BAB 4
HASIL PENELITIAN

4.1 RINGKASAN HASIL PENELITIAN

Penelitian ini mengumpulkan data ulasan pengunjung dari *platform* Google Maps untuk RSU X di Kota A dan Kota B, dengan total 1.544 ulasan yang diperoleh melalui teknik *scraping* menggunakan API SerpAPI dan Python. Proses pra-pemrosesan data dilakukan untuk meningkatkan kualitas data sebelum analisis, meliputi langkah-langkah seperti *case folding, text cleaning, text normalization, tokenizing, text filtering (stopword removal), stemming*, *text validation*, *labeling data*, dan representasi vektor. Hasil dari tahapan pra-pemrosesan menghasilkan data bersih sebanyak 1.474 ulasan. Setelah data diproses, model sentimen dilakukan menggunakan algoritma SVM, yang menunjukkan *accuracy* sebesar 90%. Hasil analisis sentimen pada ulasan positif mendominasi sebanyak 1.118, menunjukkan bahwa sebagian besar pengunjung memberikan apresiasi terhadap pelayanan rumah sakit. Namun demikian, ulasan negatif tetap muncul dalam jumlah 356, mengindikasikan adanya aspek-aspek pelayanan yang masih perlu diperbaiki.

Selanjutnya, analisis *topic modeling* dilakukan menggunakan LDA untuk mengidentifikasi tema utama dari ulasan. Pengunjung RSU X Kota A menghargai keramahan tenaga medis, kecepatan layanan, dan kenyamanan fasilitas. Di RSU X Kota B, pengunjung juga memuji keramahan staf, kualitas layanan rawat inap dan jalan, serta kebersihan fasilitas. Kedua rumah sakit berhasil memberikan pelayanan cepat, ramah, dan lingkungan yang nyaman. Keluhan dari pengunjung RSU X Kota A terkait waktu tunggu lama, antrian tidak efisien, dan ketidakjelasan informasi. Di RSU X Kota B, keluhan serupa muncul tentang waktu tunggu, komunikasi dokter yang kurang jelas, dan ketidaksesuaian jadwal. Kedua rumah sakit memerlukan perbaikan dalam hal efisiensi layanan, transparansi informasi, dan sistem penjadwalan untuk meningkatkan kepuasan pengunjung.

4.2 PENGUMPULAN DATA

Pengumpulan data penelitian ini memanfaatkan data ulasan yang diambil dari *platform* Google Maps di RSU X, yang mencakup dua lokasi, yaitu Kota A dan Kota B. Data ulasan yang digunakan terdiri dari 988 baris data untuk RSU X Kota A dan 556 baris data untuk RSU X Kota B, dengan kolom yang mencakup nama, tanggal ulasan, rating, dan teks ulasan. Data yang digunakan mencakup seluruh ulasan yang tersedia, dengan memastikan hanya ulasan yang memiliki teks (tidak kosong atau hanya berisi rating) yang diambil untuk analisis. Data diambil menggunakan teknik *scraping*, dengan API yang didapat dari *platform* SerpAPI dan bahasa pemrograman Python. Hasil ulasan yang terambil dapat dilihat pada **Tabel 4.1**.

Tabel 4.1 Hasil *Scraping*

No	Ulasan	Rating
1.	Tolong lebih ditraining lagi karyawannya terakait mekanisme asuransi biar gak bingung, prosesnya agak lama	3
2.	Periksa hari ini krn uda langganan di rs ini, di ugd di cek tensi ,30menit nunggu dokter gak dtg2, sementara pasien di sebelah dikunjungi dokter terus dgn pertanyaan yg sama,menurun bgt pelayanan nya, orang udah pusing niat ke ugd biar cepet. Alhasil pindah rs	2
3.	Ini farmasinya parahhhhhh bgtt dahh, kirain swasta baguss. Interior nya nyaman. Padahal pake umum antrinya 1 jam, gmna yg pakw bpjs. D	1
4.	Pelayanan dokter nya oke, untuk dokter bedah. Bagian pendaftaran sedikit menunggu. Fasilitas nya sudah oke	5
5.	Fisik cukup bersih, pengkondisian udara di setiap ruang publik cukup memadai. Padahal dilihat dari segi umur, sdh lumayan lama rumah sakit ini beroperasi. Design interiornya simple, tapi berasa sangat nyaman, hommy bagi pengunjung/pasien.	4

4.3 PRA-PEMROSESAN DATA

Sebelum melakukan tahap *modeling*, pra-pemrosesan data perlu agar menghasilkan data yang berkualitas. Tahapan pra-pemrosesan ini berguna untuk membersihkan data, melakukan labeling data, dan mengubah data ulasan kedalam

bentuk vektor. Tahap ini membantu dalam mengurangi noise pada data sehingga model dapat belajar lebih efektif dan menghasilkan klasifikasi yang lebih akurat.

4.3.1 Case Folding

Tahapan pertama dalam pra-pemrosesan data adalah *case folding*. *Case folding* berguna untuk mengubah huruf besar menjadi huruf kecil dalam teks. Berikut **Kode Program 4.1** untuk menjalankan proses *case folding*.

Kode Program 4.1 *Case Folding*

```
1. def CaseFoldingText(text):
2.       if isinstance(text, str):
3.            text = text.casefold()
4.       return text
```

Function casefold() adalah *function* dalam python untuk mengubah teks menjadi huruf kecil semua. Casefold() menjadi metode agresif terbaru dalam Python yang sebelumnya menggunakan lower(). Hasil dari proses *case folding* dapat dilihat pada **Gambar 4.1**.

	review_text	text_casefoldingText
0	Tolong lebih ditraining lagi karyawannya terak...	tolong lebih ditraining lagi karyawannya terak...
1	Periksa hari ini krn uda langganan di rs ini, ...	periksa hari ini krn uda langganan di rs ini, ...
2	Ini farmasinya parahhhhhh bgtt dahh, kirain sw...	ini farmasinya parahhhhhh bgtt dahh, kirain sw...
3	Pelayanan dokter nya oke, untuk dokter bedah. ...	pelayanan dokter nya oke, untuk dokter bedah. ...
4	Fisik cukup bersih, pengkondisian udara di set...	fisik cukup bersih, pengkondisian udara di set...

Gambar 4.1 Hasil *Case Folding*

4.3.2 Text Cleaning

Text Cleaning berguna untuk menghapus karakter atau simbol yang tidak relevan dalam teks ulasan. *Function* ini bermanfaat meningkatkan kualitas data ulasan, sehingga dapat meningkatkan akurasi sentimen. **Kode Program 4.2** menampilkan kode program *text cleaning* yang menggunakan beberapa *library* seperti *regular expression*. *Regular expression* digunakan untuk memanipulasi text dengan menghapusnya berdasarkan pola yang dibuat.

Kode Program 4.2 *Text Cleaning*

```
1. def cleaning_text(text):
2.       text = re.sub(r'@[A-Za-z0-9]+', '', text)
3.       text = re.sub(r'#[A-Za-z0-9]+', '', text)
4.       text = re.sub(r"http\S+", '', text)
5.       text = re.sub(r'[0-9]+', '', text)
6.       text = re.sub(r'[^\w\s]', '', text
7.       text = re.sub(r'(.)\1{2,}', r'\1', text)
8.       text = text.replace('\n', ' '
9.       text = text.strip()
10.      return text
```

Berikut adalah beberapa *function* kode dalam proses *text cleaning*, yang digunakan untuk membersihkan teks dari elemen-elemen yang tidak relevan sebelum diproses lebih lanjut. Pada *function* ini *regular expression* disingkat dengan nama re untuk mempermudah proses pemanggilan *library*.

a. re.sub(r'@[A-Za-z0-9]+', '', text) : menghapus mention atau tag (@).
b. re.sub(r'#[A-Za-z0-9]+', '', text) : menghapus simbol tagar (#).
c. re.sub(r"http\S+", '', text) : menghapus *link*/URL.
d. re.sub(r'[0-9]+', '', text) : menghapus angka dalam teks.
e. re.sub(r'[^\w\s]', '', text) : menghapus tanda baca atau karakter khusus.
f. re.sub(r'(.)\1{2,}', r'\1', text) : menghapus huruf berulang dalam setiap kata.
g. text.replace('\n', ' ') : menghapus *enter*/baris baru, sehingga memastikan bahwa teks dalam setiap *row* dalam bentuk satu baris.
h. text.strip() : menghapus spasi yang berlebihan.

Proses *text cleaning* menjadi salah satu tahapan penting dalam pra-pemrosesan data, yang dapat meningkatkan *accuracy* model. Hasil dari proses *text cleaning* dapat dilihat pada **Gambar 4.2**.

	text_casefoldingText	text_clean
0	tolong lebih ditraining lagi karyawannya terak...	tolong lebih ditraining lagi karyawannya terka...
1	periksa hari ini krn uda langganan di rs ini, ...	periksa hari ini krn uda langganan di rs ini d...
2	ini farmasinya parahhhhhh bgtt dahh, kirain sw...	ini farmasinya parah bgt dah kirain swasta bag...
3	pelayanan dokter nya oke, untuk dokter bedah. ...	pelayanan dokter nya oke untuk dokter bedah ba...
4	fisik cukup bersih, pengkondisian udara di set...	fisik cukup bersih pengkondisian udara di seti...

Gambar 4.2 Hasil *Cleaning*

4.3.3 Text Normalization

Tahapan *text normalization* dilakukan untuk mengubah kata-kata di data ulasan yang sebelumnya tidak baku atau tidak sesuai menjadi bentuk baku yang standar yang mudah dipahami. Dalam data ulasan terdapat banyak ulasan yang menggunakan bahasa informal, singkatan, atau ejaan yang tidak standar, sehingga perlu dilakukan proses normalisasi menggunakan kamus *slangwords*. Dalam tahapan ini kamus *slangwords* diperoleh dari sumber *open-source* Github yang berisi kumpulan kata tidak baku dan padanannya dalam bahasa formal. Kamus *slangwords* yang digunakan dalam penelitian ini dapat dilihat pada **Gambar 4.3**.

	aj	saja
0	ajk	ajak
1	alm	almarhum
2	almt	alamat
3	alaamat	alamat
4	ank	anak
5	ap	apa
6	aq	saya
7	aqu	aku
8	asek	asyik
9	asik	asyik

Gambar 4.3 Sampel Kamus *Slangwords* Indonesia
(Sumber: https://github.com/azizsembada/text-preprocessing-api)

Kamus *slangwords* ini terdiri dari 1.800 *slang*. Untuk menggunakannya dalam proses normalisasi teks, kamus tersebut perlu dimuat terlebih dahulu. Proses tersebut dapat dilihat pada kode program di **Lampiran 1**. Setelah kamus *slangwords* dimuat, setiap kata dalam teks akan dibandingkan dengan entri dalam kamus, dan jika ditemukan kata *slang*, maka kata tersebut akan digantikan dengan bentuk bakunya. Proses ini bertujuan untuk meningkatkan kualitas data teks sebelum digunakan dalam analisis sentimen atau *topic modeling*. Kode untuk menjalankan proses ini dapat dilihat pada kode program di **Lampiran 1**.

Proses tersebut akan mengurangi variasi kata yang dapat memengaruhi hasil analisis. Setelah dilakukan normalisasi, teks yang telah disesuaikan dapat digunakan untuk tahapan pemrosesan lebih lanjut. Hasil dari normalisasi teks ini dapat dilihat pada **Gambar 4.4**.

	text_clean	text_slang
0	tolong lebih ditraining lagi karyawannya terka...	tolong lebih ditraining lagi karyawannya terka...
1	periksa hari ini krn uda langganan di rs ini d...	periksa hari ini karena sudah langganan di rs ...
2	ini farmasinya parah bgt dah kirain swasta bag...	ini farmasinya parah banget dah kirain swasta ...
3	pelayanan dokter nya oke untuk dokter bedah ba...	pelayanan dokter nya oke untuk dokter bedah ba...
4	fisik cukup bersih pengkondisian udara di seti...	fisik cukup bersih pengkondisian udara di seti...

Gambar 4.4 Hasil *Normalization (Slangwords)*

4.3.4 Tokenizing

Tahapan selanjutnya dalam pra-pemrosesan data adalah *tokenizing*. *Tokenizing* berguna untuk memecah teks ulasan menjadi kata dalam bentuk list. Berikut **Kode Program 4.3** adalah kode program untuk *tokenizing*.

Kode Program 4.3 *Tokenizing*

```
1. def tokenizingText(text):
2.       text = word_tokenize(text)
3.       return text
```

Tokenizing dapat mempermudah dan mempercepat proses *filtering*, karena data telah dipisah menjadi potongan kata yang terstruktur. Hasil dari *tokenizing* dapat dilihat pada **Gambar 4.5**.

	text_slang	text_token
0	tolong lebih ditraining lagi karyawannya terka...	['tolong', 'lebih', 'ditraining', 'lagi', 'kar...
1	periksa hari ini karena sudah langganan di rs ...	['periksa', 'hari', 'ini', 'karena', 'sudah', ...
2	ini farmasinya parah banget dah kirain swasta ...	['ini', 'farmasinya', 'parah', 'banget', 'dah'...
3	pelayanan dokter nya oke untuk dokter bedah ba...	['pelayanan', 'dokter', 'nya', 'oke', 'untuk',...
4	fisik cukup bersih pengkondisian udara di seti...	['fisik', 'cukup', 'bersih', 'pengkondisian', ...

Gambar 4.5 Hasil *Tokenizing*

4.3.5 Text Filtering

Text Filtering atau *stopwords* berguna untuk menghapus kata yang tidak memiliki makna. Kata-kata yang biasa muncul dan memiliki makna dapat mempengaruhi hasil akurasi pemodelan, contohnya kata ‘dan’, ‘yang’, dan sejenisnya. Berikut kode program *filtering* dapat dilihat pada **Kode Program 4.4**.

Kode Program 4.4 *Filtering* (*Stopwords*)

```
1. def remove_stopwords(text):
2.     if isinstance(text, str):
3.         return stopword_remover.remove(text)
4. return text
```

Pada proses *filtering*, penulis menggunakan *custom stopwords* untuk menghapus kata-kata yang tidak bermakna, baik dari daftar standar bawaan *library* maupun yang telah ditentukan. Hasil dari *filtering* dapat dilihat pada **Gambar 4.6**.

	text_token	text_stopwords
0	['tolong', 'lebih', 'ditraining', 'lagi', 'kar...	lebih ditraining karyawannya terkait mekanisme...
1	['periksa', 'hari', 'ini', 'karena', 'sudah', ...	periksa hari langganan rs ugd cek tensi menit ...
2	['ini', 'farmasinya', 'parah', 'banget', 'dah'...	farmasinya parah banget dah kirain swasta bagu...
3	['pelayanan', 'dokter', 'nya', 'oke', 'untuk',...	pelayanan dokter oke dokter bedah bagian penda...
4	['fisik', 'cukup', 'bersih', 'pengkondisian', ...	fisik cukup bersih pengkondisian udara setiap ...

Gambar 4.6 Hasil *Filtering* (*Stopwords*)

4.3.6 Stemming

Stemming bekerja dengan menghapus akhiran atau imbuhan yang ada dalam kata. Contohnya kata "pelayanan" menjadi "layanan", "pendaftaran" menjadi "daftar", dan "merawat" menjadi "rawat". Kode program untuk menjalankan *stemming* dapat dilihat pada **Kode Program 4.5**.

Kode Program 4.5 *Stemming*

```
1. stemmer_factory = StemmerFactory()
2. stemmer = stemmer_factory.create_stemmer()
3. def stemmedtext(text):
4.     if isinstance(text, str):
5.         return stemmer.stem(text)
6.     return text
```

Kode **Kode Program 4.6** akan menghapus akhiran atau imbuhan menggunakan *library* Sastrawi. *Stemming* banyak digunakan dalam pra-pemrosesan data sebelum analisis lebih lanjut seperti analisis sentimen dan *topic modeling*. Hasil dari *stemming* dapat dilihat pada **Gambar 4.7**.

	text_stopwords	text_stemming
0	lebih ditraining karyawannya terkait mekanisme...	lebih training karyawan kait mekanisme asurans...
1	periksa hari langganan rs ugd cek tensi menit ...	periksa hari karena langganan rs ugd cek tensi...
2	farmasinya parah banget dah kirain swasta bagu...	farmasi parah banget kira swasta bagus interio...
3	pelayanan dokter oke dokter bedah bagian penda...	layanan_dokter oke dokter bedah bagi daftar di...
4	fisik cukup bersih pengkondisian udara setiap ...	fisik cukup bersih kondisi udara tiap ruang pu...

Gambar 4.7 Hasil *Stemming*

4.3.7 Text Validation

Tahap selanjutnya dalam pra-pemrosesan data adalah tahapan *text validation*. *Text validation* berguna untuk memastikan data yang akan digunakan dalam analisis tidak mengandung *duplicates* dan data kosong (NaN). Sebelum melakukan *text validation*, penting untuk melihat jumlah data NaN dan *duplicates.* Berikut **Kode Program 4.6** untuk menghitung jumlah data NaN dan *duplicates.*

Kode Program 4.6 Menghitung Data NaN dan *Duplicates*

```
1. nan_count = clean_df['stemmed_text'].isna().sum().sum()
2. print(f"Jumlah data NaN: {nan_count}")
3. duplicate_count=clean_df['stemmed_text'].duplicated().sum()
4. print(f"Jumlah data duplikat: {duplicate_count}")
```

Hasil dari perhitungan data NaN dan *duplicates* dapat dilihat pada **Gambar 4.8**. Hasil perhitungan data NaN dan *duplicates* menunjukkan bahwa tidak terdapat data NaN, tetapi terdapat 100 data *duplicates*.

```
Jumlah data NaN Sebelum Validate Data: 0
Jumlah data duplikat Sebelum Validate Data: 100
```

Gambar 4.8 Jumlah Data NaN dan *duplicates* (Sebelum Validasi)

Penanganan data NaN dapat menggunakan *function* dropna() bawaan dari Python, dan *function* drop_duplicates() untuk menangani data *duplicate*. Proses penanganan ini dapat dilihat pada **Kode Program 4.7**.

Kode Program 4.7 *Text Validation*

```
1. def validate_data(df, column):
2.       df_cleaned = df.dropna(subset=[column])
3.       df_cleaned=df_cleaned.drop_duplicates(subset=[column])
4.       return df_cleaned
```

Proses validasi data akan menghapus data NaN dan data *duplicates* yang ada di data. Hasil dari proses validasi ini ditampilkan pada **Gambar 4.9**, yang memperlihatkan data setelah *text validation*.

```
Jumlah data NaN Sesudah Validate Data: 0
Jumlah data duplikat Sesudah Validate Data: 0
```

Gambar 4.9 Jumlah Data NaN dan *duplicates* (Setelah Validasi)

Setelah dilakukan beberapa tahapan pra-pemrosesan data terhadap data ulasan pengunjung, terjadi sedikit pengurangan jumlah data. Untuk RSU X Yogyakarta, jumlah data menjadi 987. Sementara itu, untuk RSU X Kota B, jumlah data berkurang lebih signifikan, menjadi 487.

Selain proses otomatis, tahap *text validation* juga mencakup pemeriksaan manual terhadap data hasil pra-pemrosesan. Pemeriksaan ini dilakukan untuk memastikan bahwa proses pembersihan teks telah berjalan dengan baik. Proses cek manual dapat mengidentifikasi potensi kesalahan yang tidak terdeteksi secara otomatis, serta melakukan perbaikan terhadap data yang salah atau tidak sesuai, sehingga kualitas data tetap terjaga dan layak digunakan untuk analisis lebih lanjut.

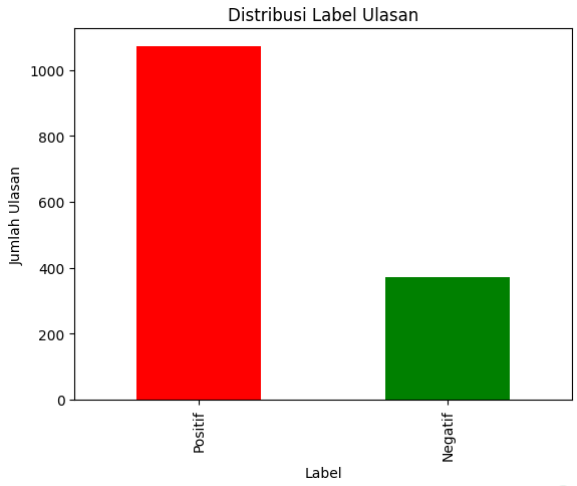
4.3.8 Labeling Data

Setelah memastikan data ulasan bersih, tahap berikutnya adalah *labeling data*. *Labeling data* dilakukan dengan skema *average labeling* berdasarkan rating yang diberikan pengunjung di Google Maps. Rating 1, 2, dan 3 dikategorikan sebagai sentimen negatif, yang mencerminkan adanya keluhan pengunjung. Rating 4 dan 5 dikategorikan sebagai sentimen positif, yang menunjukkan kepuasan pengunjung. **Kode Program 4.8** menunjukkan kode *labeling data*.

Kode Program 4.8 *Labeling Data*

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

Berdasarkan kode *labeling data*, hasil *labeling data* menunjukkan jumlah label sentimen positif sebanyak 1072 dan label sentimen negatif sebanyak 372. Berikut visualisasi hasil *labeling data* dapat dilihat pada **Gambar 4.10**.



Gambar 4.10 Distribusi *Labeling Data*

4.3.9 Representasi Vektor

Sebelum membangun model sentimen, data ulasan perlu dikonversi dalam bentuk representasi vektor. TF-IDF menjadi salah satu metode yang paling umum digunakan dalam merepresentasikan teks ke numerik. Berikut **Kode Program 4.9** menunjukkan kode TF-IDF untuk representasi vektor data ulasan.

Kode Program 4.9 Representasi Vektor dengan TF-IDF

```
1. X = data_sentimen[text_stemming']
2. tfidf = TfidfVectorizer()
3. X_tfidf = tfidf.fit_transform(X)
```

TF-IDF akan mengukur seberapa penting kata dalam sebuah dokumen atau ulasan terhadap keseluruhan kumpulan dokumen atau ulasan. Sehingga TF-IDF memberikan bobot nilai yang lebih tinggi pada setiap kata yang memiliki makna yang signifikan. Berikut **Gambar 4.11** menunjukkan hasil dari tahapan representasi vektor dengan TF-IDF.

ada	anak	antri	bagi	bagus	baik	banget	banyak	baru	beri
0.000000	0.000000	0.000000	0.0	0.000000	0.073693	0.000000	0.000000	0.197717	0.105226
0.077195	0.000000	0.201325	0.0	0.000000	0.000000	0.000000	0.055845	0.078442	0.027832
0.000000	0.000000	0.220398	0.0	0.077505	0.000000	0.000000	0.000000	0.085873	0.000000
0.000000	0.000000	0.080251	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
0.000000	0.148725	0.000000	0.0	0.000000	0.224128	0.000000	0.000000	0.000000	0.000000

Gambar 4.11 Hasil Representasi Vektor

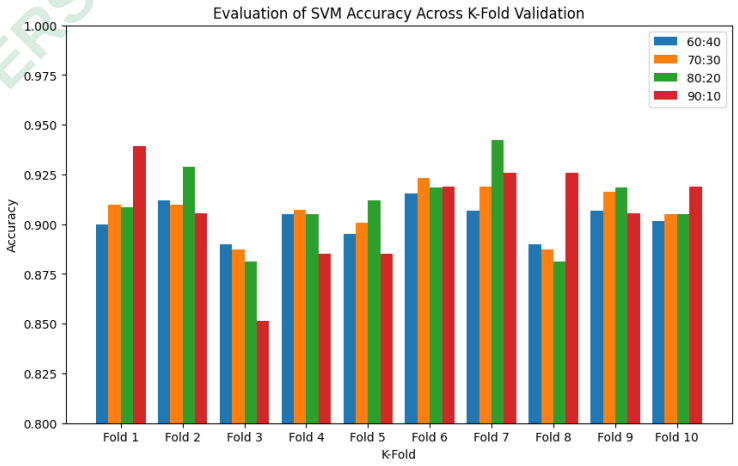
4.4 MODEL SENTIMENT

4.4.1 K-Fold Validation

K-Fold *validation* diterapkan untuk memastikan data terbagi dengan baik dan menghindari bias dalam *splitting data* sebelum proses pembangunan model sentimen. K-Fold bekerja dengan membagi data menjadi K (*folds*) yang sama besar, kemudian model dilakukan *training* dan *testing* sebanyak K kali menggunakan kombinasi data yang berbeda berdasarkan pembagian pada K-Fold. Kode program untuk proses K-Fold dapat dilihat di **Lampiran 2**.

Hasil dari K-Fold akan menunjukkan hasil *accuracy* dari setiap *Fold* yang diproses sebanyak K kali. Untuk memahami hasil dari K-Fold *validation*, hasil validasi divisualisasikan dalam bentuk *bar chart*. *Bar chart* ini menampilkan *accuracy* dari setiap *fold*, sehingga memudahkan dalam melihat performa model di berbagai subset *splitting data*. **Lampiran 3** menunjukkan Kode Program yang digunakan untuk membuat visualisasi *bar chart* K-Fold.

Hasil visualisasi K-Fold menggunakan berbagai rasio *splitting data* memberikan gambaran bagaimana pengaruh *splitting data* terhadap performa model sentimen. Hasil visualisasi menampilkan warna biru (60:40), orange (70:30), hijau (80:20), dan merah (90:10). Setiap warna *bar chart* mewakili hasil *accuracy* dari setiap *splitting data*. Visualisasi tersebut dapat dilihat pada **Gambar 4.12**.



Gambar 4.12 Visualisasi Hasil K-Fold *Validation*

Selain memvisualkan hasil K-Fold, penulis menampilkan hasil K-Fold dalam bentuk tabel. Tabel ini memberikan informasi hasil yang lebih rinci

mengenai performa model pada setiap subset *splitting data*. Untuk menghasilkan tabel hasil K-Fold dapat menggunakan kode program dapat dilihat di **Lampiran 4**.

Hasilnya menunjukkan bahwa rasio 80:20 menghasilkan rata-rata akurasi tertinggi, yaitu sebesar 0.9101, sehingga dipilih sebagai rasio yang paling optimal untuk pembagian data dalam proses pelatihan dan pengujian model. Tabel hasil K-Fold dapat dilihat pada **Gambar 4.13**.

	60:40	70:30	80:20	90:10
Fold 1	0.9	0.909706546275395	0.9084745762711864	0.9391891891891891
Fold 2	0.911864406779661	0.909706546275395	0.9288135593220339	0.9054054054054054
Fold 3	0.8898305084745762	0.8871331828442438	0.8813559322033898	0.8513513513513513
Fold 4	0.9050847457627119	0.90744920993228	0.9050847457627119	0.8851351351351351
Fold 5	0.8949152542372881	0.9006772009029346	0.911864406779661	0.8851351351351351
Fold 6	0.9152542372881356	0.9232505643340858	0.9186440677966101	0.918918918918919
Fold 7	0.9067796610169492	0.9187358916478555	0.9423728813559322	0.9256756756756757
Fold 8	0.8898305084745762	0.8871331828442438	0.8813559322033898	0.9256756756756757
Fold 9	0.9067796610169492	0.9164785553047404	0.9186440677966101	0.9054054054054054
Fold 10	0.9016949152542373	0.9051918735891648	0.9050847457627119	0.918918918918919
Average	0.9022033898305086	0.9065462753950339	0.9101694915254237	0.9060810810810811

Gambar 4.13 Tabel Hasil K-Fold *Validation*

4.4.2 Splitting Data

Sebelum membangun model sentimen, data ulasan perlu dibagi menjadi dua bagian utama yaitu data *train* dan data *test*. Pada penelitian ini data ulasan pengunjung dibagi dengan rasio perbandingan 80:20 sesuai dengan hasil terbaik pada tahap K-Fold *validation*. 80:20 artinya 80% data digunakan untuk *training*, dan 20% data digunakan untuk *testing*. *Splitting data* menggunakan *library* Sklearn dengan *function* train_test_split yang dapat dilihat pada **Kode Program 4.10**.

Kode Program 4.10 *Splitting Data*

```
1. X_train, X_test, y_train, y_test = train_test_split(X_tfidf,
   y, test_size=0.2, random_state=42)
```

Proses pembagian data menggunakan *train_test_split* menghasilkan dua set data yaitu data *train* sebanyak 1.179 sampel dan data *test* sebanyak 295 sampel. Hasil *splitting data* dapat dilihat pada **Gambar 4.14**.

```
Jumlah data Train: 1179
Jumlah data Test: 295
```

Gambar 4.14 Hasil *Splitting Data*

4.4.3 Model *Support Vector Machine*

Membangun model sentimen dengan algoritma SVM dari Sklearn, Sklearn menyediakan pustaka untuk implementasi berbagai algoritma *machine learning*, termasuk SVM. Model sentimen dibangun menggunakan data yang telah direpresentasikan dalam bentuk vektor dengan TF-IDF sebelumnya. *Splitting data* telah ditentukan dengan proporsi 80:20 berdasarkan hasil K-Fold *validation*. Kode program untuk membangun model sentimen dengan SVM dapat dilihat pada **Kode Program 4.11**.

Kode Program 4.11 Model *Support Vector Machine*

```
1. svm_model = SVC()
2. svm_model.fit(X_train.toarray(), y_train)
3. y_pred_train_svm = svm_model.predict(X_train.toarray())
4. y_pred_test_svm = svm_model.predict(X_test.toarray())
```

4.4.4 Model Evaluation

Setelah model sentimen berhasil dibangun, evaluasi model dilakukan dengan mengukur *accuracy* pada setiap data *train* dan *test*. *Accuracy* menunjukkan sejauh mana model dapat memprediksi data yang belum dilihat atau dilatih sebelumnya. Penulis juga menyajikan evaluasi model melalui *classification report* yang memuat metrik *precision, recall*, dan F1-*Score* untuk menggambarkan kinerja model dalam mengklasifikasikan ulasan positif dan negatif.

Model sentimen menggunakan *Support Vector Machine* (SVM) menghasilkan akurasi pengujian sebesar 90%. *Classification report* menunjukkan bahwa kelas Negatif memiliki *precision* sebesar 0.91, *recall* 0.72, dan F1-*score* 0.81, sedangkan kelas Positif memiliki *precision* 0.90, *recall* 0.97, dan F1-*score* 0.94. Secara keseluruhan, model mencapai akurasi 91% dengan nilai *macro average* dan *weighted average* yang tinggi, yang mengindikasikan performa model yang baik dan seimbang dalam mengklasifikasikan kedua kelas.

Sebagai pembanding, model *Linear Regression* yang diterapkan pada data yang sama menghasilkan akurasi lebih rendah, yaitu sebesar 77,9%. Meskipun *precision* dan *recall*-nya cukup kompetitif pada kelas tertentu, performanya secara umum masih berada di bawah model SVM. Oleh karena itu, SVM dipilih sebagai

model utama karena mampu memberikan hasil klasifikasi sentimen yang lebih akurat dan andal. Selain itu, jika dibandingkan dengan penelitian sebelumnya dalam tinjauan pustaka, model SVM dalam penelitian ini juga menunjukkan performa yang kompetitif. Akurasi SVM yang dicapai sebesar 90,5% lebih tinggi dibandingkan studi sebelumnya yang menggunakan SVM dengan akurasi maksimum 88%, dan setara dengan akurasi tertinggi *Linear Regression* dari studi lain, yaitu 90,5%. Hal ini menunjukkan bahwa pendekatan analisis sentimen yang diterapkan pada penelitian ini cukup efektif dan dapat dijadikan rujukan untuk studi sejenis di masa mendatang. Perbandingan kedua model tersebut dapat dilihat pada **Gambar 4.15.**

Linear Regression - accuracy: 0.7796610169491526

Classification Report:

	precision	recall	f1-score	support
Negatif	0.91	0.72	0.81	80
Positif	0.90	0.97	0.94	215
accuracy			0.91	295
macro avg	0.91	0.85	0.87	295
weighted avg	0.91	0.91	0.90	295

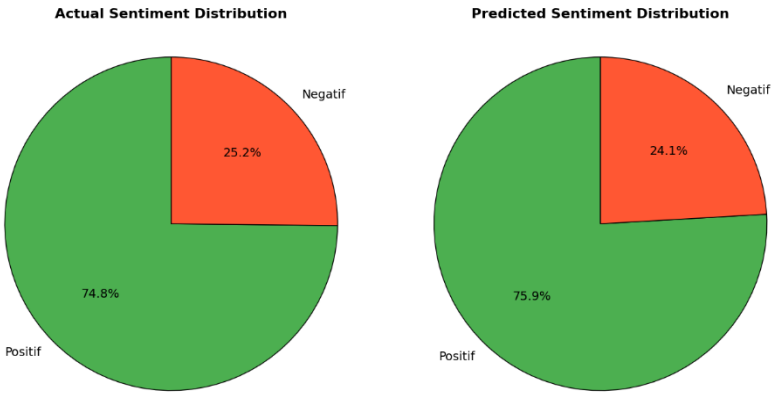
Support Vector Machine - accuracy: 0.9050847457627119

Classification Report (data uji):

	precision	recall	f1-score	support
Negatif	0.91	0.72	0.81	80
Positif	0.90	0.97	0.94	215
accuracy			0.91	295
macro avg	0.91	0.85	0.87	295
weighted avg	0.91	0.91	0.90	295

Gambar 4.15 Hasil Evaluasi Model

Hasil visualisasi pada **Gambar 4.16** menunjukkan bahwa distribusi sentimen pada data asli lebih dominan dengan penilaian positif, mencapai 75%, sementara sisanya, yaitu 25%, memberikan penilaian negatif. Di sisi lain, pada data prediksi, model menunjukkan bahwa 76% pengunjung memberikan sentimen positif, sedangkan 24% memberikan sentimen negatif. Perbandingan ini menunjukkan bahwa model prediksi cukup akurat dalam mencocokkan sentimen yang diberikan oleh pengunjung, dengan sedikit perbedaan antara prediksi dan data asli.



Gambar 4.16 Hasil Sentimen

Gambar 4.17 pada kolom Predicted_Label2 merepresentasikan hasil prediksi sentimen oleh model. Sementara itu, kolom *correct* menjadi acuan evaluasi performa model dengan menunjukkan apakah hasil prediksi sesuai dengan label asli di kolom sentimen. Nilai *True* menandakan prediksi yang tepat, sedangkan *False* menunjukkan kesalahan model. Informasi dapat mengevaluasi tingkat akurasi serta mengidentifikasi pola kesalahan yang mungkin terjadi dalam proses klasifikasi sentimen.

	text_lemmatized	sentimen	Predicted_Label2	correct
0	lebih training karyawan kait mekanisme asurans...	Negatif	Negatif	True
1	periksa hari karena langganan rs ugd cek tensi...	Negatif	Negatif	True
2	farmasi parah banget kira swasta bagus interio...	Negatif	Negatif	True
3	layanan_dokter oke dokter bedah bagi daftar di...	Positif	Positif	True
4	fisik cukup bersih kondisi udara tiap ruang pu...	Positif	Positif	True

Gambar 4.17 Hasil Sentimen

4.5 Model Topic Modeling

4.5.1 Data Segmentation

Proses selanjutnya dalam penelitian ini adalah membangun model *topic modeling* dengan LDA. Sebelum membangun model, data ulasan perlu dilakukan segmentasi data. Tujuan dari segmentasi data ini adalah untuk melihat lebih dalam mengenai topik kepuasan dan keluhan pengunjung dari masing-masing rumah sakit. Kode program segmentasi data ini dapat dilihat pada **Kode Program 4.12**.

Kode Program 4.12 Segmentasi Data

```
1. a_data = data[data['location'] == 'Kota A']
2. b_data = data[data['location'] == 'Kota B']
3. positive_a = a_data[a_data['Predicted_Label'] ==
   'Positif']['text_lemmatized']
4. negative_a = a_data[a_data['Predicted_Label'] ==
   'Negatif']['text_lemmatized']
5. positive_b = b_data[b_data['Predicted_Label'] ==
   'Positif']['text_lemmatized']
6. negative_b = b_data[b_data['Predicted_Label'] ==
   'Negatif']['text_lemmatized']
```

Tahap pertama dalam segmentasi data ini adalah dengan memisahkan data berdasarkan lokasi, yaitu RSU X di Kota A (*a_data*) dan RSU X di Kota B (*b_data*),

analisis dapat lebih spesifik terhadap setiap rumah sakit. Selanjutnya, data juga dipisahkan berdasarkan hasil analisis sentimen, yaitu ulasan dengan sentimen positif (*positive_a* dan *positive_b*) serta ulasan dengan sentimen negatif (*negative_a* dan *negative_b*). Pemisahan ini memungkinkan identifikasi topik yang lebih terfokus, baik dalam aspek kepuasan maupun keluhan pengunjung, sehingga dapat memberikan wawasan yang lebih akurat dalam model *topic modeling*. Hasil dari segmentasi data dapat dilihat pada **Gambar 4.18**.

```
Jumlah Data Positif RSU X Kota A: 703
Jumlah Data Negatif RSU X Kota A: 284
Jumlah Data Positif RSU X Kota B: 416
Jumlah Data Negatif RSU X Kota B: 71
```

Gambar 4.18 Hasil Segmentasi Data

4.5.2 Coherence Score

Sebelum membangun model LDA, dilakukan perhitungan *coherence score* untuk menentukan jumlah topik yang optimal dalam setiap kategori data. *Coherence score* berfungsi untuk mengevaluasi kualitas topik yang dihasilkan dengan mengukur keterkaitan antar kata dalam suatu topik. Nilai *coherence* yang lebih tinggi menunjukkan bahwa topik yang terbentuk lebih bermakna dan mudah diinterpretasikan.

Pada implementasinya, sebuah *function* dibuat untuk menghitung *coherence score* dan mencari jumlah topik optimal. Tahap pertama dalam *function* ini adalah membuat kamus (*dictionary*) dan korpus dari teks yang diberikan, lalu melatih model LDA dengan berbagai jumlah topik dalam rentang yang ditentukan. Setiap model dievaluasi menggunakan *CoherenceModel*, dan hasilnya diplot untuk membantu memilih jumlah topik terbaik berdasarkan nilai *coherence* tertinggi. *Function* ini kemudian dipanggil untuk setiap kategori data RSU X, Dengan cara ini, setiap kelompok data dianalisis secara terpisah untuk menemukan jumlah topik yang paling sesuai dengan pola distribusi kata dalam data tersebut. Kode program *Coherence Score* ini dapat dilihat di **Lampiran 5**.

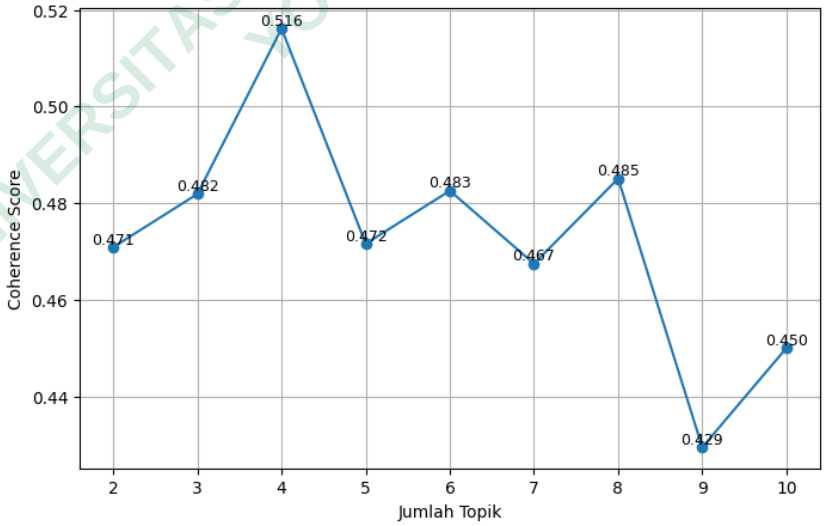
1. Coherence Score Kota A Positif

Function determine_optimal_topics dipanggil untuk menghitung *coherence score* pada data positif Kota A. Proses ini menghasilkan jumlah topik optimal dalam ulasan sentimen positif di Kota A. Hasilnya divisualisasikan dalam bentuk grafik *coherence score* yang menunjukkan nilai *coherence* untuk setiap jumlah topik yang diuji. Pemanggilan *function determine_optimal_topics* tersebut dapat dilihat di **Kode Program 4.13**.

Kode Program 4.13 *Coherence Score* Positif Kota A

```
1. determine_optimal_topics(tokenized_texts1, "Positive Kota
   A", max_topics=10)
```

Berdasarkan hasil pemanggilan *function determine_optimal_topics* pada data positif Kota A, sumbu X merepresentasikan jumlah topik yang diuji, sedangkan sumbu Y menunjukkan nilai *coherence score*. **Gambar 4.19** merupakan hasil pengujian *coherence score* di data positif Kota A. Dari grafik tersebut, dapat diamati bahwa nilai *coherence* tertinggi diperoleh saat jumlah topik 4 dengan skor 0.516, kemudian mengalami perubahan dengan nilai terendah di topik 9 sebesar 0.429. Jumlah topik optimal dipilih berdasarkan nilai *coherence* tertinggi, yaitu pada topik 4.



Gambar 4.19 *Coherence Score* Kota A Positif

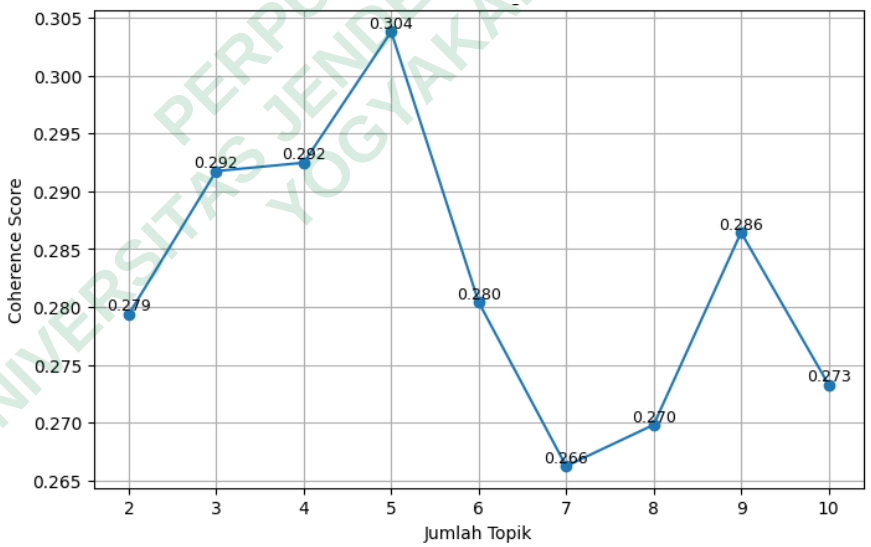
2. Coherence Score Kota A Negatif

Proses yang sama diterapkan pada data negatif Kota A, untuk menemukan jumlah topik optimal dalam ulasan negatif dari RSU X Kota A. Dengan menghitung *coherence score* pada berbagai jumlah topik, model LDA dapat mengidentifikasi pola utama yang muncul dalam ulasan negatif. Hasil **Kode Program 4.14** akan menghasilkan grafik untuk memudahkan interpretasi.

Kode Program 4.14 *Coherence Score* Negatif Kota A

```
1. determine_optimal_topics(tokenized_texts2, "Negative Kota
   A", max_topics=10)
```

Visualisasi yang dihasilkan **Gambar 4.20** menunjukkan perubahan *coherence score* berdasarkan jumlah topik yang diuji untuk ulasan negatif di Kota A. Dari visualisasi grafik ini, dapat diamati bahwa nilai *coherence* memiliki pola naik turun. Nilai tertinggi terdapat di topik 5 dengan score 0.304 dan terendah di topik 7 dengan nilai 0.266. Topik 5 akan digunakan dalam pemodelan LDA guna mendapatkan representasi topik yang sesuai dengan data ulasan negatif di Kota A.



Gambar 4.20 *Coherence Score* Kota A Negatif

3. Coherence Score Kota B Positif

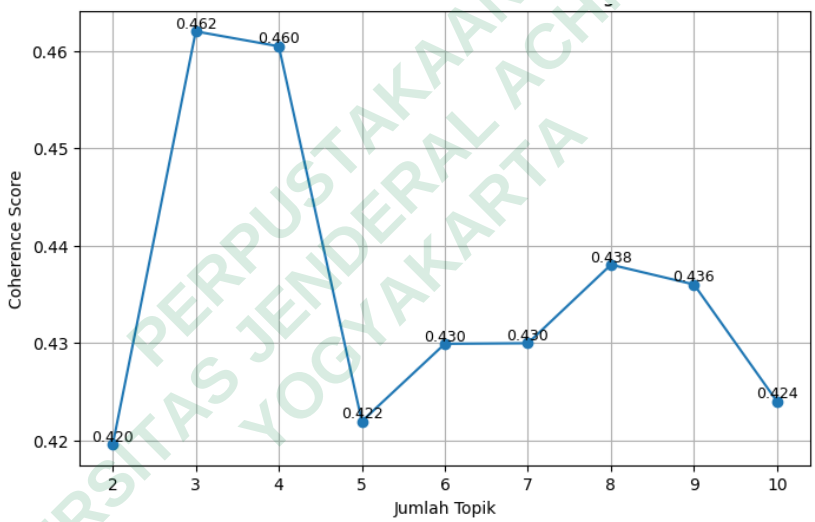
Pengujian untuk data positif Kota B, *function* ini digunakan untuk mengevaluasi jumlah topik optimal dalam ulasan positif dari RSU X Kota B. Dengan pendekatan yang sama, *coherence score* dihitung untuk berbagai jumlah topik, dan hasilnya divisualisasikan dalam bentuk grafik serta tabel. Pengujian

coherence score untuk data ulasan positif dari RSU X Kota B dapat dilihat pada **Kode Program 4.15**.

Kode Program 4.15 *Coherence Score* Positif Kota B

```
1. determine_optimal_topics(tokenized_texts3, "Positive Kota
   B", max_topics=10)
```

Gambar 4.21 menunjukkan visualisasi grafik *coherence score* yang menampilkan hubungan antara jumlah topik yang diuji dengan nilai *coherence* untuk ulasan positif di Kota B. Berdasarkan hasil visualisasi, hasil nilai *coherence* cenderung mengalami penurunan, walaupun terdapat beberapa topik yang naik. Nilai *coherence* tertinggi terdapat di topik 3 dengan *score* sebesar 0.462, yang dijadikan topik optimal di data ulasan positif RSU X Kota B.



Gambar 4.21 *Coherence Score* Kota B Positif

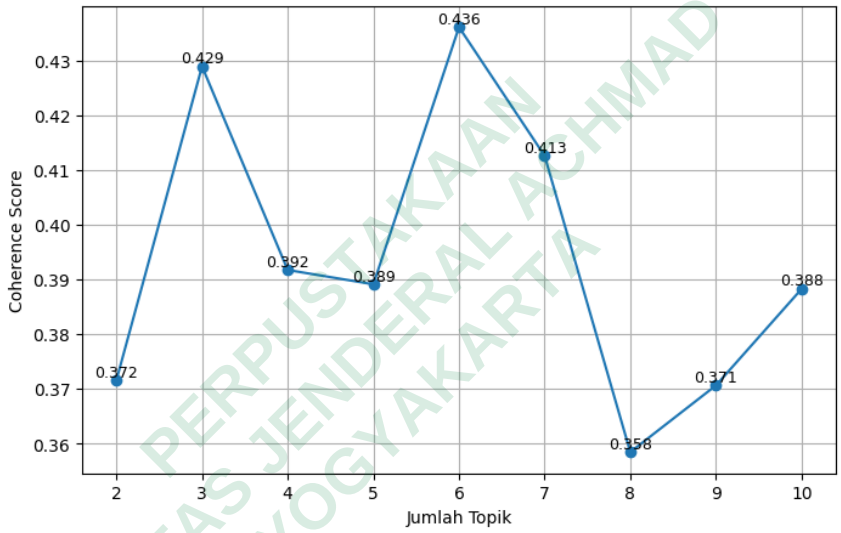
4. Coherence Score Kota B Negatif

Terakhir, untuk data ulasan negatif Kota B, dilakukan perhitungan *coherence score* guna menentukan jumlah topik yang optimal dalam ulasan dengan sentimen negatif. Proses ini memastikan bahwa model LDA dapat menangkap pola utama dalam ulasan negatif di Kota B secara akurat. Kode Program untuk perhitungan *coherence score* data ulasan negatif Kota B dapat ditunjukkan pada **Kode Program 4.16**.

Kode Program 4.16 *Coherence Score* Negatif Kota B

```
1. determine_optimal_topics(tokenized_texts4, "Negative Kota
   B", max_topics=10)
```

Visualisasi grafik pada **Gambar 4.22** menunjukkan perubahan *coherence score* pada berbagai jumlah topik dalam ulasan negatif di Kota B. Dengan melihat pola kenaikan dan penurunan *coherence score*, jumlah topik optimal dapat ditentukan dengan memilih nilai tertinggi yang menunjukkan keseimbangan antara interpretabilitas dan akurasi model. Nilai *coherence* tertinggi terlihat pada topik 6 dengan *score* sebesar 0.436.



Gambar 4.22 *Coherence Score* Kota B Negatif

4.5.3 Model *Latent Dirichlet Allocation* (LDA)

Setelah menentukan jumlah topik optimal untuk keempat data ulasan, langkah selanjutnya adalah menerapkan model LDA untuk mengekstrak topik dari masing-masing data ulasan. Untuk itu, dibuat model LDA berdasarkan jumlah topik optimal yang telah ditentukan sebelumnya. Berdasarkan nilai *coherence score* yang telah dihitung sebelumnya, jumlah topik optimal untuk data ulasan positif RSU X Kota A adalah 4. Pemodelan topik menggunakan LDA diterapkan dengan parameter tersebut untuk mengidentifikasi distribusi topik dalam data ini. Berikut **Kode Program 4.17** merupakan kode program untuk membangun model LDA di data ulasan positif RSU X Kota A.

Kode Program 4.17 Model LDA Positif Kota A

```
1. lda_model1 = gensim.models.LdaModel(
                  corpus=corpus1,id2word=dictionary1,
                  num_topics=4,random_state=42,
                  passes=10,per_word_topics=True)
```

Sama halnya dengan data ulasan positif RSU X Kota A, ulasan negatif juga menerapkan model LDA. Jumlah topik yang digunakan adalah 5 topik berdasarkan hasil perhitungan *coherence score* pada data ulasan negatif. Kode program untuk model LDA ini dapat dilihat pada **Kode Program 4.18**.

Kode Program 4.18 Model LDA Negatif Kota A

```
1. lda_model2 = gensim.models.LdaModel(
                  corpus=corpus2,id2word=dictionary2,
                  num_topics=5,random_state=42,
                  passes=10,per_word_topics=True)
```

Model LDA untuk data ulasan positif RSU X Kota B dapat dilihat pada **Kode Program 4.19**. Jumlah topik optimal yang digunakan dalam model ini berjumlah 3 topik. Jumlah topik yang digunakan sesuai dengan perhitungan *coherence score* sebelumnya.

Kode Program 4.19 Model LDA Positif Kota B

```
1. lda_model3 = gensim.models.LdaModel(
                  corpus=corpus3,id2word=dictionary3,
                  num_topics=3,random_state=42,
                  passes=10,per_word_topics=True)
```

Terakhir, data ulasan negatif RSU X Kota B digunakan untuk *inputan* model LDA. Berdasarkan hasil perhitungan topik optimal dengan *coherence score*, data ulasan negatif ini mendapatkan topik optimal berjumlah 6. Kode Program untuk model LDA dapat dilihat pada **Kode Program 4.20**.

Kode Program 4.20 Model LDA Negatif Kota B

```
1. lda_model4 = gensim.models.LdaModel(
                  corpus=corpus4, id2word=dictionary4,
                  num_topics=6, random_state=42,
                  passes=10, per_word_topics=True)
```

4.6 VISUALISASI

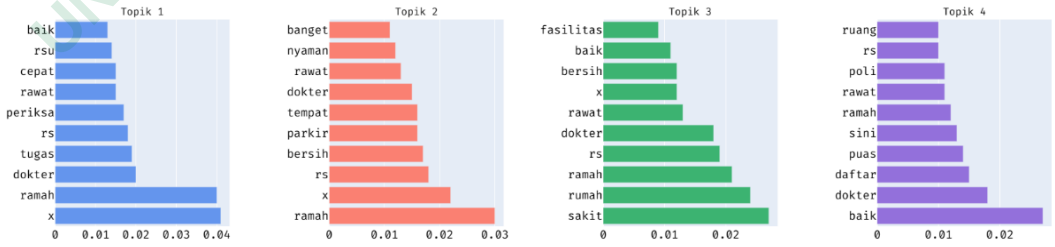
Setelah membangun model LDA, langkah berikutnya adalah melakukan visualisasi memahami topik yang terbentuk. Visualisasi topik dibuat untuk memahami topik yang paling dominan dalam setiap data ulasan. Setiap data ulasan dikategorikan berdasarkan topik dengan probabilitas tertinggi, sehingga dapat diidentifikasi topik yang paling sering muncul dalam data. Visualisasi pyLDAvis memungkinkan analisis distribusi topik dalam dokumen serta hubungan antar topik berdasarkan kemunculan kata-kata yang tumpang tindih. Dengan menggunakan pyLDAvis, dapat diperoleh wawasan lebih dalam mengenai struktur topik yang dihasilkan oleh model LDA. **Kode Program 4. 21** menunjukkan kode program untuk membuat visualisasi pyLDAvis.

Kode Program 4.21 Visualisasi pyLDAvis

```
1. def visualize_lda(lda_model, corpus, dictionary):
2.       pyLDAvis.enable_notebook()
3.       vis = pyLDAvis.gensim.prepare(lda_model, corpus,
         dictionary=dictionary)
4.       display(vis)
```


1. RSU X Kota A (Positif)

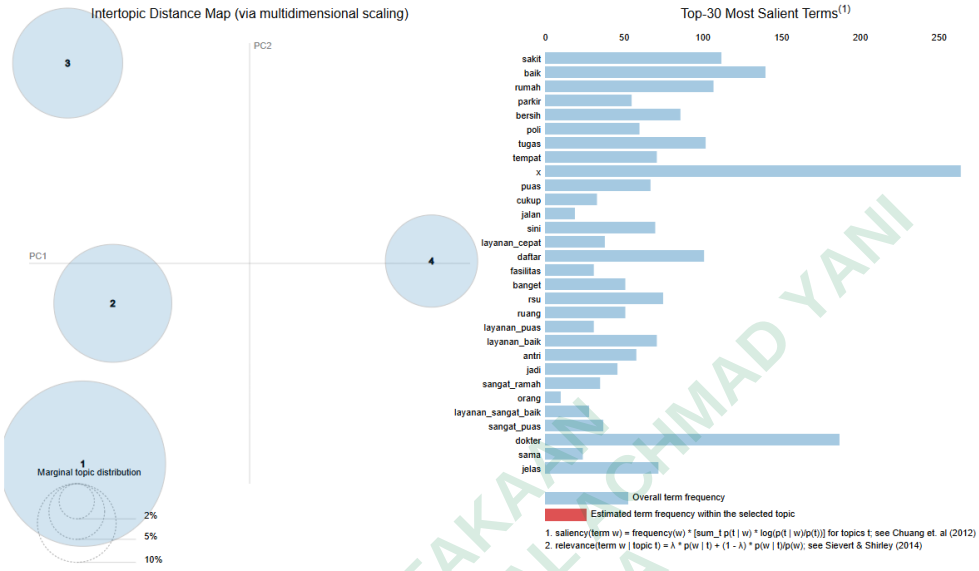
Visualisasi **Gambar 4.23** merupakan hasil pemodelan topik yang menampilkan kata-kata kunci dari masing-masing topik yang terbentuk berdasarkan ulasan positif pengunjung di RSU X Kota A. Data positif tersebut menghasilkan empat topik utama, dimana setiap *bar chart* menunjukkan kata-kata yang paling mewakili topik tersebut beserta proporsi kemunculannya.



Gambar 4.23 Hasil Topik Positif RSU X Kota A

Hasil visualisasi pyLDAvis untuk model LDA ulasan positif Kota A menunjukkan empat topik utama yang terbentuk. Pada *Intertopic Distance Map*, terlihat topik 1 dan 3 sedikit beririsan, serta topik 2 dan 4 cukup berjauhan. Hal

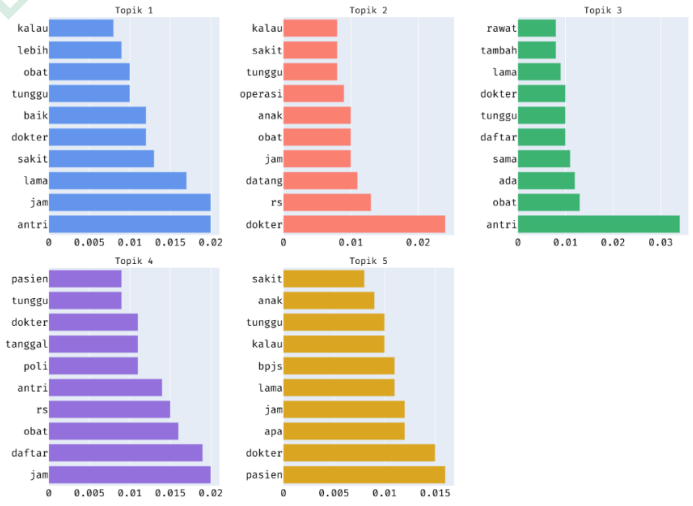
tersebut menandakan bahwa masing-masing topik memiliki karakteristik kata yang berbeda walaupun terdapat sedikit beririsan. Hasil pyLDAvis untuk model LDA ulasan positif Kota A ditunjukkan pada **Gambar 4.24**.



Gambar 4.24 pyLDAvis Positif RSU X Kota A

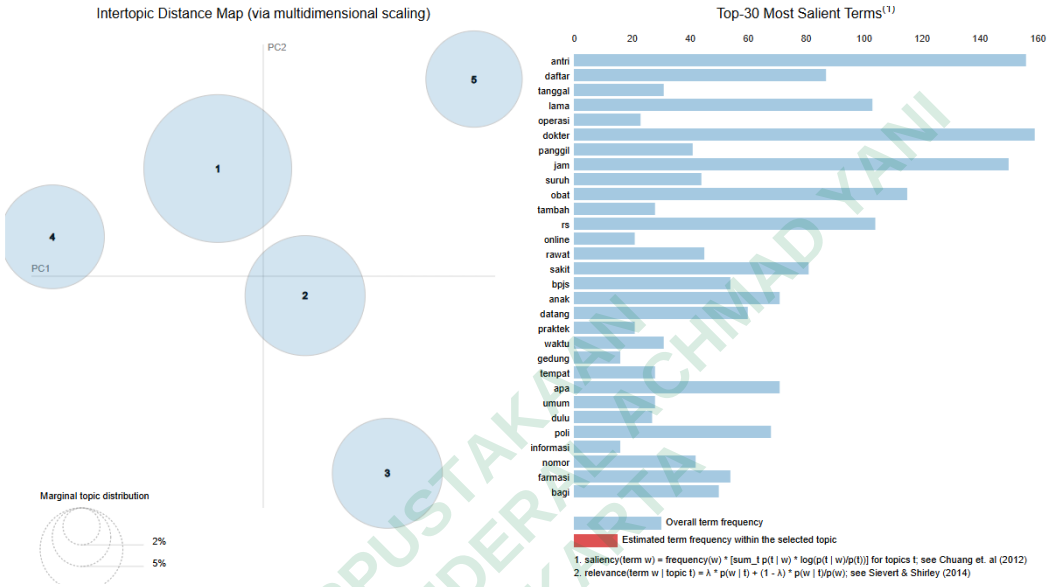
2. RSU X Kota A (Negatif)

Gambar 4.25 menunjukkan hasil pemodelan topik dari ulasan negatif pengunjung terhadap layanan RSU X Kota A. Terdapat 5 topik yang berhasil diidentifikasi. Setiap topik digambarkan dalam bentuk diagram batang horizontal yang memuat kata kunci utama dan distribusinya, mencerminkan isu-isu yang paling banyak dikeluhkan oleh pengunjung.



Gambar 4.25 Topik Negatif RSU X Kota A

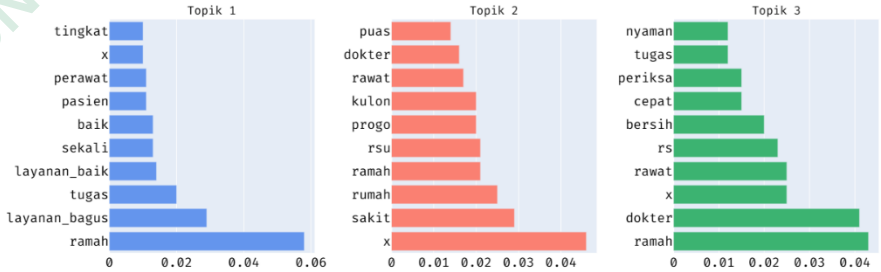
Hasil visualisasi pyLDAvis untuk model LDA pada ulasan negatif RSU X Kota A menghasilkan 6 topik utama. Dalam *Intertopic Distance Map*, terlihat bahwa sebagian besar topik tersebar cukup berjauhan, menunjukkan adanya variasi isu atau keluhan yang disampaikan pengunjung. Hasil pyLDAvis untuk model ini ditampilkan pada **Gambar 4.26**.



Gambar 4.26 pyLDAvis Negatif RSU X Kota A

3. RSU X Kota B (Positif)

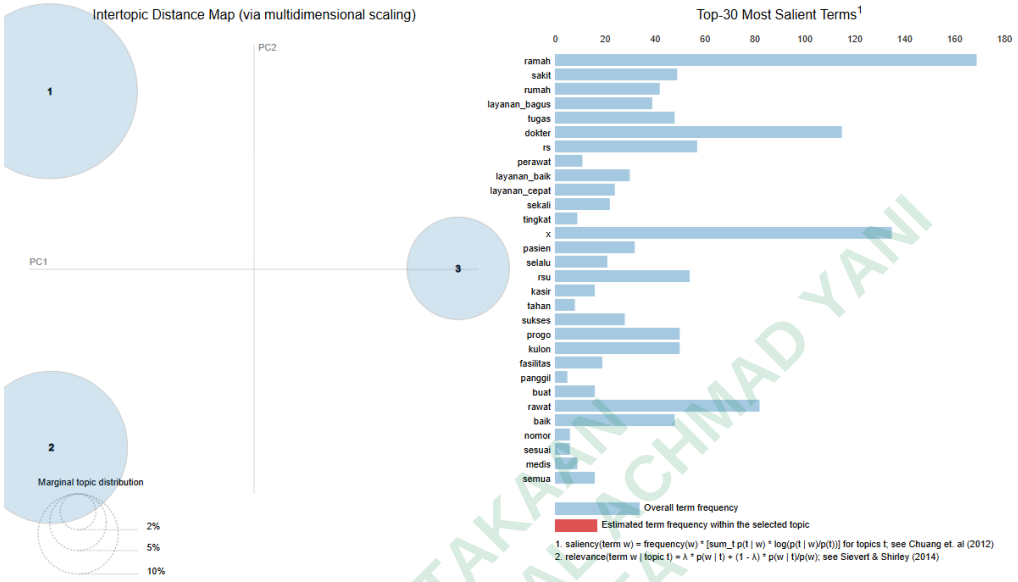
Gambar 4.27 menyajikan hasil visualisasi pemodelan topik dari ulasan positif pengunjung RSU X Kota B. Dari analisis ini terbentuk tiga topik utama, yang masing-masing menampilkan kata-kata kunci dominan dan tingkat kemunculannya dalam data.



Gambar 4.27 Topik Positif RSU X Kota B

Visualisasi pyLDAvis untuk model LDA ulasan positif Kota B menunjukkan tiga topik utama. Dalam *Intertopic Distance Map*, ketiga topik terlihat cukup terpisah satu sama lain, yang mengindikasikan bahwa masing-masing

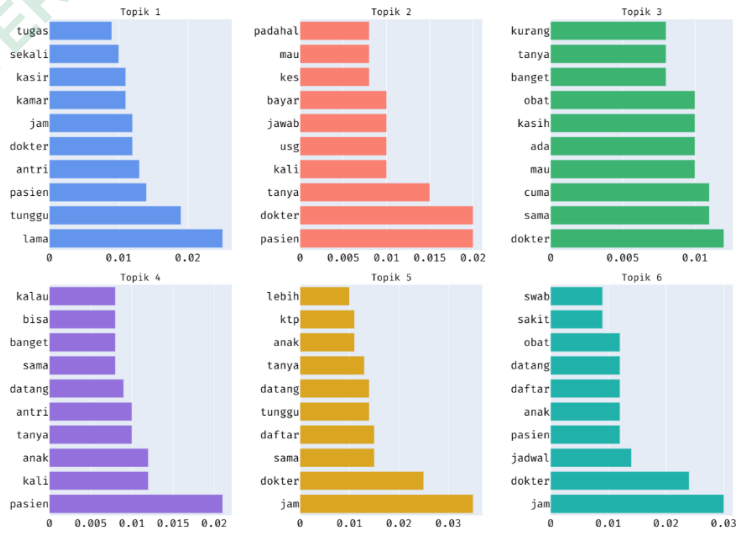
topik memiliki karakteristik kata yang unik. Hal ini mencerminkan keragaman aspek positif yang disampaikan pengunjung. Hasil pyLDAvis ditunjukkan **Gambar 4.28**.



Gambar 4.28 pyLDAvis Positif RSU X Kota B

4. RSU X Kota B (Negatif)

Gambar 4.29 menggambarkan hasil pemodelan topik berdasarkan ulasan negatif dari pengunjung RSU X Kota B. Sebanyak 6 topik berhasil diekstraksi dari data, yang divisualisasikan dalam bentuk *bar chart* dengan kata-kata kunci yang mencerminkan masalah atau keluhan paling sering disampaikan pengunjung.



Gambar 4.29 Topik Negatif RSU X Kota B

Hasil pyLDAvis untuk model LDA pada ulasan negatif pengunjung RSU X Kota B mengungkapkan adanya 6 topik utama yang mencerminkan berbagai masalah yang dialami pengunjung. *Intertopic Distance Map* menunjukkan bahwa topik-topik ini terlihat lebih terpisah satu sama lain, menandakan bahwa masing-masing topik fokus pada isu yang berbeda. Visualisasi yang disajikan pada **Gambar 4.30**, memberikan gambaran yang jelas tentang keragaman keluhan pengunjung dan dapat menjadi dasar untuk perbaikan layanan di rumah sakit. Dengan pemahaman yang lebih mendalam terhadap masalah yang dihadapi pengunjung, rumah sakit dapat mengidentifikasi area yang perlu diperbaiki dan meningkatkan kualitas pelayanan secara keseluruhan.

Gambar 4.30 pyLDAvis Negatif RSU X Kota B

4.7 ANALISIS HASIL

1. Kota A Positif

Hasil analisis pada **Tabel 4.2** RSU X Kota A menyoroti keramahan tenaga medis, kecepatan layanan, serta kenyamanan rawat jalan maupun inap. Kebersihan, kenyamanan fasilitas, dan kemudahan akses seperti parkir turut membentuk citra positif RSU X. Secara keseluruhan, pengunjung merasa puas dengan proses pendaftaran, pelayanan poli, hingga interaksi dengan staf dokter.

Tabel 4.2 Hasil Analisis Positif RSU X Kota A

Topik	Kata Utama	Analisis	Insight
1	x, ramah, dokter, tugas, rs, periksa, rawat, cepat, rsu, baik	Menekankan pada keramahan tenaga medis dan kecepatan layanan periksa serta rawat jalan atau inap.	Pengunjung merasa puas dengan keramahan dokter dan tenaga medis serta kecepatan pelayanan.
2	ramah, x, rs, bersih, parkir, tempat, dokter, rawat, nyaman, banget	Fokus pada aspek non-medis seperti kebersihan, kenyamanan fasilitas, dan ketersediaan parkir yang membentuk kesan positif.	Lingkungan RS yang bersih dan nyaman, serta fasilitas pendukung yang menjadi nilai tambah kepuasan pengunjung.
3	sakit, rumah, ramah, rs, dokter, rawat, x, bersih, baik, fasilitas	Keseluruhan pengalaman perawatan saat sakit, dari keramahan hingga fasilitas dan kebersihan RS.	Kualitas pelayanan yang menyeluruh, mulai dari dokter hingga fasilitas, yang membuat pengunjung merasa nyaman.
4	baik, dokter, daftar, puas, sini, ramah, rawat, poli, rs, ruang	Kepuasan pada sistem pendaftaran, layanan poli, serta keramahan staf dan dokter.	Proses pendaftaran dan layanan poli yang baik meningkatkan kepuasan pengunjung terhadap layanan RS.

RSU X Kota A berhasil membangun citra positif di mata pengunjung berkat kombinasi pelayanan yang ramah, cepat, dan profesional. Banyak pengunjung menyoroti keramahan dokter serta tenaga medis yang membuat proses pengobatan terasa lebih nyaman. Fasilitas rumah sakit yang bersih dan nyaman juga menjadi nilai tambah, didukung dengan sistem pendaftaran dan layanan poli yang tertata rapi, sehingga menciptakan pengalaman berobat yang efisien dan memuaskan.

2. Kota A Negatif

Berdasarkan analisis pada **Tabel 4.3**, pengunjung RSU X Kota A banyak mengeluhkan lamanya waktu tunggu saat antrian, proses pendaftaran, dan pengambilan obat. Masalah juga muncul dalam koordinasi jadwal pelayanan dan ketidakjelasan informasi, terutama bagi pengguna BPJS dan pengunjung anak.

Meski begitu, sebagian tetap menilai dokter bersikap baik meskipun penanganannya dinilai lambat.

Tabel 4.3 Hasil Analisis Negatif RSU X Kota A

Topik	Kata Utama	Analisis	Insight
1	antri, jam, lama, sakit, dokter, baik, tunggu, obat, lebih, kalau	Pengalaman pengunjung terkait antrian dan waktu tunggu lama soal pelayanan, meskipun ada yang menilai dokter baik.	Pengunjung merasa waktu tunggu lama dan antrian tidak efisien, meskipun kualitas dokter diapresiasi.
2	dokter, rs, datang, jam, obat, anak, operasi, tunggu, sakit, kalau	Berkaitan kedatangan pengunjung, penanganan dokter, obat dan prosedur operasi. menunjukkan keluhan terhadap waktu layanan dan penanganan.	Pengunjung membutuhkan informasi yang jelas terkait jadwal dokter dan prosedur layanan, termasuk anak dan operasi.
3	antri, obat, ada, sama, daftar, tunggu, dokter, lama, tambah, rawat	Proses pendaftaran dan pengambilan obat yang lambat.	Alur pendaftaran dan pengambilan obat perlu diperbaiki agar pengunjung tidak mengalami antrian ganda.
4	jam, daftar, obat, rs, antri, poli, tanggal, dokter, tunggu, pasien	Masalah koordinasi jadwal pelayanan, seperti jam buka, poli, dan tanggal yang tidak sesuai dengan informasi yang didapat pengunjung.	Sistem penjadwalan dan informasi layanan perlu transparan dan konsisten untuk menghindari ketidaksesuaian antara informasi dan kenyataan.
5	pasien, dokter, apa, jam, lama, bpjs, kalau, tunggu, anak, sakit	Ketidakpastian informasi layanan, khususnya pengguna BPJS dan pengunjung anak, serta waktu tunggu yang lama.	pengunjung membutuhkan komunikasi yang lebih baik dari pihak RS mengenai prosedur dan layanan BPJS.

Berdasarkan *insight*, ada beberapa area yang memerlukan perhatian lebih dari RSU X Kota A untuk meningkatkan kualitas pelayanan. Pengunjung mengeluhkan waktu tunggu yang lama dan antrian yang tidak efisien, meskipun kualitas dokter tetap mendapat apresiasi. Mereka juga menginginkan informasi yang lebih jelas mengenai jadwal dokter serta prosedur layanan, terutama terkait dengan pengunjung anak dan prosedur operasi. Alur pendaftaran dan pengambilan obat perlu diperbaiki agar pengunjung tidak terjebak dalam antrian ganda. Sistem penjadwalan dan informasi layanan harus lebih transparan dan konsisten, agar tidak ada ketidaksesuaian antara apa yang diinformasikan dengan kenyataan di lapangan. Selain itu, pengunjung berharap ada komunikasi yang lebih baik dari pihak rumah sakit mengenai prosedur layanan BPJS dan informasi terkait layanan lainnya.

3. Kota B Positif

Hasil analisis ulasan positif RSU X Kota B pada **Tabel 4.4** menunjukkan pengunjung RSU X Kota B secara umum merasa puas terhadap pelayanan, terutama karena keramahan staf dan dokter, serta kualitas layanan rawat inap maupun rawat jalan. Kebersihan fasilitas, kecepatan pemeriksaan, dan kenyamanan lingkungan juga turut membentuk kesan positif terhadap rumah sakit.

Tabel 4.4 Hasil Analisis Positif RSU X Kota B

Topik	Kata Utama	Analisis	Insight
1	ramah, layanan_bagus, tugas, layanan_baik, sekali, baik, pasien, perawat, x, tingkat	Penilaian tinggi terhadap keramahan staf dan kualitas layanan secara umum, terutama yang diberikan perawat dan staf.	Pengunjung menghargai keramahan dan kualitas layanan dari perawat dan staf, yang dinilai bagus dan baik.
2	x, sakit, rumah, ramah, rsu, progo, kulon, rawat, dokter, puas	Pengalaman rawat inap atau jalan dengan sentimen puas terhadap layanan selama sakit, serta keramahan dokter.	RSU X Kota B tempat yang baik untuk perawatan, dengan layanan yang memuaskan, dan keramahan dokter.
3	ramah, dokter, x, rawat, rs, bersih,	Kombinasi keramahan, kebersihan, dan	Pelayanan cepat, lingkungan bersih, serta dokter yang

Topik	Kata Utama	Analisis	Insight
	cepat, periksa, tugas, nyaman	kecepatan pemeriksaan, dan kenyamanan secara keseluruhan.	ramah membuat pengunjung merasa nyaman.

Pengunjung sangat menghargai keramahan dan kualitas layanan dari perawat dan staf di RSU X Kota B, yang secara konsisten dinilai bagus dan baik. Rumah sakit ini dianggap sebagai tempat yang sangat baik untuk perawatan, dengan layanan yang memuaskan dan keramahan dokter yang meningkatkan kenyamanan pengunjung. Pelayanan yang cepat, lingkungan yang bersih, serta sikap ramah dokter membuat pengunjung merasa sangat nyaman dan diterima selama perawatan mereka.

4. Kota B Negatif

Keluhan negatif dari pengunjung RSU X Kota B pada **Tabel 4.5** mengeluhkan lamanya waktu tunggu di berbagai layanan seperti antrian, dokter, kamar, dan kasir. Komunikasi dari dokter dinilai kurang jelas, terutama terkait prosedur medis dan pembayaran. Beberapa pengunjung merasa hanya diberi obat tanpa penjelasan memadai. Ada juga keluhan dari orang tua pengunjung anak terkait lambatnya pelayanan dan kurangnya respons terhadap pertanyaan. Selain itu, informasi jadwal dokter sering tidak sesuai, dan proses administrasi seperti pendaftaran dan persyaratan KTP membingungkan.

Tabel 4.5 Hasil Analisis Negatif RSU X Kota B

Topik	Kata Utama	Analisis	Insight
1	lama, tunggu, pasien, antri, dokter, jam, kamar, kasir, sekali, tugas	Keluhan pengunjung terkait waktu tunggu berbagai layanan, seperti antrian, pelayanan dokter, hingga proses kamar dan kasir.	Waktu tunggu yang panjang dan alur layanan yang lambat menjadi sumber ketidakpuasan pengunjung.
2	pasien, dokter, tanya, kali, usg, jawab,	Minimnya komunikasi dan kurangnya penjelasan dari dokter	Kurangnya kejelasan dan komunikasi dari dokter serta

Topik	Kata Utama	Analisis	Insight
	bayar, kes, mau, padahal	terkait tindakan medis, seperti prosedur USG dan proses pembayaran.	transparasi biaya mempengaruhi pengalaman pengunjung secara negatif.
3	dokter, sama, cuma, mau, ada, kasih, obat, banget, tanya, kurang	Keluhan kepada dokter yang hanya memberikan obat tanpa pemeriksaan atau penjelasan yang memadai.	Pelayanan yang terkesan buru-buru dan minim interaksi membuat pengunjung merasa tidak mendapatkan perawatan yang baik.
4	pasien, kali, anak, tanya, antri, datang, sama, banget, bisa, kalau	Keluhan dari orang tua pengunjung anak mengenai panjangnya antrian, lambatnya pelayanan, dan kurangnya perhatian terhadap pertanyaan.	Pengunjung anak dan keluarganya mengalami ketidaknyamanan karena waktu tunggu dan komunikasi yang kurang responsif.
5	jam, dokter, sama, daftar, tunggu, datang, tanya, anak, ktp, lebih	Ketidaksesuaian antara jadwal pelayanan dokter dengan kenyataan saat datang, termasuk kendala administratif seperti KTP.	Ketidaksesuaian jadwal dan sistem pendaftaran yang belum efisien menganggu kelancaran akses layanan bagi pengunjung.
6	jam, dokter, jadwal, pasien, anak, daftar, datang, obat, sakit, swab	Ketidakjelasan informasi mengenai jadwal dokter dan layanan seperti swab test, serta proses pendaftaran yang membingungkan.	Ketidakpastian informasi layanan dan buruknya koordinasi jadwal berdampak ketidaknyamanan pengunjung.

Beberapa aspek di RSU X Kota B memerlukan perbaikan untuk meningkatkan kepuasan pengunjung. Pengunjung di RSU X Kota B seringkali merasa tidak puas dengan waktu tunggu yang panjang dan alur layanan yang lambat, yang menjadi sumber ketidaknyamanan. Kurangnya kejelasan dan komunikasi dari dokter, serta transparansi biaya, mempengaruhi pengalaman pengunjung secara negatif, menciptakan rasa frustrasi dan ketidakpastian.

Pelayanan yang terkesan buru-buru dan minim interaksi membuat pengunjung merasa tidak mendapatkan perhatian dan perawatan yang optimal.

Untuk pengunjung anak dan keluarganya, ketidaknyamanan semakin meningkat akibat waktu tunggu yang lama dan komunikasi yang kurang responsif. Ketidaksesuaian jadwal dan sistem pendaftaran yang belum efisien juga mengganggu kelancaran akses layanan. Secara keseluruhan, ketidakpastian informasi layanan dan buruknya koordinasi jadwal memperburuk pengalaman pengunjung dan menurunkan kepuasan mereka terhadap rumah sakit. Perbaikan dalam hal komunikasi, penjadwalan, dan efisiensi layanan sangat diperlukan untuk meningkatkan pengalaman pengunjung di RSU X Kota B.