# PENDAHULUAN

## Latar Belakang

## Tujuan

# LANDASAN TEORI

## Studi Pustaka

# METODE PENELITIAN

## Kerangka Pemikiran

Kerangka berpikir adalah gambaran yang menggambarkan bagaimana penelitian akan berlangsung secara logis dan menyeluruh. Tahapan-tahapan dalam kerangka berpikir penelitian meliputi antara lain:

1. Pengumpulan Data

Penelitian ini menggunakan data dari platform *e-commerce* Tokopedia. Data yang diambil adalah riwayat ulasan pelanggan yang pernah berbelanja di toko Hanafashion\_shop, yang diperoleh melalui teknik web scraping.

1. Pelabelan

Pelabelan adalah tahap di mana setiap ulasan diberi label yang akan digunakan dalam proses pelatihan pada tahap klasifikasi. Atribut ulasan berisi tentang pengalaman pelanggan terkait kepuasan mereka saat berbelanja di toko Hanafashion\_shop. Pelabelan dilakukan berdasarkan nilai rating produk, di mana rating 1-3 dikategorikan sebagai negatif, sementara rating 4-5 dikategorikan sebagai positif.

1. Preprocessing

Adapun tahapan yang dilakukan pada preprocessing yaitu sebagai berikut :

1. *Cleaning*, yaitu untuk menghapus data yang memiliki nilai yang sama *(duplicate)* dan data yang kosong (*nan)*.
2. *Normalize*, yaitu untuk mengoreksi kata-kata yang salah ketik atau singkatan agar kembali ke bentuk aslinya.
3. *Stopword*, yaitu untuk menghapus kata-kata umum yang tidak memiliki informasi penting untuk analisis dari teks.
4. *Tokenizing*, yaitu untuk melakukan pemenggalan pada tiap suku kata.
5. *Stemming*, yaitu untuk mengubah kata-kata menjadi bentuk dasarnya dengan menghapus akhiran atau imbuhan.
6. Klasifikasi

Proses klasifikasi dibedakan menjadi dua proses, diantaranya :

* 1. *Training* adalah proses melatih algoritma klasifikasi, yakni *Support Vector Machine* dan *IndoBert*, agar dapat berfungsi sesuai harapan. Pertama, data atribut ulasan diberi bobot menggunakan perhitungan TF-IDF, namun hanya memperhitungkan frekuensi istilah *(term frequency)*. Proses ini menghasilkan model klasifikasi yang kemudian digunakan dalam tahapan *testing*.
  2. *Testing* adalah tahap di mana dataset diklasifikasikan dengan menggunakan model klasifikasi yang dihasilkan selama proses *training* data. Pada tahap ini, ulasan dikategorikan ke dalam sentimen positif dan negatif. Langkah-langkah proses ini dapat dilihat pada gambar di bawah.

A screenshot of a cell phone

Description automatically generated

**Gambar 3. 1** Kerangka Berpikir

## Bahan/Data

### Prosedur Pengumpulan Data

Penelitian ini menggunakan metode scraping menggunakan Selenium untuk mengumpulkan data ulasan toko Hanafashion\_shop di situs web Tokopedia. Metode ini memungkinkan pengendalian browser untuk mengakses halaman toko Hanafashion\_shop, mengambil ulasan pelanggan, dan mengumpulkan data secara otomatis. Dengan demikian, penelitian ini dapat menganalisis ulasan pelanggan dengan efisien dan mendalam untuk memperoleh wawasan yang relevan terkait dengan pengalaman pelanggan di toko Hanafashion\_shop di Tokopedia. Tampilan flowchart pengumpulan data dan scraping data Tokopedia dapat dilihat pada gambar di bawah.

A screenshot of a computer

Description automatically generated

**Gambar 3. 2** Produk Hanafashion\_shop

Selanjutnya, Selenium akan berinteraksi dengan halaman toko Hanafashion\_shop di situs web Tokopedia untuk mencari ulasan yang relevan. Proses ini mungkin melibatkan mengklik pada halaman produk yang tepat, menavigasi melalui halaman ulasan pelanggan, dan mengumpulkan data ulasan yang ada. Selenium akan mengekstrak teks ulasan dari elemen HTML yang sesuai dan menyimpannya dalam format csv untuk preprocessing lebih lanjut. Langkah-langkah tambahan seperti pembersihan data, penghapusan karakter khusus, atau penggabungan ulasan yang terpisah juga dapat dilakukan untuk memastikan data siap untuk analisis sentimen.

### Data yang diperoleh

Penelitian ini menggunakan data ulasan toko Hanafashion\_shop yang diambil melalui platform Tokopedia sebagai sumber data primer. Data ulasan pelanggan yang dikumpulkan dari toko Hanafashion\_shop di Tokopedia akan digunakan sebagai dataset utama dalam penelitian ini. Dengan menggunakan data ulasan yang diperoleh secara langsung dari toko Hanafashion\_shop, penelitian ini dapat memberikan wawasan yang lebih spesifik dan relevan terkait dengan pengalaman pelanggan dalam berinteraksi dengan produk-produk yang ditawarkan oleh toko tersebut. Contoh flowchart dan hasil pengumpulan data dapat dilihat pada gambar di bawah.

A black background with white squares

Description automatically generated

**Gambar 3. 3** Flowchart Pengumpulan Data



**Gambar 3. 4** Data Ulasan Tokopedia

Data yang diambil setiap kali melakukan preprocessing berjumlah maksimal 200 baris data.  
A screenshot of a computer code

Description automatically generated

**Gambar 3. 5** Code Scraping Data

Pengguna perlu memasukkan URL sumber produk Hanafashion\_shop dari platform Tokopedia, serta jumlah data yang ingin di-scrape. Selain itu, pengguna perlu menentukan kisaran rating untuk ulasan produk yang akan di-scrape. Rentang rating ini akan digunakan untuk menentukan label sentimen ulasan, rating 1-3 dianggap negatif, sedangkan rating 4-5 dianggap positif. Setelah semua input dimasukkan, pengguna dapat mengklik tombol untuk memulai proses scraping data.

### Labeling

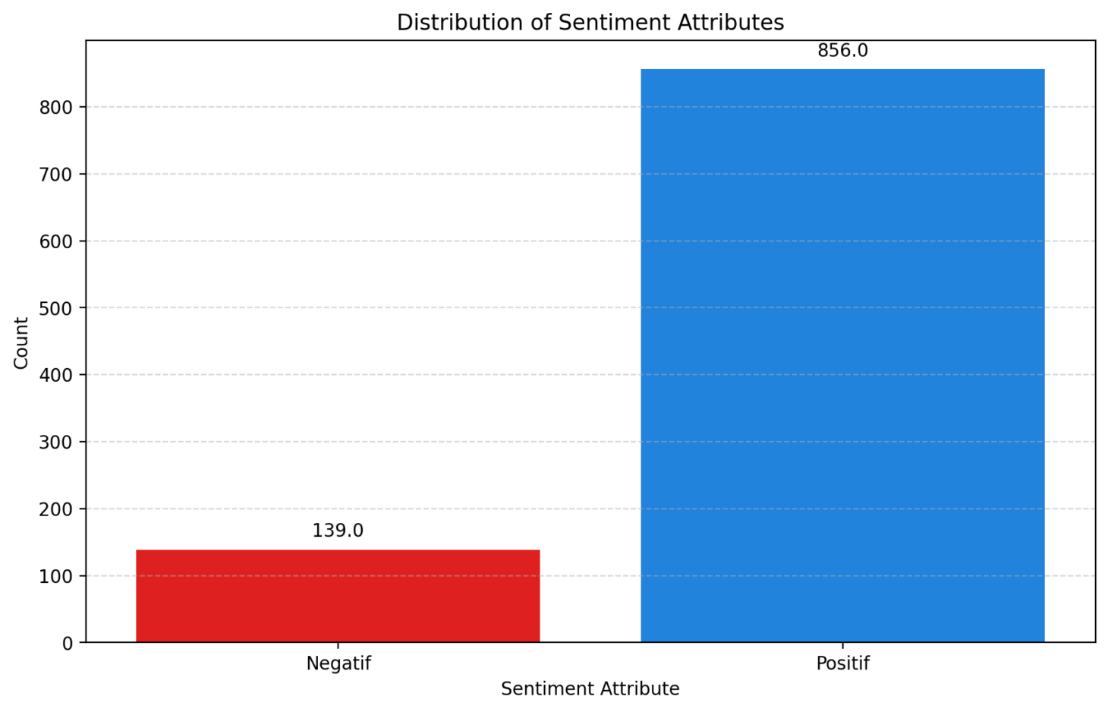
Pelabelan adalah tahap di mana ulasan produk diberi label yang nantinya akan digunakan untuk melatih model dalam proses klasifikasi. Ulasan tersedia untuk 10 produk dengan ulasan terbanyak di toko Hanafashion\_shop. Labeling didasarkan pada rating yang diberikan oleh pelanggan: rating 1-3 dianggap ulasan negatif, sedangkan rating 4-5 dianggap ulasan positif. Kelemahan pelabelan berdasarkan rating adalah adanya subjektivitas dalam hasilnya. Pada tahap pelabelan, ulasan diberi label sentimen positif dan negatif. Kode program untuk pelabelan dapat dilihat pada gambar di bawah ini.

A screenshot of a computer program

Description automatically generated

**Gambar 3. 6** Code Labeling Data

Dari 995 data yang telah diambil dari web scraping sebelumnya, dilakukan pelabelan berdasarkan rating dan menghasilkan sentiment positif dan negatif sebagai berikut :



**Gambar 3. 7** Visualisasi Sentimen

### Cleaning

Dalam proses *cleaning* data, tahap awal adalah mengidentifikasi dan menghapus nilai yang terduplikasi. Selain itu, langkah penting lainnya adalah menangani nilai yang hilang (NaN) agar data menjadi lebih bersih dan siap digunakan dalam pelatihan model SVM dan indoBert, sehingga kinerja model dapat dioptimalkan. Adapun *flowchart* dan kode program dari cleaning data dapat dilihat pada gambar di bawah.



**Gambar 3. 8** Flowchart Cleaning Data



**Gambar 3. 9** Code Cleaning Data

### Normalize

Pada tahapan *normalize* data, fokus utamanya adalah memperbaiki kata-kata yang mungkin disingkat atau tertulis dengan typo seperti "seller", "dll", "bgos", dan sejenisnya. Proses ini bertujuan untuk memastikan konsistensi dalam penggunaan kata-kata sehingga data menjadi lebih mudah dipahami dan diolah. Adapun contoh dari tahap *normalize* dapat dilihat pada table di bawah.

**Tabel 3. 1** Contoh Normalize

|  |  |
| --- | --- |
| **Text Input** | **Text Output** |
| seller responsif brg ny ok bangettt | penjual responsif, barang nya oke banget |

Normalisasi data adalah tahap penting dalam pra-pemrosesan data yang bertujuan untuk menghasilkan konsistensi dalam representasi data. Dengan melakukan normalisasi, kita dapat menghindari ambiguitas dan meningkatkan akurasi analisis data. Selain itu, normalisasi membantu dalam menyediakan input yang lebih baik untuk model pembelajaran mesin, seperti SVM dan indoBert, karena mengurangi variasi dalam data yang mungkin membingungkan model. Dengan demikian, normalisasi data membantu meningkatkan kualitas dan konsistensi data, yang pada gilirannya dapat meningkatkan kinerja model yang dibangun berdasarkan data tersebut.



**Gambar 3. 10** Code Normalize

### Stopword

Pada tahap ini, dilakukan penghapusan kata-kata yang kurang relevan atau sering muncul dalam teks, yang dikenal sebagai stopword. Stopword ini mencakup kata-kata penghubung dan kata-kata keterangan umum yang tidak memberikan kontribusi signifikan terhadap makna atau inti teks, seperti "sebuah", "oleh", "pada", dan lainnya. Proses ini bertujuan untuk menyederhanakan representasi teks, memfokuskan pada kata-kata kunci, dan mengurangi dimensi yang tidak perlu dalam analisis data, sehingga memungkinkan model untuk lebih fokus pada informasi yang penting. Adapun contoh dari tahap *stopword* dapat dilihat pada table di bawah.

**Tabel 3. 2** Contoh Stopword

|  |  |
| --- | --- |
| **Hasil *Normalize*** | **Hasil *Stopword*** |
| penjual responsif, barang nya oke banget | penjual responsif, barang oke banget |

A screen shot of a computer code

Description automatically generated

**Gambar 3. 11** Code Stopword

**Tabel 3. 3** List Stopword

|  |  |  |  |
| --- | --- | --- | --- |
| **Kata** | **Kata** | **Kata** | **Kata** |
| adalah  adanya  adapun  agak  agaknya  agar  akan  akankah  akhir  akhiri  akhirnya  aku  akulah  amat  amatlah  anda  andalah  antar  antara  antaranya  apa  apaan  apabila  apakah  apalagi  apatah  artinya  asal  asalkan  atas  atau  ataukah  ataupun  awal  awalnya  bagai  bagaikan  bagaimana  bagaimanakah  bagaimanapun  bagi  bagian  bahkan  bahwa  bahwasanya  baik  bakal  bakalan  balik | banyak  bapak  baru  bawah  beberapa  begini  beginian  beginikah  beginilah  begitu  begitukah  begitulah  begitupun  bekerja  belakang  belakangan  belum  belumlah  benar  benarkah  benarlah  berada  berakhir  berakhirlah  berakhirnya  berapa  berapakah  berapalah  berapapun  berarti  berawal  berbagai  berdatangan  beri  berikan  berikut  berikutnya  berjumlah  berkali-kali  berkata  berkehendak  berkeinginan  berkenaan  berlainan  berlalu  berlangsung  berlebihan  bermacam  bermacam-macam | bermula  bersama  bersama-sama  bersiap  bersiap-siap  bertanya  bertanya-tanya  berturut  berturut-turut  bertutur  berujar  berupa  besar  betul  betulkah  biasa  biasanya  bila  bilakah  bisa  bisakah  boleh  bolehkah  bolehlah  buat  bukan  bukankah  bukanlah  bukannya  bulan  bung  cara  caranya  cukup  cukupkah  cukuplah  cuma  dahulu  dalam  dan  dapat  dari  daripada  datang  dekat  demi  demikian  demikianlah  dengan | di  dia  diakhiri  diakhirinya  dialah  diantara  diantaranya  diberi  diberikan  diberikannya  dibuat  dibuatnya  didapat  didatangkan  digunakan  diibaratkan  diibaratkannya  diingat  diingatkan  diinginkan  dijawab  dijelaskan  dijelaskannya  dikarenakan  dikatakan  dikatakannya  dikerjakan  diketahui  diketahuinya  dikira  dilakukan  dilalui  dilihat  dimaksud  dimaksudkan  dimaksudkannya  dimaksudnya  diminta  dimintai  dimisalkan  dimulai  dimulailah  dimulainya  dimungkinkan  dini  dipastikan  diperbuat  diperbuatnya  dipergunakan |

### Tokenizing

Pada tahap *tokenizing*, dokumen diubah menjadi serangkaian term dengan menghapus semua karakter tanda baca yang ada pada token. Tujuan dari proses ini adalah untuk menghasilkan kumpulan kata-kata yang merupakan representasi dari teks atau dokumen tersebut. Dengan demikian, output yang dihasilkan adalah sekumpulan kata-kata yang membentuk inti dari teks, tanpa kehadiran tanda baca yang mungkin tidak relevan dalam analisis atau pemrosesan berikutnya. Adapun contoh dari tahap *tokenizing* dapat dilihat pada table di bawah.

**Tabel 3. 4** Contoh Tokenizing

|  |  |
| --- | --- |
| **Hasil *Stopword*** | **Hasil *Tokenizing*** |
| penjual responsif, barang oke banget | penjual | responsif | barang | oke | banget |

*Tokenizing* adalah proses memecah teks menjadi unit kata. Ini dilakukan dengan menggunakan karakter whitespace seperti enter, tabulasi, dan spasi sebagai pemisah kata. Namun, karakter tunggal seperti tanda kutip tunggal ('), titik (.), semikolon (;), titik dua (:), dan lainnya juga dapat berperan sebagai pemisah kata, tergantung pada konteksnya. Proses tokenisasi ini penting untuk mempersiapkan teks agar dapat diolah lebih lanjut, dengan menghasilkan kumpulan kata-kata yang mewakili teks tersebut dengan tepat.



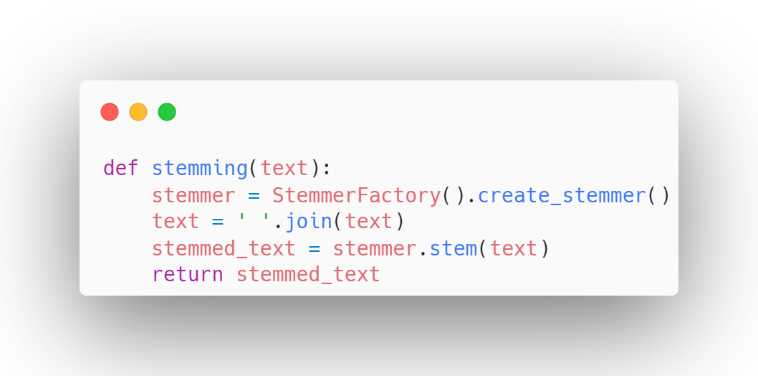
**Gambar 3. 12** Code Tokenizing

### Stemming

*Stemming* merupakan tahap dalam pra-pemrosesan teks yang bertujuan untuk menghapus imbuhan, awalan, dan akhiran dari kata-kata guna mengubahnya menjadi bentuk dasarnya. Proses ini membantu dalam menghasilkan representasi yang lebih konsisten dari kata-kata dalam teks, memungkinkan model untuk lebih mudah mengenali dan memahami makna kata-kata yang sebenarnya. Adapun contoh dari tahap *stemming* dapat dilihat pada table di bawah.

**Tabel 3. 5** Contoh Stemming

|  |  |
| --- | --- |
| **Hasil *Tokenizing*** | **Hasil *Stemming*** |
| penjual | responsif | barang | oke | banget | jual | responsif | barang | oke | banget |



**Gambar 3. 13** Code Stemming

### Pembobotan Kata

Dalam klasifikasi sentiment ulasan tokopedia, pembobotan kata digunakan untuk mendapatkan suatu kategori. Salah satu metode pembobotan adalah TF-IDF *(Term Frequency –Inverse Document Frequency)*.

*Term Weighting* TI-IDF adalah salah satu pembobotan yang sering digunakan dan merupakan gabungan dari *Term Frequency* *dan Inverse Document Frequency*. TF-IDF terdiri dari frekuensi term dan inverse dokumen yang didapatkan dari membagi seluruh jumlah dokumen terhadap jumlah dokumen yang memiliki term tersebut. Dalam TI-IDF bobot akan ditemukan dalam persamaan berikut:

|  |  |  |
| --- | --- | --- |
|  |  | (3. 1) |

Keterangan:

{tf}\_{i,j} = Bobot dari istilah i dalam dokumen j

{df}\_i = Frekuensi munculnya istilah i dalam dokumen j

N = Total istilah pada dokumen

### Support Vector Machine

*Support Vector Machine* (SVM) adalah teknik yang digunakan untuk melakukan prediksi, baik pada kasus klasifikasi maupun regresi. Prinsip dasar SVM adalah pemisah linier, yaitu kemampuan untuk mengklasifikasikan data yang secara linier dapat dipisahkan. Namun, SVM telah dikembangkan untuk bekerja pada masalah non-linier dengan mengadopsi konsep kernel, memungkinkan model bekerja dalam ruang berdimensi tinggi [2].

Pada Gambar 3.11, dijelaskan konsep dasar algoritma *Support Vector Machine* (SVM), yang berfokus pada mencari hyperplane terbaik sebagai pemisah antara dua kelas dalam sebuah data.

*A diagram of a graph

Description automatically generated with medium confidence*

**Gambar 3. 14** Klasifikasi Linear SVM

Pada input space terdapat dua kelas yang berbeda, +1 dan -1, beserta masing-masing pattern yang digambarkan dengan simbol kotak warna orange untuk pattern -1 dan simbol lingkaran hijau untuk pattern +1.

A black background with orange and green dots

Description automatically generated

**Gambar 3. 15** Detail Klasifikasi Linear SVM

­Pada gambar 3.7 dijelaskan dalam mengklasifikasi untuk mendapat hasil yang baik hyperplane digunakan untuk memisahkan menjadi dua kelas dengan mengukur margin *hyperplane* tersebut dan mencari titik maksimalnya. Margin adalah jarak antara hyperplane terdekat dengan pattern terdekat dari masing-masing kelas dan pattern yang paling dekat dengan *hyperplane* disebut *support vector*. Seperti gambar dibawah garis tidak putus-putus yangterletak tepat di tengah-tengah kedua keas. Sedangkan *support vector* tampak sebagai *pattern* yang berpotongan dengan garis putus-putus. Dari Gambar 3.6 bidang pemisah dapat dirumuskan :

m = jarak antara dua bidang

w = bidang normal

b = posisi relative terhadap origin

jarak garis dirumuskan wx+b=c ke origin adalah (c-b)/|w|

|  |  |  |
| --- | --- | --- |
|  |  | (3. 2) |

Margin m dimaksimalkan dengan memenuhi konstrain 2 bidang pembatas yang sejajar dan data yang ada pada bidang pembatas disebut *support vector.* Bidang pembatas kelas pertama membatasi kelas pertama sedangkan bidang pembatas kelas kedua membatasi kelas kedua. Sehingga diperoleh :

|  |  |  |
| --- | --- | --- |
|  | *i.w* + *b*  ≥ + 1 *for yi* = +1 | (3. 3) |
|  | *i.w* + *b*  ≥ + 1 *for yi* = -1 | (3. 4) |

Nilai maksimal margin harus memenuhi rumus di atas dan nilai b dan w dikalikan dengan sebuah konstanta yang akan menghasilkan nilai margin yang dikalikan dengan konstanta yang sama. Konstrain merupakan scaling constraint dengan dipenuhi rescaling b dan w. Karena maksimalkan dan minimalkan w dirumuskan dengan pertidaksamaan rumus di atas.

|  |  |  |
| --- | --- | --- |
|  | *yi* ( *i.w* + *b*) - 1 ≥ 0 | (3. 5) |

Dengan mengalikan b dan sebuah konstanta, maka menghasilkan nilai m kemudian dikalikan dengan konstanta yang sama. Konstrain merupakan scaling constraint yang dipenuhi dengan rescaling b dan w. Maksimalkan minimumkan .

Untuk mencari nilai margin terbesar untuk bidak pemisah terbaik dapat dirumuskan menjadi masalah optimasi konstrain, yaitu :

|  |  |  |
| --- | --- | --- |
|  |  | (3. 6) |

|  |  |  |
| --- | --- | --- |
|  | *s.t yi(* ≥ 0 | (3. 7) |

Dengan lebih mudah untuk menyelesaikan permasalah optimasi konstrain dalam formulasinya dirubah kedalam formula lagrangian yang menggunakan lagrange multiplier yang diubah menjadi :

|  |  |  |
| --- | --- | --- |
|  |  | (3. 8) |

Formula pencarian bidang pemisah terbaik ini adalah permasalahan *quadratic programming,* sehingga nilai maksimum global dari akan selalu dapat ditemukan setelah solusi permasalahan *quadratic programming* ditemukan (nilai ), maka kelas dari data pengujian x dapat ditentukan berdasarkan nilai dari fungsi keputusan :

|  |  |  |
| --- | --- | --- |
|  |  | (3. 9) |

= *support vector,*

Ns = jumlah *support vector*

*=* data yang akan diklasifikasikan

### IndoBert

Model word embedding yang telah dilatih sebelumnya, dikenal sebagai model pre-trained word embedding, dirancang untuk meningkatkan pemahaman makna dan sintaksis dari teks. Model-model ini dilatih menggunakan dataset besar yang beragam agar dapat mengenali pola bahasa secara luas. Pada tahun 2018, model canggih bernama Bidirectional Encoder Representations from Transformers (BERT) diperkenalkan dan berhasil mencapai hasil unggul dalam berbagai penelitian di bidang pemrosesan bahasa alami (NLP). BERT memanfaatkan arsitektur Transformer dengan mekanisme self-attention untuk memahami konteks hubungan antara kata-kata dalam teks. Di Indonesia, perkembangan signifikan terjadi pada tahun 2020 dengan hadirnya model pre-trained BERT yang dikenal sebagai IndoBERT. Model ini khusus disesuaikan untuk bahasa Indonesia, memungkinkan pemahaman yang lebih baik terhadap bahasa lokal dan meningkatkan performa model dalam tugas-tugas NLP berbahasa Indonesia [1].

Model ini bertujuan untuk meminimalkan gabungan fungsi kerugian dari Masked LM dan Next Sentence Prediction, sehingga menghasilkan model bahasa yang kuat dengan kemampuan yang ditingkatkan dalam memahami konteks dalam kalimat dan hubungan antar kalimat. Ada beberapa hal pada arsitektur BERT, diantaranya :

* BERT BASE memiliki 1 2 lapisan di tumpukan Encoder sedangkan BERT LARGE memiliki 24 lapisan di tumpukan Encoder . Ini lebih dari arsitektur Transformer yang dijelaskan dalam makalah asli ( 6 lapisan encoder ).
* Arsitektur BERT (BASE dan LARGE) juga memiliki jaringan feedforward yang lebih besar (masing-masing 768 dan 1024 unit tersembunyi), dan lebih banyak perhatian (masing-masing 12 dan 16) daripada arsitektur Transformer yang disarankan dalam makalah asli. Ini berisi 512 unit tersembunyi dan 8 kepala perhatian.
* BERT BASE berisi 110 juta parameter sedangkan BERT LARGE memiliki 340 juta parameter.

A close-up of a white rectangular object

Description automatically generated

**Gambar 3. 16** Arsitektur Bert Base & Bert Large

Model ini memproses masukan dimulai dengan token CLS, yang merupakan token klasifikasi, diikuti oleh serangkaian kata sebagai masukan. Token CLS ini berfungsi sebagai penanda awal untuk pemrosesan. Masukan kemudian diteruskan ke lapisan-lapisan di atasnya. Setiap lapisan menggunakan mekanisme perhatian mandiri dan hasilnya kemudian diteruskan ke jaringan feedforward sebelum akhirnya diserahkan ke pembangun enkode berikutnya. Model ini menghasilkan vektor dengan ukuran tersembunyi (768 untuk BERT BASE). Jika kita ingin menggunakan model ini untuk klasifikasi, kita dapat menggunakan keluaran yang terkait dengan token CLS.

A white rectangular object with a black background

Description automatically generated

**Gambar 3. 17** Bert Embeddings

## Perancangan Sistem

### Use Case Diagram

Berikut ini adalah use case diagram pada sistem analisis sentimen terhadap data ulasan Hanafashion\_shop di Tokopedia :



**Gambar 3. 18** Use Case Diagram

### Activity Diagram

*Activity diagram* merupakan alur aktivitas pengguna terhadap system. Dengan adanya *activity diagram* dapat mengetahui alur interaksi yang terjadi pada *use case diagram* :

1. *Activity Diagram Running System*

Diagram aktifitas ini menggambarkan aktifitas *system interface* menggunakan streamlit dijalankan. Adapun alur aktivitas proses running system dapat dilihat pada gambar di bawah.

A black background with white text

Description automatically generated

**Gambar 3. 19** Activity Diagram Running System

1. *Activity Diagram Preprocessing*

Diagram aktivitas ini menggambarkan *preprocessing* data dari ulasan Hanafashion\_shop yang ada di Tokopedia sebelum dilakukan proses training data menggunakan algoritma *Support Vector Machine* dan *indoBert*. Adapun alur aktivitas *preprocessing* dapat dilihat pada gambar di bawah.

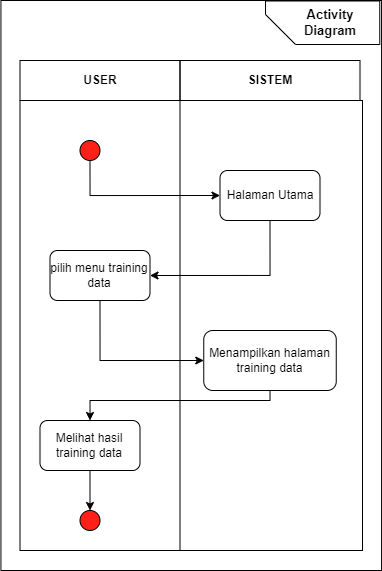
A screenshot of a cell phone

Description automatically generated

**Gambar 3. 20** Activity Diagram Preprocessing

1. *Activity Diagram Training Data*

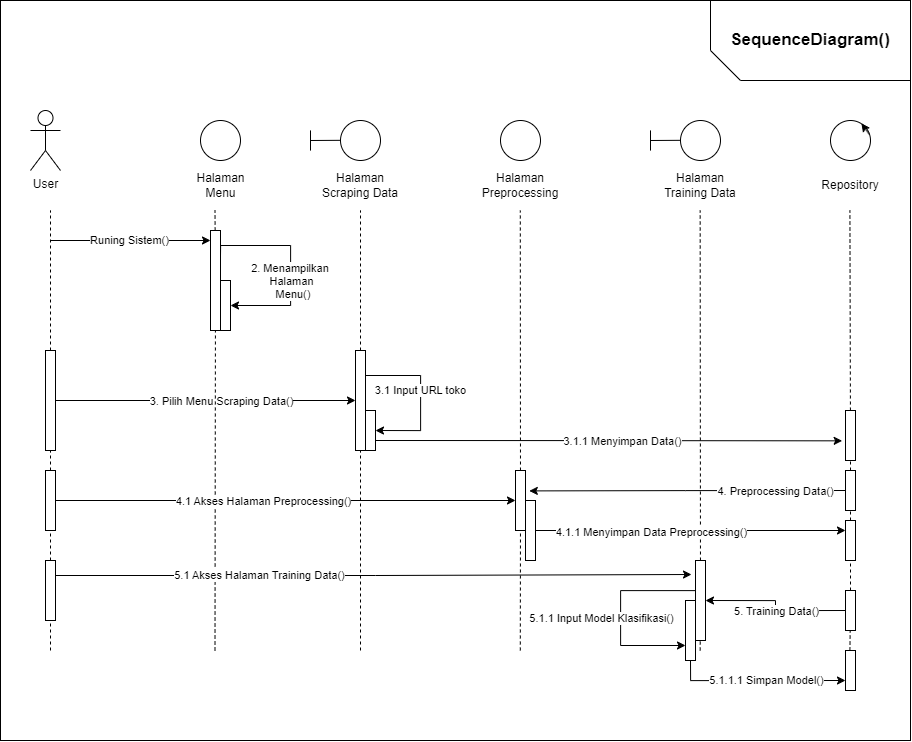
Diagram aktivitas ini menggambarkan tahapan training data menggunakan algoritma *Support Vector Machine* dan *indoBert*, user bisa memasukkan input parameter sebelum melakukan training data. Adapun alur aktivitas *training data* dapat dilihat pada gambar di bawah.



**Gambar 3. 21** Activity Diagram Training Data

### Sequence Diagram

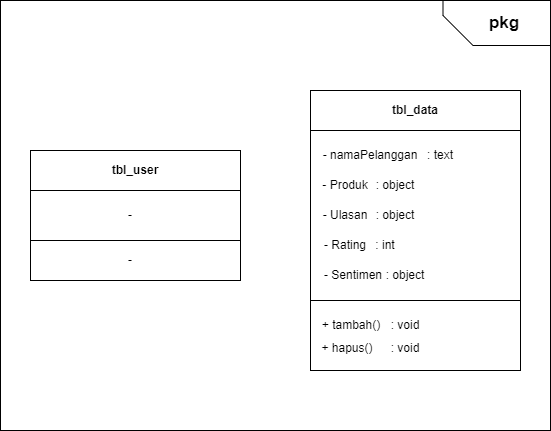
Diagram sequence menggambarkan detail alur proses berdasarkan urutan waktu. Dalam proses analisis sentimen, pengguna memulai dengan mengakses menu aplikasi yang tersedia. Sistem kemudian melakukan *scraping* data dari sumbernya, seperti media sosial, lalu melakukan *preprocessing* untuk membersihkan dan mempersiapkan data teks. Setelah itu, sistem melatih model analisis sentimen menggunakan data yang telah diproses dan menyimpannya ke dalam repository untuk penggunaan dan referensi di masa depan. Adapun tampilan proses *sequence diagram* dapat dilihat pada gambar di bawah.



**Gambar 3. 22** Sequence Diagram

### Class Diagram

Berikut ini adalah *class diagram* dari analisis sentiment pada toko Hanafashion\_shop dari platform Tokopedia menggunakan algoritma Support Vector Machine dan IndoBert.



**Gambar 3. 23** Class Diagram

# HASIL DAN PEMBAHASAN

## Environment Testing

Hasil dari langkah ini adalah persiapan perangkat keras dan perangkat lunak yang akan digunakan untuk mendesain sistem, mengembangkan sistem, dan mengujinya. Perangkat pengembangan yang dipakai termasuk:

1. Kebutuhan minimum perangkat keras

Perangkat minimum yang dibutuhkan untuk dapat menjalankan system perangkat computer atau laptop dengan detail spesifikasi sebagai berikut :

1. Intel Core i5-10351G
2. Memory RAM 8 GB
3. Kebutuhan perangkat lunak
4. Sistem operasi *Microsoft Windows* 11
5. *Google Chrome*
6. *Visual Studio Code*
7. *Python & Library*

## Hasil

Untuk mengevaluasi kinerja dalam studi ini, kami menggunakan metode matriks kebingungan untuk menghitung recall, presisi, dan akurasi dari setiap kategori. Dari total 1995 ulasan yang diambil, kami membaginya menjadi dua bagian: 80% untuk data pelatihan dan 20% untuk data pengujian. Berikut adalah hasil evaluasi kinerja dari data pengujian yang kami peroleh menggunakan *confusion matrix*.

### Hasil Support Vector Machine

Dalam eksperimen menggunakan algoritma *Support Vector Machine* dengan splitting data 80/20, confusion matrix yang dihasilkan menunjukkan bahwa dari total 190 sampel, terdapat 5 prediksi positif yang benar (True Positives), 2 prediksi positif yang salah (False Positives), 165 prediksi negatif yang benar (True Negatives), dan 18 prediksi negatif yang salah (False Negatives). Evaluasi kinerja model dilakukan dengan menghitung metrik seperti akurasi, presisi, recall, dan F1-score, yang menyediakan pemahaman yang lebih holistik tentang kecocokan model terhadap data uji. Adapun tabel dan perhitungan evaluasi dapat dilihat sebagai berikut :

**Tabel 4. 1** Test Result Support Vector Machine

|  |  |  |
| --- | --- | --- |
| Actual Class | Positif | Negatif |
| Positif | 5 | 2 |
| Negatif | 18 | 165 |

**Positif :**

**Negatif :**

***Accuracy :***

### Hasil IndoBert

Dalam eksperimen menggunakan model *IndoBERT* dengan splitting data 80/20, confusion matrix yang dihasilkan menunjukkan bahwa dari total 190 sampel, terdapat 154 prediksi positif yang benar (True Positives), 13 prediksi positif yang salah (False Positives), 11 prediksi negatif yang benar (True Negatives), dan 12 prediksi negatif yang salah (False Negatives). Evaluasi kinerja model dilakukan dengan menghitung metrik seperti akurasi, presisi, recall, dan F1-score, yang memberikan gambaran menyeluruh tentang kemampuan model IndoBERT dalam mengklasifikasikan data uji. Adapun tabel dan perhitungan evaluasi dapat dilihat sebagai berikut :

**Tabel 4. 2** Test Result IndoBert

|  |  |  |
| --- | --- | --- |
| Actual Class | Positif | Negatif |
| Positif | 154 | 13 |
| Negatif | 12 | 11 |

**Positif :**

**Negatif :**

***Accuracy :***

**Tabel 4. 3** Perbandingan Performa Model

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Performa Model** | ***Accuracy*** | ***Precision*** | ***Recall*** | ***F-Measure*** |
| *Support Vector Machine* | **89%** | **98%** | **90%** | **93%** |
| *IndoBert* | 87% | 93% | 92% | 92% |

## Deskripsi Data

### Data Hasil Scraping

### Data Training Model

## Hasil Perbandingan Model

### Hasil Model *Support Vector Machine*

### Hasil Model *IndoBert*

### Hasil Model Terbaik

## Analisa Model Terbaik

## Prosedur Penggunaan Aplikasi

### Halaman Home

### Halaman Scraping

### Halaman Merge Data

### Halaman Dataset

### Halaman Preprocessing

### Halaman Visualization

### Halaman Support Vector Machine

### Halaman IndoBert

### Halaman Testing