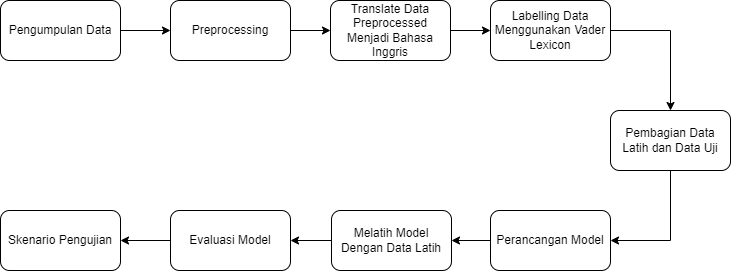
**BAB III**

**METODE PENELITIAN**

* 1. **Tahapan Penelitian**

Pada BAB ini, akan dijelaskan tahapan metode penelitian yang terdiri dari pengumpulan data, pengolahan data, pembuatan model deep learning, evaluasi model hingga skenario pengujian. Skema penelitian bisa ditinjau dalam gambar 3.1.



Gambar 3.1 Tahapan Penelitian

* 1. **Pengumpulan Data**

Data yang digunakan dalam penelitian ini adalah hasil dari *web scrapping* di situs Twitter dengan mencari 20.000 total tweet yang mengandung kata “bjorka”. Peneliti melakukan *web scrapping* untuk mengambil tweet yang tidak mengandung link serta mention/reply dari pengguna lain sehingga tweet tersebut murni sebuah tweet yang berisi ide/gagasan/pendapat yang ingin dituliskan oleh pengguna. Setelah itu, data disatukan menjadi sebuah dataframe/table yang berisi username pengunggah tweet, waktu dibuatnya tweet tersebut serta isi tweet/teks tweet tersebut. Setelah itu, dataframe disimpan dalam format excel (.xlsx) agar bisa dilihat dengan mudah dan dapat digunakan kembali ketika akan digunakan dalam penelitian. Berikut beberapa sampel dataset hasil web scrapping dapat dilihat pada table 3.1.

|  |  |  |
| --- | --- | --- |
| created at | username | text |
| 2022-11-02 | rudiartraa | yaah , udah selesai nih Bjorka Bjorkaan nya ? |
| 2022-11-02 | oi\_filix315 | Kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :((( |
| 2022-11-01 | falahsanjaya\_ | Knp ga ada yg jadi Bjorka Halloween party kemaren? :(( |
| 2022-11-01 | namanyaajess | Beritanya taik, mau ferdi sambo, mau bjorka, mau apalah yang kelaperan ga mungkin jadi tranding topik, yang tertindas sulit untuk berdiri, yang melawan akan tetap dibungkam. |
| 2022-11-01 | namakubaim | Apa kabar Bjorka? |

Tabel 3.1 Sample dataset

Selain itu, peneliti mengumpulkan dataset yang berisi kata-kata alay yang nantinya digunakan untuk melakukan transformasi kata-kata alay yang mungkin saja di dapatkan di dalam tweet menjadi bentuk normalnya. Dataset alay ini didapatkan dari Kaggle.com dalam format csv (comma separated value). Berikut beberapa sample dari kata-kata alay yang dapat dilihat pada table 3.2.

|  |  |
| --- | --- |
| Kata Alay | Pembenaran |
| aje | aja |
| ajg | anjing |
| diem | diam |
| ente | kamu |
| gr | gede rasa |
| ijo | hijau |
| knp | kenapa |

Tabel 3.2 Sample kata-kata alay dan pembenarannya

Kata-kata alay ini nantinya akan berguna Ketika memasuki tahap preprocessing data untuk mentransformasi kata alay menjadi bentuk dasar/bentuk benar.

Untuk memudahkan dalam memahami data, maka dapat melihat pada tabel 3.3 yang berisi tentang metadata dataset dan juga kamus alay.

|  |  |
| --- | --- |
| **Variable** | **Keterangan** |
| created at | Tanggal tweet diunggah |
| username | Nama pengguna akun Twitter |
| text | Tweet/teks yang diunggah oleh pengguna Twitter |

Tabel 3.3 Metadata dataset

|  |  |
| --- | --- |
| **Variable** | **Keterangan** |
| Kamus Alay | Kata dalam bentuk alay |
| Pembenaran | Kata setelah diperbaiki/dikembalikan ke bentuk semulal |

Tabel 3.4 Metadata Kamus Alay

* 1. **Preprocessing**

Tahapan ini bertujuan untuk membuat kualitas data yang akan digunakan dalam training/analisis menjadi baik. Dalam kasus sentiment analysis ujaran kebencian ini, untuk membuat kualitas dataset teks menjadi baik, maka perlu dilakukan beberapa tahapan preprocessing diantaranya adalah lowercasing, pembersihan karakter selain alphanumeric, pembersihan karakter khusus/tidak biasa, normalisasi kata-kata alay, stemming, membersihkan stopword, tokenizing serta tranformasi data.

1. Lowercasing

Lowercasing merupakan tahapan transformasi huruf menjadi huruf kecil. Input dari tahapan ini adalah sebuah kalimat yang terdiri dari huruf besar dan kecil yang akan diproses menjadi sebuah output kalimat yang terdiri dari huruf kecil saja. Berikut input dan output setelah dilakukan lowercasing dapat dilihat pada table 3.5.

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
| yaah , udah selesai nih Bjorka Bjorkaan nya ? | yaah , udah selesai nih bjorka bjorkaan nya ? |
| Apa kabar Bjorka? | apa kabar bjorka? |
| Kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :((( | kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :((( |

Tabel 3.7 Lowercasing

1. Pembersihan Non-Alphanumeric

Alphanumeric merupakan karakter yang teridiri dari huruf dan angka. Pembersihan Non-Alphanumeric bertujuan untuk membersihkan karakter selain huruf dan angka, seperti tanda baca dan karakter khusus lainnya. Tahapan ini dilakukan agar data yang akan digunakan dalam penelitian dapat bekerja secara maksimal karena dalam membuat model deep learning, nantinya setiap kata akan dijadikan sebuah token yang di transformasi menjadi sebuah angka. Berikut input dan output untuk proses pembersihan Non-Alphanumeric karakter yang dapat dilihat pada table 3.8.

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
| yaah , udah selesai nih bjorka bjorkaan nya ? | yaah udah selesai nih bjorka bjorkaan nya |
| apa kabar bjorka? | apa kabar bjorka |
| kadang gua mikir sekilas pp gua mirip bjorka, tapi kan gua udh pake pp ini lama :((( | kadang gua mikir sekilas pp gua mirip bjorka tapi kan gua udh pake pp ini lama |

Tabel 3.8 Pembersihan Non-Alphanumeric

1. Pembersihan karakter khusus

Pembersihan karakter khusus ini bertujuan untuk menghapus beberapa karakter khusus yang ada di dalam kalimat/teks, seperti kalimat yang mengandung kata “\n”, url, serta emoji. Hal ini akan membuat dataset yang nantinya akan digunakan kualitasnya menjadi baik karena kata-kata khusus tersebut yang berasal dari media sosial twitter cenderung tercantum di dalam kalimat/dataset tersebut sehingga akan membuat model machine learning kurang baik dalam melakukan klasifikasi. Berikut contoh input dan output untuk pembersihan karakter khusus yang dapat dilihat pada table 3.9.

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
| rame rame pada hapus twit yg menuduh msf adalah sosok bjorka 🤣🤣 | rame rame pada hapus twit yg menuduh msf adalah sosok bjorka |
| bjorka kembali tebar ancaman, kecam semua akun palsuhttp://inet.detik.com/security/d-6303238/bjorka-kembali-tebar-ancaman-kecam-semua-akun-palsu?bytedance=true | bjorka kembali tebar ancaman, kecam semua akun palsu |
| bjorka, aku padamu🫶🏻🖤 | bjorka, aku padamu |

Tabel 3.9 Pembersihan karakter khusus

1. Normalisasi Kata-Kata Alay

Normalisasi kata-kata alay bertujuan untuk mengubah kata-kata yang dikategorikan menjadi kata-kata alay menjadi bentuk dasarnya. Hal ini bertujuan untuk mengurangi variansi kata yang ada di dalam dataset yang akan membuat kualitas dataset menjadi lebih baik dan optimal dalam klasifikasi nantinya. Tahapan ini akan menghilangkan kata alay yang sudah terdaftar pada dataset kata-kata alay yang sudah didapatkan melalui Kaggle. Berikut beberapa contoh normalisasi kata-kata alay menjadi bentuk dasarnya yang dapat dilihat pada table 3.8.

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
| anjirrr kok bisa adek gw percaya kalo bjorka itu gus syamsudin | anjirrr kok bisa adik gue percaya kalau bjorka itu gus syamsudin |
| knp ga ada yg jadi bjorka halloween party kemaren | kenapa tidak ada yang jadi bjorka halloween party kemarin |
| pengen gitu bikin sih bjorka biar gak bikin orang gak gagal fokus rakyat karena saat banyak kasus yg gagal fokus karena hacker bjorka bayaran pemerintah | ingin begitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayaran pemerintah |

Tabel 3.10 Normalisasi kata alay

1. Stemming word

Stemming merupakan Teknik menguraikan bentuk suatu kata menjadi bentuk asalnya/dasarnya. Hal ini dilakukan untuk mengubah kata berimbuhan menjadi bentuk dasarnya agar mempermudah model dalam melakukan klasifikasi, mengurangi variansi serta meningkatkan kualitas dataset menjadi lebih optimal. Sebagai contoh, untuk kata membenci akan ditransformasi menjadi benci, menyukai menjadi suka dan lebih banyak lagi. Berikut beberapa contoh stemming yang dapat dilihat pada table 3.11

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
| anjirrr kok bisa adek gw percaya kalo bjorka itu gus syamsudin | anjirrr kok bisa adik gue percaya kalo bjorka itu gus syamsudin |
| knp ga ada yg jadi bjorka halloween party kemaren | kenapa tidak ada yang jadi bjorka halloween party kemarin |
| pengen gitu bikin sih bjorka biar gak bikin orang gak gagal fokus rakyat karena saat banyak kasus yg gagal fokus karena hacker bjorka bayaran pemerintah | ingin gitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayaran pemerintah |

Tabel 3.11 Stemming word

1. Pembersihan stopword

Tahapan pembersihan stopword berfungsi untuk menghilangkan kata umum (common words) yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna. Tujuan dari menghilangkan stop words adalah mengurangi jumlah kata dalam sebuah dokumen yang akan berpengaruh terhadap kecepatan dan performa model. Stop word yang akan dijadikan sebagai acuan didapatkan dari libraries python Sastrawi (Sastrawi, 2017). Berikut beberapa contoh data sebelum dan sesudah dilakukannya tahapan pembersihan stopword yang dapat dilihat pada tabel 3.12

|  |  |
| --- | --- |
| **Sebelum** | **Sesudah** |
| anjirrr kok bisa adik gue percaya kalo bjorka itu gus syamsudin | anjirrr kok bisa adik gue percaya bjorka gus syamsudin |
| kenapa tidak ada yang jadi bjorka halloween party kemarin | bjorka halloween party kemarin |
| ingin gitu bikin sih bjorka biar tidak bikin orang tidak gagal fokus rakyat karena saat banyak kasus yang gagal fokus karena hacker bjorka bayaran pemerintah | gitu bikin sih bjorka biar bikin gagal fokus rakyat gagal fokus hacker bjorka bayaran pemerintah |

Tabel 3.12 Stopword Removal

* 1. **Penerjemahan Data**

Tahapan ini bertujuan untuk menerjemahkan data tweet dari Bahasa Indonesia menjadi Bahasa Inggris. Hal ini perlu dilakukan karena dalam pelabelan menggunakan Vader, dibutuhkan dataset dengan Bahasa inggris sehingga dapat diketahui kutub/kecenderungan sebuah kalimat, apakah positif, negative, atau bahkan netral. Dataset diterjemahkan melalui program python menggunakan libraries translate. Penggunaan libraries ini memungkinkan untuk melakukan translate dengan memanggil API libraries tersebut dan melakukan translate di dalam program python, kemudian menambahkan hasil translate menjadi kolom baru. Berikut beberapa sample hasil penerjemahan dataset Bahasa Indonesia menjadi Bahasa Inggris yang dapat dilihat pada table 3.5.

|  |  |
| --- | --- |
| Indonesia | Inggris |
| yaah , udah selesai nih Bjorka Bjorkaan nya ? | yeah, have you finished Bjorka Bjorkaan? |
| Apa kabar Bjorka? | How are Bjorka? |
| Bjorka kemana ya | Where is Bjorka |
| Andai bjorka bisa tolong,,, 😞 | If bjorka can help ,,, 😞 |
| Pagi semua, kecuali bjorka | Morning all, except Bjorka |

Tabel 3.5 Sample dataset setelah diterjemahkan menjadi Bahasa Inggris menggunakan libraries translator dari python.

Setelah diterjemahkan, maka akan melakukan pencarian kutub kecenderungan dataset, baik bermuatan positif, negative ataupun netral.

* 1. **Pelabelan Data**

Pelabelan data berfungsi untuk memberikan label yang terdiri dari positif, negative maupun netral terhadap dataset. Hal ini perlu dilakukan karena model deep learning yang akan dibangun membutuhkan sebuah label untuk dapat berlatih dan mengklasifikasikan ujaran kebencian twitter. Untuk melakukan pelabelan terhadap dataset, menggunakan Vader Lexicon. Vader (Valence Aware Dictionary and Sentiment Reasoner) Lexicon merupakan sebuah tools/libraries yang berfungsi untuk melakukan sentiment analysis berbasis aturan yang secara khusus disesuaikan dengan sentiment yang diungkapkan di media sosial (MIT, 2014). Tools ini bersifat open-source dan dibawah lisensi MIT. Vader dapat menangani beberapa kasus khusus dalam sentiment analysis, seperti memahami negasi, memahami arti emoticon/emoji, memahami arti sebuah akronim, penggunaan tanda baca konvensional menandakan peningkatan intensitas sentiment, dan masih banyak lagi.

Dengan menggunakan Vader, data teks akan diproses dan dicari nilai sentimennya. Output yang dihasilkan oleh Vader merupakan sebuah probabilitas dari 3 kemungkinan yaitu positif, negative dan netral. Dari probabilitas tersebut, untuk melakukan labelling terhadap suatu teks di dalam dataset, maka diambil nilai compound dari ketiga probabilitas tersebut. Nilai compound adalah metric yang menghitung semua peringkat leksikon yang telah dinormalisasi antara -1 (paling negative) sampai dengan 1(paling positif). Untuk penelitian ini sentiment positif nilai compound nya dalam rentang lebih dari sama dengan 0,05, sentiment netral diantara -0,05 sampai dengan 0,05 serta untuk sentiment negative kurang dari sama dengan -0,05. Setelah menemukan nilai compound dan melakukan pelabelan data berdasarkan sentiment hasil nilai compound, maka dataset hasil pelabelan siap digunakan untuk diproses ditahap selanjutnya yaitu tahap preprocessing. Berikut contoh output dari Vader sentiment analysis.

|  |  |  |
| --- | --- | --- |
| **Teks** | **Vader Output** | **Label** |
| Bjorka is fun | {'neg': 0.0, 'neu': 0.694, 'pos': 0.306, 'compound': 0.296} | Positive |
| Bjorka please leak the secret document of suspicious plot funds | {'neg': 0.189, 'neu': 0.775, 'pos': 0.036, 'compound': -0.6728} | Negative |
| Talk to Bjorka Drinking Coffee Together | {'neg': 0.0, 'neu': 0.954, 'pos': 0.046, 'compound': 0.0387} | Neutral |

Tabel 3.6 Labeling Data menggunakan Vader Sentiment Analysis

Setelah mendapatkan label, maka label yang masih berupa text perlu dilakukan pengubahan menjadi kategori numeris. Karena terdapat tiga label, maka akan dikonversi menjadi 0, 1 dan 2. Hal ini bertujuan untuk dapat mengimplementasikan data kedalam model yang membutuhkan label numeris.

* 1. **Pemisahan Data**

Pemisahan data bertujuan untuk membagi dataset menjadi dua bagian, yaitu bagian data latih dan data uji. Hal ini dilakukan untuk memisahkan data yang akan digunakan untuk melatih model dan data yang digunakan untuk uji data tunggal/testing model. Dataset akan dibagi menjadi data latih dan data uji dengan rasio 0,8:0,2 dimana data latih sebesar 16.000 dan data uji sebesar 4.000. Setelah dipisahkan, label pada data latih dan data uji akan dipisahkan untuk siap digunakan pada tahap melatih model.

Setelah data dipisahkan, maka label harus dilakukan transformasi menjadi bentuk one hot encoder. Label akan ditransformasi dari data tunggal menjadi sebuah array yang berisi 3 komponen angka. Hal ini perlu dilakukan untuk menyesuaikan penerapan deep learning menggunakan keras libraries. Berikut contoh penerapan one hot encoder yang dapat dilihat pada tabel 3.14.

|  |  |
| --- | --- |
| Label | Label One Hot Encoder |
| 0 | [1,0,0] |
| 1 | [0,1,0] |
| 2 | [0,0,1] |

Tabel 3.13. Label one hot encoder.

1. Tokenizing

Tahapan tokenizing bertujuan untuk mengubah kata menjadi sebuah token, dan dalam penelitian ini kata akan diubah menjadi sebuah token angka. Tokenizing dapat dilakukan dengan cara berikut:

1. Mengimport library keras Tokenizer
2. Menginisialisasi tokenizer
3. Melakukan fit tokenizer dengan dataset. Hal ini bertujuan untuk mendeteksi semua kata yang ada di dalam dataset dan mengubah kata tersebut menjadi sebuah token angka.
4. Mengaplikasikan tokenizer ke setiap data yang ada di dalam data latih dan data uji dan menyimpannya sebagai sebuah sequences.
5. Memberikan padding ke setiap sequences sesuai dengan sequences terpanjang dengan menambahkan token “0” sampai ke element terakhir padding.

Untuk mempermudah melihat hasilnya, dapat dilihat pada tabel 3.13

|  |  |  |
| --- | --- | --- |
| Kalimat | Tokenizer | Padded |
| selesai nih bjorka bjorkaan | [131, 17, 2, 1] | [131, 17, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] |
| bjorka mending data kominfo data wifiku | [2, 106, 5, 60, 5, 1] | [ 2, 106, 5, 60, 5, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] |
| sejarah bjorka spill data bunuh munir tau | [947, 2, 86, 5, 114, 83, 14] | [947, 2, 86, 5, 114, 83, 14, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] |
| kabar bjorka | [89, 2] | [89, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] |

Tabel 3.14 Tokenizing kalimat

* 1. **Perancangan Model**

Model yang akan dirancang dalam penelitian ini akan terdiri dari beberapa hidden layer yang akan disusun menjadi sebuah sequences yang terdiri dari beberapa layers. Model yang akan dirancang akan mengimplementasikan model Sequentials dari libraries keras yang berarti mengelompokkan tumpukan lapisan linier. Dalam penelitian ini, akan lebih fokus pada susunan layer model deep learning dibandingkan parameter lain karena

1. Embedding Layer

Embedding layer merupakan input layer atau bisa dikatakan dengan layer pertama dalam model deep learning yang akan dirancang. Layer ini berfungsi untuk mengelompokan kata menjadi dua kutub yang berbeda. Parameter yang digunakan dalam layer ini adalah sebagai berikut:

Input\_length: Panjang input sequence

Output\_dim: dimensi output

Input\_dim: ukuran vocabulary

Hasil dari embbeding layer dapat dilihat pada tabel 3.10

|  |  |
| --- | --- |
| Sebelum | Sesudah |
| [131, 17, 2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] | [[[-0.03162958 0.04269071 0.01509007 0.00124536]  [-0.04351017 0.000172 -0.03358928 0.0472861 ]  [-0.01396496 0.02855982 0.03887559 -0.01323872]  [-0.00201822 0.03641895 0.0171499 -0.02244518]  [-0.02064002 0.02616386 -0.03546425 -0.03617959]  …………………………….  [-0.02064002 0.02616386 -0.03546425 -0.03617959]]] |

Gambar 3.11 Hasil embedding layer

1. Dense Layer

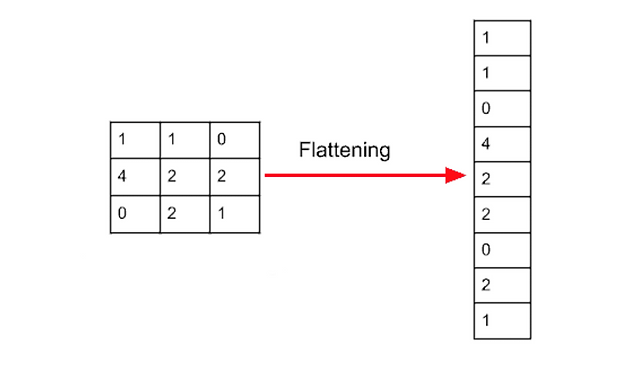
Dense layer akan memetakan input dari hidden layer sebelumnya menjadi sebuah output yang jumlahnya sesuai dengan jumlah unit di layer tersebut. Berikut contoh dari dense layer dari input hidden layer sebelumnya yang kemudian dipetakan sesuai jumlah output unit di layer tersebut.

|  |  |
| --- | --- |
| [[[-0.03162958 0.04269071 0.01509007 0.00124536]  [-0.04351017 0.000172 -0.03358928 0.0472861 ]  [-0.01396496 0.02855982 0.03887559 -0.01323872]  [-0.00201822 0.03641895 0.0171499 -0.02244518]  [-0.02064002 0.02616386 -0.03546425 -0.03617959]  …………………………….  [-0.02064002 0.02616386 -0.03546425 -0.03617959]]] | [[[-0.03966843]  [ 0.04057634]  [-0.01823187]  [ 0.05386413]  [-0.05570652]  [-0.05570652]  [-0.05570652]  ………………..  [-0.05570652]]] |

Tabel 3.12 Hasil dense layer dengan output unit=1

1. Flatten Layer

Flatten layer berfungsi untuk melakukan transformasi neurons dari multi dimensi menjadi dimensi tunggal. Dalam penelitian ini, flatten layer digunakan untuk melakukan transformasi output dari layer Bidirectional/Embedding layer yang tergolong multi dimensi (3 dimensi) menjadi dimensi tunggal agar bisa diterima oleh hidden layer dense.



Gambar 3.2. Flatten layer

1. LSTM Layer
   1. **Evaluasi Model**

Evaluasi model dilaukan dengan melakukan prediksi terhadap data uji, dan kemudian menyocokan kesamaan antara label asli dengan label hasil prediksi. Evaluasi akan ditinjau dari segi akurasi serta metric f1-score. Berikut rumus untuk f1-score yang dapat dilihat pada persamaan dibawah ini:

Keterangan :

TP = True Positive (data prediksi benar dan data actual benar)

TN = True Negative (data yang diprediksi benar dan data actual salah)

FP = False Positive (data prediksi salah dan data actual benar)

FN = False Negative (data prediksi salah dan data actual salah)

* 1. **Skenario Pengujian**

Tahapan ini bertujuan untuk menjawab rumusan masalah yang ada pada Bab I yaitu sentiment analysis terhadap ujaran kebencian twitter. pada tahapan ini, data akan dilakukan preprocessing terlebih dahulu. Setelah itu, data akan dipisahkan antara data latih dan data uji. Setelah dipisahkan dengan rasio 0.8 untuk data latih dan 0.2 untuk data uji. Setelah itu, masing-masing label dari data uji dan data latih akan ditransformasi menggunakan one hot encoder agar menjadi sebuah array yang berisi 3 angka. Setelah melalui tahap pemisahan data, lalu akan mendefinisikan model deep learning. Model akan disusun dengan ketentuan berikut:

1. Layer model

* Embedding layer
* Bidirectional LSTM layer
* Flatten layer
* Dense layer

Model akan disusun terdiri dari layer di atas.

1. Activation Function

* Tanh (LSTM layer)
* Re-Lu (Dense layer)
* Softmax (Dense layer terakhir)

1. Loss function

* Categorical\_crossentropy (klasifikasi kategorikal)

1. Optimizer

* RMSProp (Root Meann Squared Error Propagation)
* Adam

Untuk optimizer, akan diset learning rate sebesar 0,0003.

Input model berupa array yang berisi nomor index kata yang telah diset pada tokenizer yang menampung 17586 indeks kata, dengan Panjang maksimum tiap token adalah 100 kata. Proses latih model akan berlangsung selama 20 epoch/iterasi dan dilakukan validasi dengan besaran ukuran data validasi adalah 0.2 dari data latih. Untuk pengukuran performa, akan melihat pada akurasi, validasi akurasi, loss, validasi loss, f1\_m dan validasi f1\_m.

Setelah itu, akan ada beberapa skenario pengujian yang akan diuraikan sebagai dengan fokus utama layer, optimizer dan loss function. Berikut rincian pembagian skenario pengujian:

1. Skenario 1
2. Model 1.a

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Unit | Total Param | Activation Layer |
| Embedding | 100 | 1758600 | - |
| Flatten | 0 | 0 | - |
| Dense | 512 | 5120512 | Relu |
| Dense | 3 | 1539 | Softmax |

1. Model 1.b

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Unit | Total Param | Activation Layer |
| Embedding | 100 | 1758600 | - |
| Bidirectional LSTM | 64 | 84480 | tanh |
| Bidirectional LSTM | 64 | 98816 | tanh |
| Flatten | 0 | 0 | - |
| Dense | 512 | 66048 | Relu |
| Dense | 3 | 1539 | Softmax |

1. Optimizer : Adam (learning rate=0.001)
2. Loss Functtion : categorical crossentropy
3. Epochs : 5
4. Skenario 2
5. Model 2.a

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Unit | Total Param | Activation Layer |
| Embedding | 100 | 1758600 | - |
| Flatten | 0 | 0 | - |
| Dense | 512 | 5120512 | Relu |
| Dense | 3 | 1539 | Softmax |

1. Model 2.b

|  |  |  |  |
| --- | --- | --- | --- |
| Layer | Unit | Total Param | Activation Layer |
| Embedding | 100 | 1758600 | - |
| Bidirectional LSTM | 64 | 84480 | tanh |
| Bidirectional LSTM | 64 | 98816 | tanh |
| Flatten | 0 | 0 | - |
| Dense | 512 | 66048 | Relu |
| Dense | 3 | 1539 | Softmax |

1. Optimizer : RMSProp (learning rate=0.001)
2. Loss Functtion : mean squared error
3. Epochs : 5

Setelah membuat beberapa skenario pengujian model deep learning, maka akan melakukan training untuk setiap skenario pengujian untuk mencari model yang paling optimal dengan menggunakan metrics evaluasi yaitu akurasi dan f1 score. Setelah melakukan training, maka akan menampilkan grafik akurasi, f1 score serta loss dari proses training terhadap tiap epochs yang berjalan. Setelah dilakukan training, akan beralih ke tahap evaluasi model dengan menggunakan data uji yang telah dipisahkan.