulasan (47,7%) tergolong sentimen positif, sementara 5.741 ulasan (52,3%) tergolong sentimen negatif. Hasil ini

mengindikasikan bahwa secara umum, persepsi pengguna terhadap aplikasi BTN *Mobile* masih cenderung negatif, sehingga menunjukkan adanya potensi masalah dalam kualitas layanan atau fitur aplikasi yang perlu menjadi perhatian lebih lanjut.

Setelah sentimen diklasifikasikan, setiap ulasan dianalisis dan dikategorikan ke dalam lima dimensi HEART, yaitu *happiness*, *engagement*, *adoption*, *retention*, dan *task success*. Tujuan klasifikasi ini adalah untuk memperoleh pemahaman yang lebih dalam mengenai dimensi mana yang paling berkontribusi terhadap pengalaman pengguna, baik secara positif maupun negatif. Evaluasi menggunakan dimensi HEART dapat memberikan keunggulan karena mampu menangkap pengalaman pengguna dari berbagai sudut pandang, tidak hanya dari sisi emosional tetapi juga perilaku dan keberhasilan penggunaan. Kelima dimensi ini saling melengkapi, sehingga memungkinkan analisis yang lebih menyeluruh dan mendalam dibandingkan metode evaluasi yang hanya mengandalkan satu indikator saja. Gambar 4.9 dibawah ini merupakan hasil yang didapatkan peneliti untuk memperoleh wawasan lebih mendalam terhadap persepsi pengguna menggunakan dimensi *HEART framework*.



Gambar 4.14 Visualisasi Hasil Ulasan per Dimensi HEART Berdasarkan Sentimen

Hasil dan visualisasi pada Gambar 4.9 menunjukkan bahwa dari total 10.973 ulasan sentimen yang berhasil terklasifikasi dimensi HEART:

- a. Dimensi *happiness* memiliki jumlah sentimen positif tertinggi, yaitu 1.549 ulasan, menunjukkan bahwa mayoritas pengguna merasa puas dan senang terhadap aspek kenyamanan aplikasi, tampilan antarmuka, dan kemudahan dalam mengakses fitur. Fitur yang memberikan kontribusi besar terhadap sentimen positif ini antara lain fitur cek saldo cepat tanpa perlu login ulang, tampilan dashboard yang ringkas sehingga menampilkan informasi rekening secara jelas dan navigasi menu yang sederhana. Sedangkan, ulasan negatif pada dimensi ini relatif rendah, yaitu 931 ulasan, menunjukkan bahwa secara umum aplikasi memberikan kesan yang baik pada aspek emosional.
- b. Dimensi *task success* merupakan dimensi dengan jumlah ulasan tertinggi secara keseluruhan yaitu 2.601 ulasan dan tingkat sentimen negatif paling dominan yaitu 1.647 ulasan. Hal ini menunjukkan adanya tantangan dalam keberhasilan pengguna dalam menyelesaikan tugas atau tujuan pengguna seperti kegagalan atau eror dalam pengaksesan fitur aplikasi, transfer dan pembayaran tagihan atau top up melalui aplikasi.
- c. Dimensi *adoption* memiliki total 2.190 ulasan, dengan 1.134 ulasan positif dan 1.056 ulasan negatif. Distribusi sentimen yang relatif seimbang ini menunjukkan bahwa sebagian besar pengguna memiliki pengalaman yang beragam saat pertama kali menggunakan aplikasi. Ulasan positif mengapresiasi kemudahan dalam proses aktivasi awal dan kejelasan informasi pada antarmuka awal aplikasi. Namun, ulasan negatif masih cukup banyak, terutama terkait kendala saat registrasi, verifikasi OTP dan wajah, hingga kesulitan login awal yang memerlukan beberapa proses manual.
- d. Dimensi *engagement* memiliki total 1.984 ulasan, terdiri dari 912 ulasan positif dan 1.072 ulasan negatif. Meskipun selisih antara kedua jenis sentimen tidak terlalu besar, dominasi ulasan negatif menunjukkan adanya kebutuhan peningkatan pada aspek keterlibatan pengguna seperti adanya pengguna yang mengeluhkan kurangnya pembaruan fitur dan tampilan yang monoton. Sedangkan untuk ulasan positif, pengguna mengapresiasi fitur-fitur seperti

riwayat transaksi yang mudah diakses, menu favorit, dan kemudahan dalam menjadwalkan pembayaran rutin. Hal ini menunjukkan bahwa pengguna membutuhkan fitur yang lebih interaktif dan menarik untuk mempertahankan keterlibatan dalam jangka panjang.

- e. Dimensi *retention* merupakan dimensi dengan jumlah ulasan paling sedikit, yaitu 1.718 ulasan, terdiri dari 683 ulasan positif dan 1.035 ulasan negatif. Sebagian ulasan positif menyoroti fitur praktis seperti penyimpanan daftar penerima favorit dan kemudahan mengakses riwayat transaksi. Namun, dominasi ulasan negatif mengindikasikan adanya hambatan dalam mempertahankan pengguna dalam jangka panjang. Pengguna mengeluhkan masalah teknis seperti aplikasi sering logout otomatis, gangguan sistem saat transaksi, serta error saat login. Hal ini mencerminkan bahwa sebagian pengguna tidak memiliki pengalaman yang cukup baik untuk terus menggunakan aplikasi dalam jangka panjang sehingga dapat diartikan bahwa loyalitas pengguna belum sepenuhnya stabil.

Temuan ini menunjukkan bahwa pengalaman pengguna aplikasi *BTN Mobile* masih menghadapi berbagai tantangan, terutama dalam hal keberhasilan penyelesaian tugas utama seperti transaksi dan login, serta kemudahan dalam proses awal penggunaan aplikasi. Sebaliknya, aspek kenyamanan atau kepuasan emosional pengguna menunjukkan kinerja yang cukup baik. Hal tersebut menunjukkan bahwa meskipun pengguna mungkin merasa nyaman menggunakan aplikasi dari sisi tampilan atau antarmuka, mereka tetap mengalami kesulitan yang signifikan dalam menjalankan fungsi utama aplikasi.

Secara keseluruhan, hasil analisis ini mengungkapkan bahwa faktor-faktor utama yang memengaruhi pengalaman pengguna *BTN Mobile* mencakup aspek teknis seperti keandalan sistem dan stabilitas aplikasi, kemudahan dalam proses pendaftaran atau onboarding pengguna baru, serta efektivitas fitur utama aplikasi. Di sisi lain, aspek visual dan kenyamanan penggunaan menjadi kekuatan yang perlu dipertahankan. Dengan memahami distribusi sentimen pada masing-masing dimensi HEART, maka pihak pengembang dapat lebih terarah dalam merancang

strategi perbaikan dan pengembangan fitur sesuai kebutuhan dan harapan pengguna.

Selain memberikan gambaran mengenai dimensi yang masih perlu diperbaiki dalam aplikasi *BTN Mobile*, hasil analisis ini juga menunjukkan bahwa ulasan pengguna dapat berfungsi sebagai sumber data yang kaya dan representatif dalam mengevaluasi pengalaman pengguna secara menyeluruh. Tidak seperti metode tradisional seperti kuesioner atau wawancara yang memiliki keterbatasan dalam hal jumlah responden, waktu, dan kedalaman eksplorasi, ulasan pengguna yang tersedia secara daring memiliki karakteristik alami dan real-time. Melalui penelitian ini, penulis berhasil mengidentifikasi pola-pola persepsi pengguna dalam skala besar, tanpa memerlukan proses pengumpulan data secara langsung.

Dengan demikian, hasil ini memperkuat bahwa ulasan pengguna, dapat menjadi alternatif yang efektif dan efisien dalam studi evaluasi pengalaman pengguna. Bagi pengembang maupun peneliti, temuan ini memberikan wawasan bahwa pengumpulan dan analisis ulasan pengguna tidak hanya mampu menggambarkan kondisi riil dari sisi pengguna, tetapi juga berpotensi digunakan sebagai landasan dalam pengambilan keputusan pengembangan aplikasi yang lebih tepat sasaran.

Sebagai pelengkap dari penelitian yang telah diuraikan penulis, gambaran yang lebih menyeluruh mengenai penelitian ini dapat diperoleh melalui pemaparan kelebihan dan kekurangan berikut ini:

1. Kelebihan

- a. Penelitian ini menggunakan rating pengguna sebagai dasar dalam menentukan ulasan positif dan negatif terhadap aplikasi. Sebelum dilakukan pelabelan, penelitian ini telah menerapkan proses verifikasi data untuk memastikan validitas dan konsistensi antara isi ulasan dengan skor rating. Tahapan ini mencakup penghapusan data duplikat serta pemeriksaan manual terhadap kesesuaian isi ulasan dengan nilai rating pengguna. Hasilnya, diperoleh ulasan yang layak dan siap digunakan dalam tahap analisis, sehingga meningkatkan kualitas data dan keakuratan hasil klasifikasi.

- b. Penelitian ini menggunakan algoritma SVM untuk klasifikasi sentimen dengan berhasil mencapai akurasi sebesar 94%. Hasil ini membuktikan bahwa SVM memiliki performa yang sangat baik dalam mengolah data ulasan pengguna. Temuan ini juga konsisten dengan berbagai penelitian sebelumnya yang menunjukkan bahwa SVM memiliki tingkat akurasi lebih tinggi dibandingkan metode lain dalam analisis sentimen, sehingga semakin memperkuat validitas metode yang digunakan dalam penelitian ini.
- c. Penelitian ini menggunakan metode *cosine similarity* untuk melakukan klasifikasi ulasan ke dalam lima dimensi *HEART framework* dan berhasil mencapai akurasi sebesar 88%. Hasil ini menunjukkan bahwa *cosine similarity* mampu mengelompokkan ulasan pengguna secara efektif, terutama ketika distribusi kata kunci yang digunakan seimbang di setiap dimensi. Temuan ini membuktikan bahwa keberimbangan kata kunci sangat berpengaruh terhadap performa klasifikasi, di mana akurasi yang dihasilkan cenderung lebih tinggi jika setiap dimensi memiliki representasi kata kunci yang cukup.

2. Kekurangan

- 1. Penelitian hanya mengandalkan data ulasan dari Google Play Store, sehingga belum merepresentasikan sudut pandang pengguna dari platform lain seperti App Store, yang kemungkinan besar memberikan sudut pandang yang lebih beragam mengenai pengalaman pengguna.
- 2. Penyusunan daftar kata kunci untuk klasifikasi *HEART framework* dalam penelitian ini belum melibatkan pendapat atau validasi dari pakar secara langsung. Seluruh daftar kata kunci disusun berdasarkan referensi literatur dan pemahaman peneliti terhadap konteks ulasan pengguna. Meskipun pendekatan ini didasarkan pada studi yang relevan, tanpa keterlibatan pakar secara langsung, terdapat risiko bahwa makna tidak sepenuhnya sesuai dengan interpretasi yang digunakan dalam pakar profesional.