SKRIPSI

KLASIFIKASI *HATE SPEECH* MENGGUNAKAN *LONG SHORT-TERM MEMORY* (LSTM)

OLEH
Hanna Safitri
07351911083

PROGRAM STUDI INFORMATIKA
FAKULTAS TEKNIK
UNIVERSITAS KHAIRUN
TERNATE
2024

LEMBAR PENGESAHAN

KLASIFIKASI *HATE SPEECH* MENGGUNAKAN *LONG SHORT-TERM MEMORY* (LSTM)

Oleh
Hanna Safitri
07351911083

Skripsi ini telah disahkan
Tanggal 10 Juli 2024

Menyetujui
Tim Penguji

Ketua Penguji

YASIR MUIN, S.T., M.Kom.
NIDN. 9990582796

Pembimbing I

MUHAMMAD FHADLI, S.Kom., M.Sc.
NIP. 199611232023211012

Anggota Penguji

Dr. MUHAMMAD RIDHA ALBAAR, S.Kom., M.Kom.
NIP. 198504232008031001

Pembimbing II

Ir. ABDUL MUBARAK, S.Kom., M.T., IPM.
NIP. 198212062014041002

Anggota Penguji

HAIRIL KURNIADI SIRAJUDDIN, S.Kom., M.Kom.
NIP. 198204272023211009

Mengetahui/Menyetujui

Koordinator Program Studi
Informatika

ROSIHAN, S.T., M.Cs.
NIP. 197607192010121001

Dekan Fakultas Teknik
Universitas Khairun

Ir. ENDAH HARISUN, S.T., M.T., CRP.
NIP. 197511302005011013

LEMBAR PERNYATAAN KEASLIAN

Yang bertanda tangan dibawah ini:

Nama	: Hanna Safitri
Npm	: 07351911083
Fakultas	: Teknik
Jurusan/Program Studi	: Informatika
Judul	: Klasifikasi *Hate Speech* Menggunakan *Long Short-Term Memory* (LSTM)

Dengan ini menyatakan bahwa penulisan Skripsi yang telah saya buat ini merupakan hasil karya sendiri dan benar keasliannya. Apabila ternyata dikemudian hari penulisan Skripsi ini merupakan hasil plagiat atau penjiplakan terhadap karya orang lain, maka saya bersedia mempertanggung jawabkan sekaligus bersedia menerima sanksi berdasarkan aturan tata tertib di Universitas Khairun.

Demikian pernyataan ini saya buat dalam keadaan sadar dan tidak dipaksakan.

Penulis

Hanna Safitri

LEMBAR PERSEMBAHAN

Puji syukur penulis panjatkan kepada Allah SWT sehingga penulis dapat menyelesaikan skripsi ini dengan baik. Oleh karena itu, skripsi ini penulis persembahkan kepada:

1. Bapak tercinta Alm. Muhammad Hamim, beliau memang belum sempat menemani penulis dalam perjalanan selama menempuh pendidikan hingga selesai. Namun, beliau akan tetap ada dalam setiap perjalanan saya.
2. Ibunda tercinta Nahra Bodja. Terima kasih sebesar-besarnya penulis berikan kepada beliau yang sangat berjasa dalam hidup penulis. Terima kasih atas do'a, cinta, kepercayaan, serta tanpa lelah mendengar keluh kesah penulis dalam menyelesaikan skripsi ini.
3. Adik Dian Okviani, Kakak Yayan Hardian, dan juga Suci Adliah Sudirman, S.H, serta seluruh keluarga yang selalu memberikan dukungan dan perhatian penuh sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
4. Ketua Prodi Informatika, Dosen Pembimbing, Dosen Penguji, dan seluruh Dosen Prodi Informatika yang telah memberi ilmu, didikan dan pengalaman yang sangat berarti hingga penulis bisa sampai di tahap ini. Saya mengucapkan banyak terima kasih. Semoga kebaikan juga selalu menyertai bapak/ibu dosen sekalian.
5. Dan skripsi ini dipersembahkan kepada orang yang selalu bertanya "Kapan Wisuda?". Lulus tidak tepat waktu bukanlah sebuah kejahatan dan bukan pula sebuah aib. Jika mengukur kecerdasan seseorang hanya dari siapa yang paling cepat lulus. Bukankah sebaik-baiknya skripsi adalah skripsi yang selesai?

MOTTO

"Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya."

(Q.S Al-Baqarah: 286)

"Jangan terlalu dikejar, jika memang jalannya Allah pasti memperlancar"

(H.R Muslim)

"Tetap tenang, tebarkan senyuman, dan yakinlah semua pasti selesai"

(Hanna Safitri)

KATA PENGANTAR

Puji dan syukur penulis panjatkan kehadirat Allah *Subhanahu Wata'ala* dan Muhammad *Shallallahu'alaihi Wasallam*, karena berkat rahmat dan karunia-Nya semata sehingga penulis mampu menyelesaikan penyusunan Skripsi dengan judul "Klasifikasi *Hate Speech* Menggunakan *Long Short-Term Memory* (LSTM)". Penyusunan skripsi ini dibuat untuk memenuhi salah satu persyaratank kelulusan pada Program Studi Informatika Fakultas Teknik Universitas Khairun. Penyusunan skripsi ini dapat terlaksana dengan baik berkat dukuan dari pihak berbagai pihak. Maka pada kesempatan ini penulis mengucapkan terima kasih banyak kepada:

1. Bapak Dr., M. Ridha Ajam, S.H., M. Hum., selaku Rektor Universitas Khairun.
2. Bapak Ir. Endah Harisun, S.T., M.T., CRP., selaku Dekan Fakultas Teknik Universitas Khairun.
3. Bapak Rosihan, S.T., M.Cs., selaku Koordinator Program Studi Informatika.
4. Bapak Muhammad Fhadli, S.Kom., M.Cs., sebagai Dosen Pembimbing I, yang telah bersedia meluangkan waktunya untuk memberikan bimbingan, dukungan, saran, serta arahan yang berarti bagi penulis dalam penyelesaian skripsi ini.
5. Bapak Ir. Abdul Mubarak, S.Kom., M.T., IPM., sebagai Dosen Pembimbing II, yang telah bersedia meluangkan waktunya untuk memberikan bimbingan, dukungan, saran, serta arahan yang berarti bagi penulis dalam penyelesaian skripsi ini.
6. Bapak Yasir Muin, S.T., M.Kom., selaku Dosen Penguji I, yang telah memberikan masukan, kritikan, dan saran serta dorongannya dalam penyelesaian skripsi ini.
7. Bapak Dr. Muhammad Ridha Albaar, S.Kom., M.Kom., selaku Dosen Penguji II, yang telah memberikan masukan, kritikan, dan saran serta dorongannya dalam penyelesaian skripsi ini.
8. Bapak Hairil Kurniadi Sirajuddin, S.Kom., M.Kom., selaku Dosen Penguji III, yang telah memberikan masukan, kritikan, dan saran serta dorongannya dalam penyelesaian skripsi ini.
9. Bapak Anwar Nada, S.Pd., M.Hum., selaku Koordinator Prodi Bahasa Indonesia dan Sastra yang telah bersedia berpartisipasi membantu penulis dalam menyelesaikan skripsi ini.

10. Dosen Program Studi Informatika yang telah berjasa dalam penyusunan skripsi ini.
11. Kedua orang tua, kakak, adik, serta seluruh keluarga yang selalu memberikan dukungan dan perhatian penuh sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
12. Sahabat penulis Salwa Ammarie, Fitra Soleman, Putri Andini, Maulia Harodian, Puji Rahayu, Kamila Aprilia, Vivi Muliani, dan Endang Pawangka. Terima kasih telah memberikan dukungan, semangat dan dorongan kepada penulis dalam menyelesaikan skripsi ini.
13. Ibu Lurah dan Pegawai Kantor Kelurahan Mangga Dua yang telah memberikan dukungan dan perhatian penuh sehingga penulis dapat menyelesaikan skripsi ini dengan baik.
14. Tak lupa ucapan terima kasih dan semangat kepada teman-teman seperjuangan Angkatan 2019 Prodi Informatika yang memberikan semangat dan dorongan dalam penyelesaian skripsi dan segala bentuk dukunganya selama ini.
15. Semua pihak yang tidak bisa penulis sebutkan satu persatu yang telah membantu penulis baik langsung maupun tidak langsung dalam penyelesaian skripsi ini.
16. Dan yang terakhir, kepada diri saya sendiri Hanna Safitri. Terima kasih sudah bertahan sampai sejauh ini dan bertanggungjawab untuk menyelesaikan apa yang telah di mulai. Terima kasih karena tetap memilih berusaha, tidak menyerah dan merayakan dirimu sendiri sehingga sampai di titik ini.

Akhir kata, dengan penuh kesadaran diri dan segala kerendahan hati, penulis menyadari bahwa masih banyak kekurangan yang ada dalam skripsi ini. Oleh karena itu, penulis sangat menerima saran dan kritik yang membangun dari pihak manapun demi kesempurnaan skripsi ini.

Ternate, 10 Juli 2024

Penulis

DAFTAR ISI

Halaman

DAFTAR GAMBAR

Halaman

DAFTAR TABEL

Halaman

ABSTRAK

KLASIFIKASI *HATE SPEECH* MENGGUNAKAN *LONG SHORT-TERM MEMORY* (LSTM)

Hanna Safitri[1], Muhammad Fhadli[2], Abdul Mubarak[3]
[123]Program Studi Informatika, Fakultas Teknik, Universitas Khairun
Jl. Jati Metro, Kota Ternate Selatan
Email:[1]hannasafitri02@gmail.com, [2]mfhadli@unkhair.ac.id, [3]amuba029@unkhair.co.id

Media sosial adalah platform di dunia maya yang memungkinkan pengguna untuk mempresentasikan diri, berinteraksi, berkolaborasi, berbagi informasi, dan berkomunikasi dengan pengguna lain sehingga membentuk koneksi sosial secara virtual. Platform media sosial yang paling umum digunakan oleh pengguna antara lain *Facebook* dan *Instagram*. Pengguna media sosial mungkin kurang bijak dalam mengekspresikan kebebasan berekspresinya, bahkan tidak jarang mereka melontarkan kata-kata kasar saat menyampaikan pendapatnya di media sosial. Salah satunya adalah adanya kejahatan seperti ujaran kebencian. Ujaran kebencian (*hate speech*) adalah ungkapan langsung atau tidak langsung, baik lisan maupun tulisan, yang diarahkan pada suatu sasaran yang mengandung kebencian. Tujuan dari penelitian ini adalah untuk mengklasifikasi *hate speech* menggunakan metode *Long Short-Term Memory* (LSTM). Pengklasifikasian komentar *hate speech* dan *non hate speech* yang berada pada postingan akun *Facebook* dan *Instagram* Pemerintahan Maluku Utara dengan mencari nilai *recall*, *precision*, dan juga *accuracy* dari model yang dibuat. Hasil yang didapatkan *accuracy* dari perhitungan model klasifikasi yang dibangun adalah 70%, nilai perhitungan tingkat keakuratan dari data yang diminta dengan hasil dari sistem atau *precision*-nya adalah 69%, nilai perhitungan dari keberhasilan sistem dalam menemukan kembali sebuah informasi atau *recall* 72% dan *f1-score* dari perhitungan dari perbandingan antara nilai *precision* dan *recall* adalah 71%.

Kata Kunci: Media Sosial, *Hate Speech*, *Long Short-Term Memory* (LSTM), Maluku Utara

BAB I

PENDAHULUAN

1.1. Latar Belakang

Indonesia adalah negara dengan populasi pengguna internet yang besar dan penggunaan media sosial yang luas. Seiring kemajuan teknologi komunikasi, kita semakin banyak berkomunikasi dan mengakses informasi melalui media tradisional seperti media kertas dan elektronik. Media sosial adalah yang paling berkembang (Hernawati, 2023). Berdasarkan Asosiasi Penyelenggara Jasa Internet Indonesia (APJII) menunjukkan bahwa jumlah pengguna internet di Indonesia saat ini telah mencapai 215,6 juta pengguna internet atau sekitar 78,2% dari total penduduk Indonesia yang mencapai 275,7 juta (APJII, 2023). Bertambahnya pengguna internet setiap tahunnya, berdampak pada meningkatnya jumlah ujaran kebencian yang tersebar di media sosial (Antariksa, 2019). Di Indonesia, jumlah pengguna internet semakin meningkat dari tahun ke tahun. Jumlah pengguna internet di Maluku Utara juga mengalami peningkatan. Semakin banyak pengguna internet menggunakan media sosial maka akan berdampak positif dan negatif terhadap perkembangan teknologi informasi dan komunikasi yang memungkinkan siapa saja dengan mudah mengunggah dan berbagi konten di jejaring sosial. Platform media sosial yang paling umum digunakan oleh pengguna antara lain *whatsapp*, *instagram*, *facebook*, dan *twitter* (Hernawati, 2023) .

Menurut informasi dari Asosiasi Pengguna Jasa Internet Indonesia (APJII) yang malakukan perhitungan data dari tahun 2019 hingga tahun 2020 di Maluku Utara terdapat sekitar 824 ribu pengguna internet. Jumlah ini mengalami peningkatan sebanyak 93 ribu pengguna dibandingkan tahun 2018 yang hanya mencapai 731 ribu pengguna internet

Kerahasiaan (2020). Populasi penduduk di provinsi Maluku Utara diperkirakan sekitar 1.3 juta orang menurut data dari pusat statistik. Jika kita menghitung presentase pengguna internet terhadap total populasi dari tahun 2019 hingga 2022, sekitar 30% dari 1,3 juta, maka diperkirakan ada sekitar 390 ribu orang yang aktif menggunakan internet di Provinsi Maluku Utara. Apabila mempertimbangkan pola penyebaran ujaran kebencian yang menargetkan pengguna internet di platform media sosial, tampaknya sekitar 73.7% pengguna internet dan 61.8% masyarakat yang aktif di media sosial memiliki risiko menjadi sasaran ujaran kebencian. Faktor ini disebabkan oleh popularitas media sosial sebagai sarana komunikasi massa yang sangat efisien dan paling banyak dimanfaatkan pada zaman ini (Malut, 2023).

Media sosial adalah platform di dunia maya yang memungkinkan pengguna untuk mempresentasikan diri, berinteraksi, berkolaborasi, berbagi informasi, dan berkomunikasi dengan pengguna lain sehingga membentuk koneksi sosial secara virtual (Puspitarini, 2019). Pengguna media sosial mungkin kurang bijak dalam mengekspresikan kebebasan berekspresinya, bahkan tidak jarang mereka melontarkan kata-kata kasar saat menyampaikan pendapatnya di media sosial. Salah satunya adalah adanya kejahatan seperti ujaran kebencian (*hate speech*) (Yusuf Sukman, 2017).

Ujaran kebencian (*hate speech*) adalah ungkapan langsung atau tidak langsung, baik lisan maupun tulisan, yang diarahkan pada suatu sasaran yang mengandung kebencian (Fadli, 2019). Penyebaran ujaran kebencian biasanya dilakukan dengan tujuan untuk menyinggung orang meskipun secara tidak langsung. Bahasa kasar (*abusive language*) merujuk pada penggunaan kata-kata kasar yang bisa sangat mengganggu bagi seseorang yang dituju. Istilah lain dari ujaran kebencian adalah kata-kata yang

dimaksudkan untuk menghina, memprovokasi, atau menghasut orang lain, dengan tujuan untuk menimbulkan prasangka baik terhadap pelaku ujaran kebencian maupun korban perbuatannya. (Universitas Muhammadiyah Yogyakarta, 2021).

Di Indonesia ujaran kebencian dapat berbentuk penghinaan, pencemaran nama baik, penistaan, perbuatan tidak menyenangkan, memprovokasi, menghasut, dan menyebarkan berita bohong yang semuanya termasuk tindak pidana yang dapat berdampak pada tindak diskriminasi, kekerasan, penghilangan nyawa, dan atau konflik sosial (Permatasari, 2020).

Dalam Undang-Undang Nomor 11 Tahun 2008 sebagaimana diubah dengan Undang-Undang Nomor 19 Tahun 2016 tentang Perubahan Atas Undang-Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE). Banyaknya ujaran kebencian yang beredar secara online di media sosial, maka dari pihak korban maupun pengguna media sosial yang melihat kata-kata yang tidak pantas atau kata-kata yang mengandung ujaran kebencian tentu sangat merasa tidak nyaman.

Pada penelitian sebelumnya oleh Nurvania (2021) dengan judul "Analisis Sentimen Pada Ulasan di *TripAdvisor* Menggunakan Metode *Long Short-Term Memory* (LSTM)", dalam penelitian tersebut menggunakan *dataset* dari *Tripadvisor* sebanyak 600 data. Hasil pengujian pada penelitian ini menggunakan *confusion matrix* dan mendapatkan nilai akurasi sebanyak 71,67%, *precision* sebanyak 72,97%, *recall* sebanyak 95,3%, dan F1-*Score* sebanyak 82,7%.

Berdasarkan penjabaran diatas, terdapat permasalahan yang harus diteliti karena banyaknya ujaran kebencian (*hate speech*) di media sosial *instagram* dan *facebook* dapat menimbulkan konflik, mempengaruhi opini serta sikap masyarakat terhadap suatu

individual tau kelompok. Solusi dari permasalahan ini adalah dengan menerapkan salah satu algoritma *deep learning* yaitu *Long Short-Term Memory* (LSTM) untuk mengklasifikasi ujaran kebencian *(hate speech)* yang dapat merugikan suatu individual atau kelompok pada media sosial. Maka penulis akan melakukan penelitian yaitu "Klasifikasi *Hate Speech* Menggunakan *Long Short-Term Memory* (LSTM)".

1.2. Rumusan Masalah

Berdasarkan latar belakang diatas, maka rumusan masalah dalam penelitian ini adalah bagaimana klasifikasi *hate speech* menggunakan metode *Long Short-Term Memory* (LSTM).

1.3. Batasan Masalah

Adapun batasan masalah dari penelitian adalah sebagai berikut:

1. Data penelitian diperoleh dari komentar-komentar mengenai kasus dalam lingkup pemerintahan Maluku Utara pada postingan *facebook* dan *instagram.*
2. Komentar yang digunakan hanya berupa teks, tidak mengandung gambar.
3. Penelitian ini hanya mengklasifikasi *hate speech* menggunakan metode *Long Short-Term Memory* (LSTM).

1.4. Tujuan Penelitian

Adapun tujuan dari penelitian ini adalah untuk mengklasifikasi *hate speech* menggunakan metode *Long Short-Term Memory* (LSTM).

1.5. Manfaat Penelitian

Adapun manfaat dari penelitian ini adalah sebagai berikut:

1. Bagi pihak pembaca agar dapat memperoleh informasi terkait dengan *hate speech* di media sosial.

2. Bagi pihak peneliti agar dapat mengimplementasikan ilmu yang diperoleh di bangku perkuliahan.
3. Bagi pemerintahan agar dapat meningkatkan efisiensi, transparansi serta mengevaluasi program kerja yang telah dibuat oleh pemerintah yang dapat mempengaruhi masyarakat dalam memberikan komentar *hate* pada media sosial.

1.6. Sistematika Penulisan

Sistematika penulisan laporan ini merupakan pembahasan singkat dari setiap bab yang menjelaskan hubungan antara bab satu dengan bab lainnya, yaitu sebagai berikut.

BAB I PENDAHULUAN

Bab ini menguraikan tentang latar belakang masalah, rumusan masalah, maksud dan tujuan, batasan masalah, manfaat penelitian dan sistematika penelitian.

BAB II TINJAUAN PUSTAKA

Bab ini menguraikan teori-teori yang berkaitan judul penulis, hal yang untuk memberikan landasan teori sesuai dengan data-data yang diperoleh atau didapat.

BAB III METODE PENELITIAN

Bab ini menjelaskan cara pelaksanaan kegiatan penelitian, mencakup cara pengumpulan data, cara analisa data, serta penerapan metode LSTM.

BAB IV HASIL DAN PEMBAHASAN

Bab ini menjelaskan hasil yang telah dilakukan yang terdiri dari analisis data, pengujian metode dan evaluasi metode menggunakan *confusion matrix.*

BAB V PENUTUP

Bab ini memuat kesimpulan dari hasil penelitian yang dilakukan dan saran untuk penelitian selanjutnya.

BAB II

TINJAUAN PUSTAKA

2.1. Penelitian Terkait

Penelitian terkait bertujuan sebagai referensi dan rujukan terhadap hasil penelitian sebelumnya yang berkaitan dengan penelitian yang akan dilakukan. Adapun beberapa penelitian yang telah dilakukan mengenai klasifikasi *hate speech* menggunakan metode *long-short term memory*. Ini perlu dipahami untuk menjadi sebuah referensi maupun perbandingan dengan metode yang sedang digunakan sekarang. Berikut beberapa penelitian terkait dapat dilihat pada tabel 2.1. sebagai berikut:

Tabel 2.1. Perbandingan Penelitian Terkait

No	Nama dan Tahun	Judul	Hasil
1.	(Fadli, 2019)	Identifikasi *Cyberbullying* pada media sosial *twitter* menggunakan metode LSTM dan BiLSTM	*Cyberbullying* merupakan masalah yang harus menjadi perhatian penting bagi masyarakat. Dalam kasus ini, sumber data penelitian ini berasal dari media sosial *Twitter*. Setidaknya ada 6835 data yang telah dikumpulkan. Data tersebut terdiri dari dua jenis cuitan dengan masing-masing cuitan memiliki kecenderungan *cyberbullying* dan *noncyberbullying*. Tujuan penelitian akan tercapai dengan melakukan beberapa langkah yaitu pengumpulan data, *preprocessing*, klasifikasi, evaluasi, dan diakhiri dengan deteksi konten. Dua algoritma *deep learning* diimplementasikan dalam penelitian ini, yaitu LSTM dan BiLSTM. Pada penelitian ini digunakan untuk melihat model yang terbaik dalam klasifikasi dengan melihat beberapa parameter yaitu *accuracy, pression, recall*, dan F1-*Score*. Hasil yang didapat pada penelitian ini relatif sama dimana algoritma LSTM sebanyak 93,77% dan algoritma BiLSTM sebanyak 95,24%.

			Lalu, untuk nilai F1-*Score* dari masing-masing algoritma yaitu LSTM 92,02% dan BiLSTM 93,84%.
2.	(Isnain, 2022)	Analisis perbandingan algoritma LSTM dan *Naïve Bayes* untuk analisis sentimen	Analisis sentimen adalah bentuk representasi dari *text mining* dan *text processing*. Pada penelitian ini melakukan perbandingan kinerja metode *Long Short-Therm Memory* dengan *Naïve Bayes* terhadap analisis sentimen Kebijakan *New* Normal. Data yang digunakan dalam tahapan pemrosesan sebanyak 1.823 data. Kemudian data tersebut diberi label dengan nilai 1 sebagai sentimen positif dan nilai 0 sebagai sentimen negatif. Total *dataset* yang digunakan berjumlah 1.590 data. Pada pengujian ini, LSTM memiliki nilai akurasi yaitu 83,33% sedangkan *naïve bayes* memiliki nilai akurasi sebesar 82%. Hal ini menunjukkan bahwa LSTM menghasilkan nilai akurasi yang lebih baik dibandingkan dengan *naïve bayes.*
3.	(Ihsan, 2021)	*Algoritme decision tree* untuk mendeteksi ujaran kebencian dan bahasa kasar multilabel pada Twitter berbahasa Indonesia	Pada penelitian ini, *dataset* yang digunakan sebanyak 13.126 twitt asli dari *twitter*. Fitur *leksikon* di klasifikasi *Decision Tree* menghasilkan akurasi tertinggi untuk deteksi ketiga kelas, yaitu kelas ujaran kebencian, kata-kata kasar dan level ujaran kebencian, daripada rekayasa fitur khusus dan fitur tekstual. Rata-rata akurasi dari ketiga kelas meningkat dari 69,77% menjadi 70,48% untuk komposisi data latih-uji 90:10, dan dari 69,35% menjadi 69,54% untuk komposisi 80:20.
4.	(Nurvania, 2021)	Analisis Sentimen Pada Ulasan di *TripAdvisor* Menggunakan Metode *Long Short-Term Memory* (LSTM)	Salah satu informasi yang diperlukan oleh masyarakat adalah informasi mengenai sebuah tempat wisata. Media yang berkontribusi besar dalam penyebaran informasi ini adalah situs *web*. Dalam penyebarannya, informasi dapat dibagi menjadi dua jenis yaitu informasi yang negatif, maupun positif. Analisis sentimen digunakan dalam pengolahan paragraf yang berisi kalimat menggunakan bahasa sehari-hari manusia menjadi bahasa komputer. Penelitian ini bertujuan untuk mengklasifikasikan ulasan pengunjung

			tentang pengaruh COVID-19 terhadap tempat wisata di Bali dari *Tripadvisor* menggunakan metode *Long Short-Term Memory* (LSTM). Sebelum diproses dengan LSTM, setiap teks pada ulasan akan divektorisasi dengan *word2vec*. Pada penelitian ini menggunakan *dataset* dari *Tripadvisor* sebanyak 600 data dengan pembagian data 80% untuk data latih dan 20% untuk data uji. Hasil pengujian menggunakan *confusion matrix* mendapatkan nilai akurasi sebanyak 71,67%, *precision* sebanyak 72,97%, *recall* sebanyak 95,3%, dan F1-*Score* sebanyak 82,7%.
5.	(Bert, 2023)	Deteksi *Hate Speech* Pada *Twitter* Menggunakan Algoritma *Bert*	*Hate speech* atau ujaran kebencian pada salah satu platform sosial media yaitu *Twitter* sudah tidak jarang ditemukan. Pada platform *Twitter*, pengguna bebas mendapatkan, bertukar informasi, serta mengungkapkan opini. Hal ini merupakan salah satu faktor utama seseorang dapat terkena ujaran kebencian pada *Twitter*. Pada penelitian ini, dilakukan proses simulasi menggunakan *website* beserta dengan pengujian dan analisis terhadap pendeteksian ujaran kebencian. Pengujian dilakukan dengan cara pengguna akan melakukan *input* kalimat pada *website hate speech*, lalu *website* akan melakukan *preprocessing* dan menganalisa kalimat tersebut menggunakan Algoritma *BERT* untuk mengklasifikasikan apakah kalimat tersebut termasuk *hate speech* atau tidak. Pada penelitian ini menggunakan algoritma *Bert* untuk mengklasifikasikan apakah kalimat tersebut termasuk *hate speech* atau tidak. Dari hasil pengujian mendapatkan akurasi sebesar 78,69%, presisi sebesar 78,90%, *recall* sebesar 78,69%, dan F1-*score* sebesar 78,77%.

Perbedaan dari penelitian ini dengan penelitian terdahulu yang terdapat dalam tabel 2.1 oleh Ihsan (2021*)* dan *Bert* (2023) pada penelitian tersebut menggunakan metode

yang berbeda yaitu *Algoritme Decision Tree* dan Algoritma Bert untuk mengetahui ujaran kebencian/*hate speech* pada *Twiiter*. Penelitian yang dilakukan oleh Fadli (2019) menggunakan metode LSTM dan BiLSTM mendapatkan hasil BiLSTM lebih akurat dengan presentase lebih tinggi dari pada LSTM dimana algoritma LSTM sebanyak 93,77% dan algoritma BiLSTM sebanyak 95,24%. Dari Isnain (2022) hasil penelitian metode LSTM memiliki nilai akurasi yaitu 83,33% sedangkan *naïve bayes* memiliki nilai akurasi sebesar 82%. Hal ini menunjukkan bahwa LSTM menghasilkan nilai akurasi yang lebih baik dibandingkan dengan *naïve bayes*. Kemudian pada penelitian yang dilakukan oleh Nurvania (2021) Analisis Sentimen Pada Ulasan di *TripAdvisor* menggunakan metode *Long Short-Term Memory* (LSTM) perbedaannya adalah dengan *TripAdvisor* penyebaran informasi dapat dibedakan mana informasi yang positif dan negatif. Sedangkan pada penelitian saya dengan menggunakan metode yang sama yaitu LSTM dapat membedakan mana kata yang termasuk ujaran kebencian dan yang bukan termasuk ujaran kebencian.

Pada penelitian yang saya lakukan ini berfokus metode LSTM untuk mengklasifikasi teks ujararan kebencian/*hate speech* dengan data set berjumlah 1.200 komentar dengan pembagian data latih sebesar 80% dan data uji sebesar 20%. Objek dari penelitian ini adalah *facebook* dan *instagram*.

2.2. *Hate Speech*

Ujaran kebencian (*hate speech*) dalam Kamus Besar Bahasa Indonesia (KBBI) terdiri dari dua suku kata yakni ujar-an (perkataan yang diucapkan) dan benci, ke-benci-an (perasaan benci). Istilah ujaran kebencian (*hate speech*) sendiri merupakan suatu ungkapan yang dilakukan oleh seseorang individu atau kelompok secara langsung maupun tidak langsung yang mengandung kebencian berdasarkan ras, warna kulit,

gender, orientasi seksual, agama dan lain sebagainya (Ihsan, 2021).

Dalam Surat Edaran Nomor SE/06/X/2015 tentang Penanganan Ujaran Kebencian yang dikeluarkan oleh Kapolri menyebutkan bahwa ujaran kebencian adalah perbuatan yang dapat berupa tindak pidana yang diatur dalam Kitab Undang-Undang Hukum Pidana (KUHP) yang berbentuk penghinaan, pencemaran nama baik, penistaan, perbuatan tidak menyenangkan, memprovokasi, menghasut, dan penyebaran berita bohong bila ditujukan pada seseorang pribadi yang dapat merugikan secara langsung maupun tidak langsung (Parulian, 2022).

2.3. *Text Mining*

Text mining adalah salah satu bidang khusus dari data *mining.* Teks *mining* dapat didefinisikan sebagai suatu proses pengambilan informasi dimana pengguna berinteraksi dengan kumpulan dokumen menggunakan tools analisis yang merupakan komponen dalam data *mining* yang salah satunya adalah klasifikasi (Ariyanti, 2020).

Dalam memberikan solusi, teks *mining* menerapkan dan mengembangkan banyak teknik dari bidang lain seperti data *mining*, *information retrieval, statistic and matematik, machine learning, linguistic, natural languange processing, and visualization* (Ariyanti, 2020).

Tujuan dari teks *mining* adalah untuk memperoleh informasi yang berguna dari sekumpulan dokumen. Jadi, sumber data yang digunakan dalam teks *mining* adalah kumpulan teks yang memiliki format yang tidak terstruktur atau setidaknya semi terstruktur. Tugas khusus dari teks mining meliputi mengklasifikasikan teks dan mengelompokkan teks (Ariyanti, 2020).

Teks yang akan dilakukan proses teks mining, pada umumnya memiliki beberapa

karakteristik antara lain ukuran yang tinggi, terdapat *noise* pada data, dan struktur teks yang tidak baik. Cara yang digunakan untuk mempelajari suatu data teks adalah dengan terlebih dahulu mengidentifikasi fitur-fitur yang mewakili setiap kata untuk setiap fitur pada dokumen. Sebelum mengidentifikasi fitur-fitur yang representatif diperlukan tahap praproses yang dilakukan secara umum dalam teks mining, yaitu *case folding, tokenizing, filtering, stemming, tagging* dan *analyzing* (Ariyanti, 2020).

2.4. *Natural Language Processing*

Natural Language Processing (NLP) merupakan salah satu ilmu dalam *Artificial Intelligence* (AI) yang mengembangkan pembelajaran bahasa alami untuk memproses atau menerjemahkan kata-kata dari manusia yang tidak terstruktur agar dapat dipahami oleh komputer. Bahasa ini tidak mudah diaplikasikan pada komputer karena dibutuhkan pemrosesan yang cukup lama dalam menerjemahkan bahasa tersebut (Bert, 2023).

Natural Language Processing (NLP) adalah kombinasi bidang ilmu komputer dan bidang kecerdasan buatan yang berkaitan dengan linguistik. NLP berkaitan dengan bagaimana mesin memahami bahasa manusia untuk saling berinteraksi satu sama lain. Dengan adanya NLP, komputer dapat mempelajari dan memahami bahasa manusia untuk dapat berkomunikasi dengan manusia. Bahasa manusia itu unik karena dibuat khusus untuk menyampaikan makna. Membuat komputer untuk memahami bahasa manusia adalah tugas yang sulit, karena bahasa manusia memiliki struktur yang kompleks. Selain itu, setiap bahasa memiliki keunikannya masing-masing dan dapat memiliki banyak arti. Sebagai contoh dapat dilihat dari kalimat berikut, *"Look at the dog with one eye*", di mana kalimat tersebut dapat memiliki arti "melihat anjing dengan satu mata" atau "melihat anjing yang mempunyai mata satu" (Prasetyo, 2021).

Dua teknik utama untuk memahami NLP adalah analisis sintaksis dan analisis semantik. Kedua teknik tersebut digunakan untuk mengkaji struktur bahasa. Analisis sintaksis mengacu pada tata bahasa, sedangkan analisis semantik mengacu pada penafsiran suatu kalimat (Prasetyo, 2021).

Analisis sintaksis adalah teknik menyusun pada suatu kalimat sehingga kalimat memiliki tata bahasa yang benar. Analisis sintaksis melibatkan penentuan stuktur kalimat seperti subjek, predikat, kata benda, kata kerja, kata ganti, dan lain-lain. Sistem akan dapat membaca kalimat masukan yang akan dipecah menjadi kata-kata dan akhirnya menghasilkan deskripsi yang terstruktur. Teknik ini dapat digunakan untuk menyederhanakan kalimat agar informasi lebih mudah ditemukan. Selain itu, penggunaan analisis sintaksis juga dapat membantu mendeteksi keberadaan kata atau kalimat baru atau tidak biasa (Prasetyo, 2021).

2.5. *Text Pre-processing*

Text preprocessing merupakan tahapan penting dalam melakukan proses klasifikasi pada teks. *Text preprocessing* bertujuan untuk membersihkan data yang tidak diperlukan sehingga menjadi data yang baik dan terstruktur sebelum memasuki tahap selanjutnya (Bert, 2023). Adapun tahapan *text preprocessing* adalah sebagai berikut.

1. *Case Folding*

   Case Folding dalam pemrosesan teks adalah penting karena membantu mengatasi masalah variasi huruf besar dan kecil dalam kata-kata yang sama. Pada tahap ini merupakan proses mengubah huruf kapital dalam teks menjadi huruf kecil.

2. *Tokenizing*

   Tokenizing adalah proses memisahkan setiap kata pada teks sehingga tidak lagi

membentuk kalimat yang utuh. Token bisa berupa kata, frasa, atau bahkan karakter individu.

3. *Stopwords Removal*

   Stopwords Removal adalah proses menghilangkan kata-kata yang sering muncul dalam teks tetapi tidak memberikan makna yang signifikan atau informasi kontekstual penting.

4. *Stemming*

   Stemming adalah proses mengubah kata-kata yang terinfleksi atau memiliki variasi bentuk kata (seperti kata-kata berimbuhan) ke bentuk dasar atau "akar" kata.

2.6. *Long Short-Term Memory*

Metode LSTM pertama kali dibuktikan pada tahun 1997 oleh Hochreiter dan Schmidhuber untuk mengolah data yang panjang. LSTM adalah versi terbaru yang lebih efisien dan memiliki keuntungan yang lebih dibandingkan dengan *Recurrent Neural Network* (RNN). Pada LSTM meskipun terdapat jarak antar teks analisis masih bisa dilakukan. Hal ini terjadi karena LSTM memiliki memori yang menyimpan data di masa lalu dan juga data dapat disimpan untuk jangka waktu yang lebih lama. Jaringan LSTM memiliki ukuran yang besar, jaringan ini terdapat LSTM *cell* yang menggantikan posisi lapisan tersembunyi di RNN yang berguna untuk menyimpan informasi sebelumnya (Nurvania, 2021). Metode LSTM digunakan dengan cara mengklasifikasi data dalam jangka panjang dengan menyimpan pada sel memori (Astari, 2021). Pada jaringan *Long Short-Term Memory* (LSTM) terdapat beberapa *gate* yaitu forget *gate*, *input gate*, output *gate* dan *memory cell* yang akan menghitung nilai keluaran sebagai *hidden layer* untuk jaringan selanjutnya.

Berikut adalah penjelasan dan rumus terkait *gate* yang ada di dalam *memory cell* LSTM:

1. *Forget Gate*

Langkah pertama pada LSTM bertujuan untuk menentukan informasi apa yang akan dilupakan dalam *cell state.* Pada gerbang ini nilai output sebelumnya dengan *input* saat ini digabung lalu melewati fungsi aktivasi *sigmoid*. Gerbang inilah yang menentukan apakah informasi sebelumnya akan dilupakan atau tidak. Kemudian informasi ini dilanjutkan ke *memory cell* atau *cell* state (Nurvania, 2021). Persamaan *forgate gate* dapat dilihat pada rumus 2.1 sebagai berikut:

$$f_t = \sigma\left(W_f \times [x_t + h_{t-1}] + b_f\right) \text{.....(2.1)}$$

Keterangan:

f_t : *Forgate gate*

σ : Fungsi *sigmoid*

W_f : Nilai bobot untuk *forgate gate*

X_t : Nilai *input* sel

h_{t-1} : Nilai keluaran sebelum orde ke t

b_f : Nilai bias pada *input gate*

2. Input Gate

Input gate bertujuan untuk menentukan informasi baru yang akan disimpan pada *cell state*. Pada gerbang ini nilai *output* sebelumnya dengan *input* saat ini digabung, lalu ada dua fungsi aktivasi yang akan dilewatinya. Jalur satu melewati fungsi aktivasi *sigmoid* untuk nilai *input*, jalur lainnya melewati fungsi aktivasi tanh untuk nilai *candidate memory cell* (Nurvania, 2021). Persamaan *forgate gate* dapat dilihat pada rumus 2.2 dan 2.3

sebagai berikut:

$$\mathrm{I}_t = \sigma\,(W_i \times [x_t + h_{t-1}] + b_i) \quad \text{.....(2.2)}$$

$$\mathrm{C\tilde{}}_t = \tanh\,(\mathrm{W}_c \times \mathrm{x}_t + h_{t-1}] + b_c) \quad \text{.....(2.3)}$$

Keterangan:

I_t : *Input Gate*

W_i : Nilai bobot untuk *input gate*

b_i : Nilai bias pada *input gate*

$C\tilde{}_t$: Nilai baru yang dapat ditambahkan pada *cell state*

Tanh : fungsi *tanh*

W_C : Nilai bobot untuk *cell state*

b_C : Nilai bias pada *cell state*

3. Cell State

Cell state bertujuan untuk memperbaharui *cell state* yang lama menjadi *cell state* yang baru. Pada tahap ini ada penggabungan dari dua nilai. Nilai pertama adalah nilai dari *forget gate* akan dikalikan dengan nilai dari *cell state* sebelumnya. Nilai kedua adalah nilai dari *input gate* dikalikan dengan nilai dari *candidate memory cell* (Nurvania, 2021). Persamaan *forgate gate* dapat dilihat pada rumus 2.4 sebagai berikut:

$$\mathrm{C}_t = \mathrm{f}_t \times \mathrm{C}_{t-1} + \mathrm{i}_t \times \mathrm{C}_t^{\sim} \quad \text{.....(2.4)}$$

Keterangan:

C_t : *Cell state*

C_{t-1} : Nilai *cell state* sebelum orde ke t

4. Output Gate

Gerbang ini menghasilkan nilai *output*, dimana nilai ini berasal dari gabungan nilai sebelumnya dengan nilai saat ini yang telah melalui fungsi aktivasi *sigmoid* (Nurvania,

2021). Persamaan *forgate gate* dapat dilihat pada rumus 2.5 sebagai berikut:

$$o_t = \sigma(W_o \times [x_t + h_{t-1}] + b_o \quad \text{.....(2.5)}$$

Keterangan:

o_t : *Output gate*

W_o : Nilai bobot untuk *output gate*

b_o : Nilai bias pada *output gate*

5. *Hidden Layer*

Hidden layer berpengaruh untuk nilai di proses selanjutnya, nilai dari *layer* ini berasal dari nilai *output* yang dikalikan dengan nilai dari *cell state* atau *memory cell* yang telah diaktivasi dengan fungsi tangen (Nurvania, 2021). Persamaan *forgate gate* dapat dilihat pada rumus 2.6 sebagai berikut:

$$h_t = o_t \times \tanh C_t \quad \text{.....(2.6)}$$

Keterangan:

h_t : *Hidden state*

2.7. *Confusion Matrix*

Confusion matrix adalah sebuah tabel atau *matrix* yang berisikan empat nilai yang merupakan pengukuran kinerja pada masalah klasifikasi yang telah dilakukan. Ada empat nilai atau *point* pada *confusion matrix* yaitu *true positive, true negative, false positive,* dan *false negative* (Nurvania, 2021). Gambaran *confusion matrix* dapat dilihat pada tabel 2.2. berikut.

Tabel 2.2. Gambaran *Confusion Matrix*

		True Values	
		Positive	*Negative*
Predictions	*Positive*	*True Positive*	*False Positive*
	Negative	*False Negative*	*True Negative*

Keterangan:

TP = Prediksi yang bernilai positif dan benar sesuai target

TN = Prediksi yang bernilai negatif dan benar sesuai target.

FP = Prediksi yang bernilai positif dan salah tidak sesuai target.

FN = Prediksi yang bernilai negatif dan salah tidak sesuai target

1. *Accuracy*

   Accuracy adalah perhitungan seberapa akurat model klasifikasi yang telah dibangun sesuai dengan target yang ada. Dengan kata lain, *accuracy* adalah nilai dari tingkat kedekatan antara nilai prediksi dan nilai yang aktual (Nurvania, 2021). Persamaan dari *accuracy* dapat dilihat pada rumus 2.7 sebagai berikut.

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} x100\% \quad \text{.....(2.7)}$$

2. *Precision*

   Precision adalah perhitungan tingkat keakuratan dari data yang diminta dengan hasil prediksi yang diberikan sistem (Nurvania, 2021). Persamaan dari *precision* dapat dilihat pada rumus 2.8 sebagai berikut.

$$Precision = \frac{TP}{TP+FP} x100\% \quad \text{.....(2.8)}$$

3. *Recall*

   Recall adalah perhitungan yang menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi (Renaldo, 2023). Persamaan dari *recall* dapat dilihat pada rumus 2.9 sebagai berikut.

$$Recall = \frac{TP}{TP+FN} x100\% \quad \text{.....(2.9)}$$

4. F1-*Score*

   F1-*Score* adalah perhitungan yang menggambarkan perbandingan antara *precision*

dan *recall* (Nurvania, 2021). Persamaan dari f1-*score* dapat dilihat pada rumus 2.7 sebagai berikut.

$$F1 - Score = 2x \frac{Precesion\ x\ Recall}{Precesion + Recall} x100\% \quad \text{.....(2.10)}$$

2.8. *Python* Versi 3.10.12

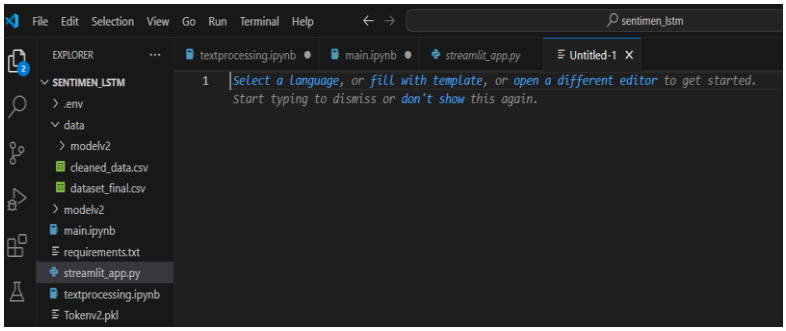
Python merupakan bahasa pemrograman yang pertama kali dirilis oleh Guido Van Rossum pada tahun 1991 di Stichting Mathematisch Centrum Belanda sebagai kelanjutan dari bahasa pemrograman ABC. Nama *python* digunakan oleh Guido sebagai nama bahasa yang diciptanya karena Guido penggemar dari program TV *Monty Python's Flying Circus*. Sehingga ungkapan khas dari program tersebut sering muncul dalam korespondensi antara pengguna *python*. Saat ini *python* sudah mencapai versi 3.12.10. Bahasa pemrograman ini dikembangan bersifat *open source* menggunakan lisensi GFL *compatible* pada sebagian besar versinya (Wahyono, 2018).

Python adalah bahasa pemrograman tingkat lanjut yang mudah dipelajari karena sintaksnya yang jelas dan elegan dikombinasikan dengan penggunaan modul yang mempunyai struktur data yang tinggi, efisien, dan dapat digunakan. *Source code* aplikasi bahasa pemrograman *python* biasanya dikompilasi menjadi format perantara yang disebut *bytecode* yang kemudian akan dieksekusi (Ratna, 2020).

Phyton dinilai mampu menangangani pembuatan aplikasi-aplikasi kekinian yang mengandung kata kunci *big* data, *deep learning*, data *science* hingga *machine learning*. Dengan kata lain, *python* merupakan bahasa pemrograman sederhana untuk membuat aplikasi berbasis kecerdasan buatan *(artificial intelligence). Phyton* dapat digunakan untuk berbagai tujuan pengembangan perangkat lunak dan dapat berjalan di berbagai platform sistem operasi (Jubilee, 2019).

2.9. Visual Studio *Code*

Visual studio *code* atau dikenal dengan VS*Code* adalah *software* untuk membuat kode atau *coding* yang dapat diakses di berbagai sistem operasi seperti *windows, linux,* dan *macOS*. VS*Code* dikembangkan oleh *Microsoft* dan pertama kali diperkenalkan pada konferensi *Build* pada tanggal 29 April 2015. Vs*code* mendukung berbagai bahasa pemrograman seperti *java, javascript*, C, C++, *python*, dan masih banyak lagi (Salendah, 2022). Tampilan vs*code* dapat dilihat pada gambar 2.1 berikut.



Gambar 2.1. Tampilan Visual Studio *Code*

Adapun langkah-langkah dalam menggunakan visual studio *code* sebagai berikut:

1. Pertama yaitu buka visual studio *code.*
2. Pilih *file* > *new file.*
3. Kemudian masukkan kode program dan simpan *file* dengan format .py.
4. Jalankan program dengan cara klik pada logo segitiga di sisi kanan atas editor.
5. Selanjutnya hasil program akan ditampilkan.

2.10. *Figma*

Figma adalah alat desain yang biasa digunakan untuk membuat tampilan aplikasi *mobile, desktop, website* dan lain-lain. *Figma* dapat digunakan pada sistem operasi *windows, linux* atau *mac* dengan koneksi internet. Keuntungan *figma* adalah

memungkinkan banyak orang melakukan tugas yang sama pada waktu yang sama meskipun berada di lokasi yang berbeda.

Ini bisa disebut kerja kelompok dan karena kemampuan *figma* menjadikannya pilihan pertama banyak desain UI/UX untuk membuat prototipe situs *web* dan aplikasi dengan cepat dan efektif.

2.11. *User Interface* dan *User Experience* (UI/UX)

Menurut Solikin (2022) *User Interface* dan *User Experience* (UI/UX) memiliki peranan penting dalam pembuatan suatu aplikasi/sistem karena desain aplikasi harus tertata dengan baik agar pengguna dapat dengan mudah menggunakan fitur-fitur yang disediakan oleh aplikasi/sistem. Desain UI/UX harus sesuai dengan kebutuhan pengguna aplikasi/sistem yang anda buat, termasuk tampilan, fungsional, dan berbagai persyaratan lainnya (Al-Faruq, 2022).

UI adalah sistem yang digunakan pengguna dan juga dapat mendengar, melihat dan menyentuh. Artinya suatu sistem yang dapat mengontrol tampilan antarmuka pengguna sekaligus memudahkan pengguna dalam berinteraksi dengan sistem (Al-Faruq, 2022).

Menurut Utama (2020) UX adalah sistem yang mendefinisikan pengalaman yang dimiliki pengguna saat menggunakan perangkat lunak dan mengevaluasi tingkat kemudahan penggunaan dan kenyamanan yang terkait dengan fungsionalitas perangkat lunak (Al-Faruq, 2022).

2.12. *Streamlit*

Streamlit adalah kerangka *web* untuk menyebarkan model dan visualisasi dengan mudah menggunakan *python* dengan cepat. Ada widget bawaan untuk masukan pengguna seperti unggahan gambar, penggeser, masukkan teks, dan elemen HTML lainnya seperti

kotak centang dan tombol radio. Setiap kali pengguna berinteraksi dengan *streamilit, script python* dijalankan ulang dari atas ke bawah (Widi Hastomo, 2022).

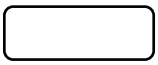
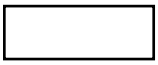
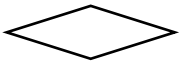
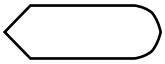
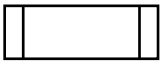
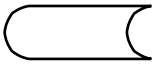
Streamlit adalah aplikasi gratis dan pengguna tidak memerlukan pengetahuan lanjutan tentang pengembangan *front-end* untuk menggunakannya. *Streamlit* dapat dijalankan pada editor *vscode* 3.7 ke atas tetapi tidak mendukung editor *jupyter* dan *notebook* sehingga harus dikonversi ke editor vs*code* (Widi Hastomo, 2022)..

2.13. *Flowchart*

Menurut Rosaly (2019) *Flowchart* atau sering disebut dengan diagram alir merupakan salah satu jenis diagram yang mewakili suatu algoritma atau langkah-langkah instruksi yang berurutan dalam suatu sistem. Seorang analis sistem menggunakan *flowchart* sebagai bukti dokumentasi untuk menjelaskan kepada programmer gambaran logis dari sistem yang akan dibangun.

Dengan begitu, *flowchart* dapat membantu untuk memberikan solusi terhadap masalah yang bisa saja terjadi dalam membangun sistem. Pada dasarnya, *flowchart* digambarkan dengan menggunakan simbol-simbol. Setiap simbol mewakili suatu proses tertentu. Sedangkan untuk menghubungkan satu proses ke proses selanjutnya digambarkan dengan menggunakan garis penghubung. Dengan adanya *flowchart*, setiap urutan proses dapat digambarkan menjadi lebih jelas. Selain itu, ketika proses baru dapat ditambahkan, proses tersebut dengan mudah diimplementasikan menggunakan diagram ini. Setelah proses pembuatan flowchart selesai, maka giliran *programmer* yang akan menerjemahkan rancangan logis tersebut ke dalam bentuk program dengan berbagai bahasa pemrograman yang telah disepakati (Rosaly, 2019). Simbol-simbol *flowchart* dapat dilihat pada tabel 2.3. berikut:

Tabel 2.3. Simbol-simbol *Flowchart*

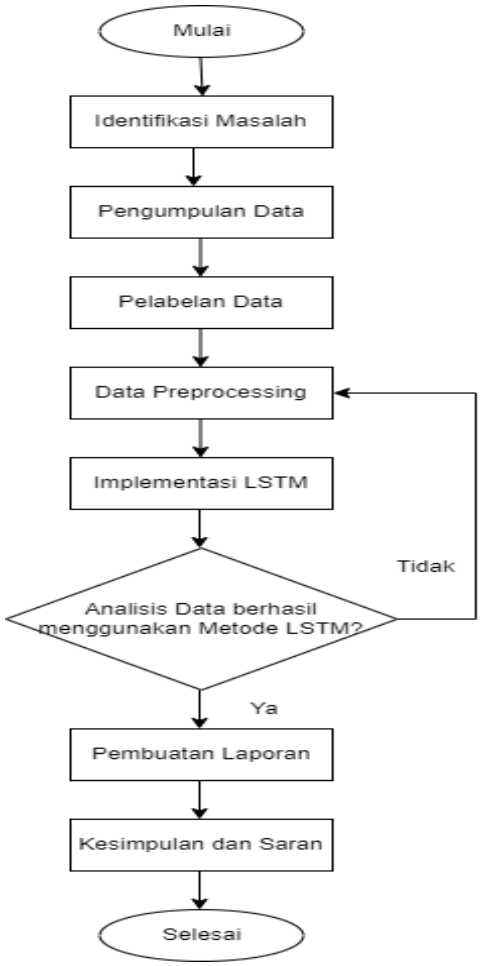
No	Simbol	Fungsi
1		Terminal, untuk memulai dan mengakhiri suatu proses kegiatan.
2		Proses, suatu yang menunjukan setiap pengolahan yang dilakukan oleh komputer.
3		*Input*, untuk memasukkan hasil dari suatu proses.
4		*Decision*, suatu kondisi yang akan menghasilkan beberapa kemungkinan jawaban atau pilihan.
5		*Display, output* yang ditampilkan di layar terminal.
6		*Connector*, suatu prosedur akan masuk atau keluar melalui simbol ini dalam lembar yang sama.
7		*Off Page Connector*, merupakan simbol masuk atau keluarnya suatu prosedur pada kertas lembar lain.
8		Arus *Flow*, simbol ini digunakan untuk menggambarkan arus proses dari suatu kegiatan lain.
9		*Hard Disk Storage, input output* yang menggunakan *hardisk.*
10		*Predefined Process*, untuk menyatakan sekumpulan langkah proses yang ditulis sebagai prosedur.
11		*Storad* data, *input*, *output* yang menggunakan disket.
12		*Printer*, simbol ini digunakan untuk menggambarkan suatu dokumen atau kegiatan untuk mencetak informasi dengan mesin *printer*.

BAB III

METODE PENELITIAN

3.1. Tahapan Penelitian

Tahapan penelitian dapat dilihat pada gambar 3.1. berikut.



Gambar 3.1. Tahapan Penelitian

Berikut adalah uraian rencana tahapan penelitian yaitu sebagai berikut:

1. Identifikasi Masalah

   Pada tahap ini peneliti melakukan identifikasi masalah dari latar belakang dan perumusan masalah yang ada. Permasalahan tersebut terkait dengan maraknya ujaran kebencian di media sosial.

2. Pengumpulan Data

   Pengumpulan data yang dilakukan yaitu diperoleh dari komentar/ulasan dari *instagram* dan *facebook* yang ditinggalkan oleh pengguna pada suatu postingan kemudian komentar tersebut disalin ke dalam *file excel* yang telah dibuat.

3. Pelabelan Data

   Pelabelan data digunakan untuk melabeli data komentar untuk melihat apakah data tersebut termasuk dalam *hate speech* atau *non hate speech*.

4. Data *Preprocessing*

   Pada tahap ini merupakan langkah awal sebelum data diolah yang bertujuan untuk membersihkan data mentah berupa cuitan-cuitan sehingga menjadi data yang lebih terstruktur.

5. Implementasi LSTM

   Pada tahap ini dilakukan proses pembuatan model LSTM yang terdiri dari *forget gate*, *input gate*, memori sel, dan *output gate.*

6. Analisis Data Menggunakan Metode *Long Short-Term Memory*

   Pada tahap ini, data yang telah diperoleh dan dilabeli kemudian akan dianalisis dan diolah berdasarkan alur metode *Long Short-Term Memory.*

7. Pembuatan Laporan

   Setelah selesai mengumpulkan data dan mengimplementasikan ke metodenya, selanjutnya membuat laporan dari hasil yang didapatkan.

8. Kesimpulan dan Saran

   Pada tahap ini, melakukan analisis yang mendalam terhadap data untuk mendapatkan hasil yang akurat dan juga dapat menarik kesimpulan serta

memberikan saran.

3.2. Objek dan Waktu Penelitian

Objek yang penulis gunakan dalam penelitian ini adalah komentar-komentar dari media sosial *instagram* dan *facebook.* Penelitian ini dilakukan pada semester genap tahun ajaran 2022-2023.

3.3. Alat dan Bahan Penelitian

Dalam melakukan penelitian ini, ada beberapa spesifikasi alat penelitian yang harus dipenuhi. Spesifikasi alat maksudnya adalah standar minimal dari alat *(tools)* yang digunakan sebagai wadah utama untuk melakukan penelitian ini, spesifikasi alat antara lain sebagai berikut:

3.3.1. Detail Spesifikasi *Hardware*

Adapun spesifikasi perangkat keras *(hardware)* yang digunakan pada penelitian ini sebagai penunjang proses perancangan dan pembuatan sistem selama penelitian berlangsung. Dapat dilihat pada tabel 3.1. berikut:

Tabel 3.1. Detail Spesifikasi *Hardware*

No	Jenis	Spesifikasi
1	PC	LAPTOP-SK31NREO
2	*Processor*	Intel® Core™ i3-10110u CPU @ 2.1GHz 2.50 GHz
3	Memori	4,00 GB
4	SSD	256 GB
6	I/O	Monitor, *keyboard* dan *mouse*

3.3.2. Detail Spesifikasi *Software*

Selain kebutuhan *hardware*, penulis juga membutuhkan kebutuhan *software* untuk melakukan perancangan dalam pembuatan sistem. Dapat dilihat pada tabel 3.2. berikut.

Tabel 3.2. Detail Spesifikasi *Software*

No	Jenis	Spesifikasi	Keterangan
1	OS	*Windows* 11	Sistem operasi yang digunakan saat pengembangan sistem
2	Bahasa Pemrograman	*Python* versi 3.10.12	Digunakan untuk pembuatan sistem
3	Aplikasi Pengolahan Data	*Microsoft Excel*	Untuk mengolah *dataset*
4	*Text Editor*	*Google Colab*	Digunakan untuk menulis *script python*

3.4. Pengumpulan Data

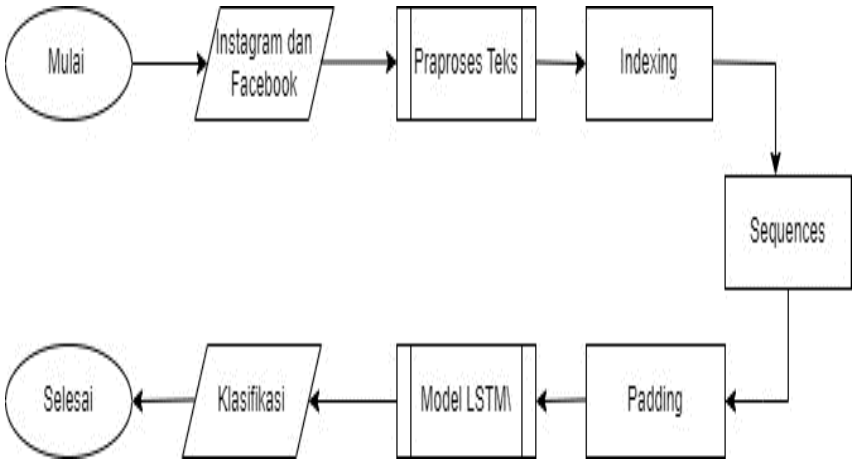
Pada penelitian ini peneliti menggunakan pendekatan kuantitatif dengan metode eksperimen yang dilakukan terhadap variabel yaitu *instagram* dan *facebook* dengan data yang ada terhadap subjek penelitian yaitu komentar-komentar pada postingan *instagram* dan *facebook*. Penelitian ini juga menggunakan jenis data primer dan data sekunder. Data primer adalah pengumpulan informasi dan data yang diperoleh secara langsung dari sumber utama Pramiyati (2017). Sumber data yang diperoleh dalam penelitian ini adalah dengan mewawancarai ahli bahasa dalam melakukan labeling. Sedangkan, data sekunder adalah suatu data yang diperoleh secara tidak langsung. Data yang diperoleh yaitu komentar/ulasan di *instagram* dan *facebook* yang ditinggalkan oleh pengguna pada suatu postingan.

Pengambilan data dilakukan secara manual yaitu dengan cara mengambil komentar-komentar pada suatu postingan *instagram* dan *facebook*. Komentar yang ada pada postingan tersebut disalin ke dalam *file excel* yang telah dibuat untuk dijadikan *dataset*. Jumlah *dataset* sebanyak 1200 komentar kemudian dilakukan pembagian untuk data latih sebesar 80% dengan jumlah 960 komentar dan pembagian data uji sebesar 20% dengan jumlah 240 komentar. Lalu data yang terkumpul akan dibawah ke ahli bahasa

untuk dilakukan wawancara. Dari wawancara tersebut dilakukan validasi untuk mengetahui kata atau kalimat apa saja yang mengandung unsur *hate speech* sehingga data tersebut dapat dilakukan *labelling* apakah komentar tersebut termasuk *hate speech* atau *non hate speech.*

3.5. *Flowchart System*

Flowchart adalah suatu metode yang menggambarkan tahapan-tahapan pemecahan suatu masalah dengan menggunakan simbol-simbol tertentu. Tahapan-tahapan dalam penelitian ini dapat dilihat pada gambar 3.2. berikut.



Gambar 3.2. *Flowchart System* (Udin, 2022)

Pada gambar 3.2. menjelaskan tahapan kerja dari sistem yang akan dibangun. Diawali dengan meng-*input* data dari komentar-komentar pada *instagram* dan *facebook,* kemudian melakukan praproses teks, indexing, *text to sequence*, *padding*, model LSTM, dan yang terakhir *output* berupa klasifikasi teks oleh sistem.

3.5.1. *Input* Data

Pada tahap ini peneliti menyiapkan data komentar dari *instagram* dan *facebook* untuk melakukan peng-*input*-an pada sistem. Adapun contoh data dari *instagram* dan *facebook* pada penelitian ini dapat dilihat pada tabel 3.3. dan tabel 3.4. berikut:

Tabel 3.3. Contoh Kata yang Mengandung Unsur *Hate Speech*

No	Contoh Kata
1	Setang
2	Fuma
3	Goblok
4	Bangsat
5	Golojo

Tabel 3.4. Contoh Komentar pada *Instagram* dan *Facebook*

No	Sumber	Nama Pengguna	Komentar
1	*Instagram*	umikalsumf_	bicara sama dg orang tra skolah ni ketua DPRD lagi ijazah bayar saja kaapa
2	*Facebook*	Udhy Lylha	Ketua DPR tara tau bicara. Ngo yang komunis kerja Cuma tau mkn Gaji Buta
3	*Facebook*	Yamin Salam	Ketua DPR kog bicara sama deg orang gila, catat dpe nama deg kanal dpe orang bae" jang sampe 2024 ni diam o bcalon lgi
4	*Instagram*	dwikurnia.sandi	Anggota DPR satu Kong fuma sampe tinggal lah nakes kase polisikan pa ngana
5	*Instagram*	1901fath	Manusia Goblok

3.5.2. *Text Pre-processing*

Text preprocessing bertujuan untuk menyiapkan dan membersihkan data mentah berupa cuitan-cuitan sehingga menjadi data yang baik dan terstruktur sebelum memasuki tahap selanjutnya. Maksud dari kalimat yang baik dan terstruktur adalah kalimat dalam komentar berupa data mentah yang memiliki huruf kapital, angka serta simbol/tanda baca yang berlebihan. Kemudian diubah, dipisahkan dan dihilangkan untuk menjadi kalimat yang lebih baik dan terstruktur. Tahapan pada *text preprocessing* sebagai berikut:

1. *Case Folding*

   Case Folding dalam pemrosesan teks adalah penting karena membantu mengatasi masalah variasi huruf besar dan kecil dalam kata-kata yang sama. Pada tahap ini

merupakan proses mengubah huruf kapital dalam teks menjadi huruf kecil. Misalnya, “Ibu” dan “ibu” dianggap berbeda oleh komputer jika tidak dilakukan *case folding,* padahal sebenarnya kedua mengacunya pada hal yang sama. Huruf yang digunakan yaitu huruf ‘a’ sampai ‘z’, selain dari huruf-huruf itu maka dihilangkan karena dianggap delimiter. Dengan melakukan *case folding,* teks menjadi konsisten dalam formatnya, mempermudah analisis dan pengolahan lebih lanjut. Dapat dilihat pada tabel 3.5. berikut.

Tabel 3.5. Contoh Penerapan *Case Folding*

Teks	-Ibu membuatkan sarapan untuk ayah -Fita pergi ke rumah kakek sejak tiga minggu lalu
Case Folding	-ibu membuatkan sarapan untuk ayah -fita pergi ke rumah kakek sejak tiga minggu lalu

2. *Tokenizing*

Tokenizing adalah proses memisahkan setiap kata pada teks sehingga tidak lagi membentuk kalimat yang utuh. Token bisa berupa kata, frasa, atau bahkan karakter individu. Tahap *tokenizing* juga meliputi *removing number* dan *removing punctuation*. Dapat dilihat pada tabel 3.6. berikut.

Tabel 3.6. Contoh Penerapan *Tokenizing*

Teks	-ibu membuatkan sarapan untuk ayah -fita pergi ke rumah kakek sejak tiga minggu lalu
Tokenizing	[‘ibu’, ‘membuatkan’, ‘sarapan’, ‘untuk’, ‘ayah’] [‘fita’, ‘pergi’, ‘ke’, ‘rumah’, ‘kakek’, ‘sejak’, ‘tiga’, ‘minggu’, ‘lalu’]

3. *Stopwords Removal*

Stopwords Removal adalah proses menghilangkan kata-kata yang sering muncul dalam teks tetapi tidak memberikan makna yang signifikan atau informasi kontekstual penting. Contoh kata *stopwords* yaitu ‘untuk’, ‘dan’, ‘serta, juga, dan

lainnya. Dengan melakukan *stopwords removal* ini dapat mengurangi *noise* serta mempercepat waktu pemrosesan. Dapat dilihat pada tabel 3.7. berikut.

Tabel 3.7. Contoh Penerapan *Stopwords Removal*

Teks	-ibu membuatkan sarapan untuk ayah -fita pergi ke rumah kakek sejak tiga minggu lalu
Stopword Removal	-ibu membuatkan sarapan ayah -fita pergi ke rumah kakek tiga minggu lalu

4. *Stemming*

Stemming adalah proses mengubah kata-kata yang terinfleksi atau memiliki variasi bentuk kata (seperti kata-kata berimbuhan) ke bentuk dasar atau "akar" kata. Dapat dilihat pada tabel 3.8. berikut.

Tabel 3.8. Contoh Penerapan *Stemming*

Teks	-ibu membuatkan sarapan untuk ayah -fita pergi ke rumah kakek sejak tiga minggu lalu
Stemming	-ibu buat sarapan untuk ayah -fita pergi rumah kakek sejak tiga minggu lalu

3.5.3. *Indexing*

Indexing adalah suatu proses mengubah atau mengubah setiap kata (token) yang terdapat pada teks menjadi bentuk bilangan numerik. Tahapan ini berperan penting dalam mengonversi setiap kata pada teks menjadi format numerik agar dapat di proses oleh model. Hal ini disebabkan karena setiap model *machine learning* atau *deep learning* hanya dapat memproses data dalam bentuk numerik atau angka-angka. Dapat dilihat pada tabel 3.9. berikut.

Tabel 3.9. Contoh Penerapan *Indexing*

Teks	*Indexing*

-ibu membuatkan sarapan untuk ayah -fita pergi ke rumah kakek sejak tiga minggu lalu	{'UNK':0, 'ibu':1, 'membuatkan':2, 'sarapan':3, 'untuk':4, 'ayah':5, 'fita':6, 'pergi':7, 'ke':8, 'rumah':9, 'kakek':10, 'sejak':11, 'tiga':12, 'minggu':13, 'lalu':14}

3.5.4. *Sequence*

Pada langkah ini menggunakan fitur keras *text_to_sequence*s. Dimana, fitur tersebut bertujuan untuk mengubah setiap teks menjadi *sequence*. Setiap *sequence* mempresentasikan teks dalam bentuk *array* yang terdiri dari kumpulan token dari teks yang telah melalui tahapan *tokenizing* pada langkah sebelumnya. Lebih jelasnya dapat dilihat pada tabel 3.10. berikut.

Tabel 3.10. Contoh Penerapan *Sequence*s

Teks	*Indexing*
-ibu membuatkan sarapan untuk ayah -fita pergi ke rumah kakek sejak tiga minggu lalu	[1, 2, 3, 4, 5] [6, 7, 8, 9, 10, 11, 12, 13, 14]

3.5.5. *Padding*

Selanjutnya dilakukan proses untuk menyamakan panjang *sequence* yang umumnya disebut proses *padding*. Hal ini bertujuan karena setiap teks memiliki panjang yang berbeda sehingga bisa di proses pada model jaringan syaraf tiruan. Konsep *padding* yaitu memangkas *sequence* yang terlalu panjang dan memberikan nilai 0 pada tiap *sequence* yang terlalu pendek sehingga sesuai dengan panjang *sequence* yang ditetapkan.

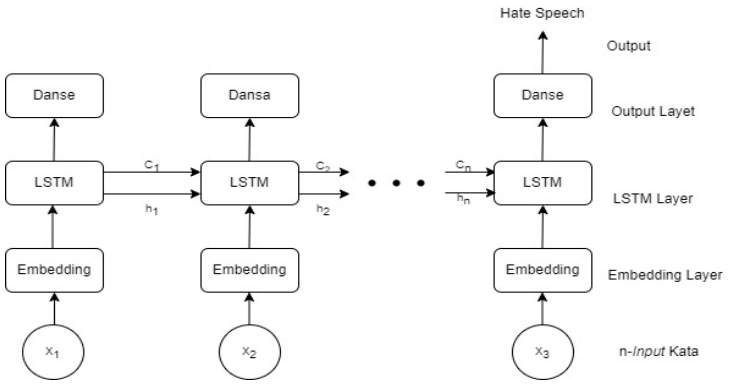
3.5.6. Klasifikasi Teks Menggunakan Model yang Sudah Dilatih

Pada tahapan terakhir, data diolah menggunakan model LSTM. Yang pertama adalah data masuk ke dalam *forget gate* yang fungsinya untuk mengecek dan menyeleksi informasi pada memori. Data tersebut kemudian masuk ke *input gate* yang fungsinya untuk

mengecek dan menentukan informasi baru apa yang ditambahkan ke sel LSTM. Memori sel LSTM kemudian diperbarui. Kemudian masuk menuju *output gate* yang fungsinya untuk mengeluarkan atau menghasilkan data yang sudah dilakukan perhitungan pada sel LSTM tersebut.

3.6. Implementasi Algoritma *Long Short-Term Memory*

Pada penelitian ini pembuatan model LSTM terdiri dari tujuh *layer* diantaranya menerapkan *embedding layer*, LSTM *layer*, dropout *layer*, serta dense *layer*. Lebih jelasnya dapat dilihat pada gambar 3.2. berikut.



Gambar 3.3. Implementasi Algoritma LSTM

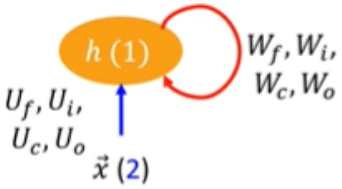
Gambar 3.3. merupakan pembuatan model dari jaringan LSTM. Tahapan ini merupakan proses pendefinisian model keras. Adapun penjelasan dari tiap-tiap *layer* sebagai berikut:

1. *Embedding layer* adalah tahapan mengubah kata ke dalam vektor dan *array* pada bilangan rill.
2. LSTM, tahap ini data akan diproses menggunakan algoritma *Long Short-Term Memory* (LSTM) untuk mendapatkan hasil/*output* berupa model prediksi.
3. *Dropout* digunakan untuk mencegah terjadinya *overfitting*/underfitting. Nilai yang digunakan berkisar 0 hingga 1.

4. *Danse layer* digunakan untuk menambah *layer* berdasarkan jumlah *class* yang ditentukan

3.7. Contoh Perhitungan *Long Short-Term Memory* (LSTM)

Dibawah ini merupakan contoh dari perhitungan dari algoritma LSTM.



Gambar 3.4. Jaringan LSTM

Pada gambar 3.4. dalam jaringan LSTM ini terdapat 2 dimensi *input* untuk setiap nilai x dengan hanya 1 unit LSTM. Setelah itu dalam contoh perhitungan ini, kita telah memiliki data *input.* Dapat di lihat pada tabel 3.10. berikut.

Tabel 3.11. Data *Input*

A1	A2	Target
1	2	0,5
0,5	3	1

Diasumsikan bahwa data yang akan diproses telah tersedia. Model yang siap digunakan terdiri dari sekumpulan matriks bobot U dan bobot W. Setiap matriks U memiliki dimensi 1x2 yang menunjukkan jumlah LSTM *layer* dikalikan dengan dimensi *input*nya, sedangkan matriks W dan b memiliki dimensi masing-masing 1x1 serta terdapat nilai h(t-1) dan C(t-1). Bobot tiap matriks serta nilainya dapat dilihat pada tabel berikut.

Tabel 3.12. Nilai Bobot

Bobot			
$U_f . x_t$	$W_f . h_{t-1} + b_f$	net_ft	Ft
1.600	0.150	1.750	0.852

$U_i . x_t$	$W_i . h_{t-1} + b_i$	net_ft	Ft
2.550	0.650	3.200	0.961
$U_c . x_t$	$W_c . h_{t-1} + b_c$	net_~ct	~ct
0.950	0.200	1.150	0.818
$U_o . x_t$	$W_o . h_{t-1} + b_o$	net_ot	Ot
1.400	0.100	1.500	0.818
h_{t-1}		C_{t-1}	
0		0	

Tabel diatas membuat himpunan bobot U, W, dan b. bobot-bobot ini dihasilkan dari pelatihan model jaringan saraf.

1. *Forget Gate*

Untuk menghitung nilai *forget gate* sebagai berikut

$$\mathrm{f}_t = \sigma\left(\mathrm{U}_f\, x_t + W_f h_{t-1} + \mathrm{b}_f\right)$$
$$= \sigma\ 1.600 + 0.150$$
$$= \sigma\ 1{,}750$$
$$\sigma = \frac{1}{1+e^{-xi}}$$
$$= \frac{1}{1+e^{(-1.750)}}$$
$$= 0.852$$

2. *Input Gate*

Selanjutnya, menghitung nilai *input gate* sebagai berikut.

$$\mathrm{f}_i = \sigma\ (\mathrm{U}_i\, x_t + W_i h_{t-1} + \mathrm{b}_i)$$
$$= \sigma\ 2.550 + 0.650$$
$$= \sigma\ 3.200$$
$$\sigma = \frac{1}{1+e^{-xi}}$$
$$= \frac{1}{1+e^{(-3.200)}}$$
$$= 0.961$$
$$\mathrm{C}_t^{\sim} = \mathrm{Tanh}\ (\mathrm{U}_c\, x_t + W_c h_{t-1} + \mathrm{b}_c)$$
$$= Tanh\ 0.950 + 0.200$$
$$= Tanh\ 1.150$$

$$Tanh = \frac{e^{2xi}-1}{e^{2xi}+1}$$
$$= \frac{e^{2(1.150)}-1}{e^{2(1.150)}+1}$$
$$= 0.818$$

3. *Memory Update*

Setelah mendapatkan nilai *input gate* dan *forget gate,* maka kita dapat menghitung memori *cell new state* sebagai berikut.

$$\mathrm{C}_t = \mathrm{f}_t . C_{t-1} + i_t . C_t^{\sim}$$
$$= (0.852\ x\ 0) + (0.961\ x\ 0.818)$$
$$= 0.786$$

4. *Output Gate*

$$\mathrm{o}_t = \sigma\,(\mathrm{U}_o . x_t + W_o h_{t-1} + \mathrm{b}_o)$$
$$= \sigma\ 1.400 + 0.100$$
$$= \sigma\ 1.500$$
$$\sigma = \frac{1}{1+e^{-xi}}$$
$$= \frac{1}{1+e^{(-1.500)}}$$
$$= 0.818$$
$$h_t = o_t . Tanh\ C_t$$
$$= 0.818\ x\ Tanh\ (0.786)$$
$$= 0.818\ x\ \frac{e^{2(0.786)}-1}{e^{2(0.786)}+1}$$
$$= 0.818\ x\ \frac{3.816}{5.816}$$
$$= 0.818\ x\ 0.656$$
$$= 0.536$$

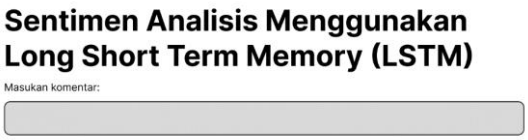
3.8. Evaluasi

Evaluasi yang digunakan pada penelitian ini yaitu *confusion matrix* yang menyimpan informasi yang digunakan sebagai acuan dari performa/kinerja klasifikasi dari algoritma yang digunakan apakah berkerja dengan baik atau tidak. Hal ini dapat dilihat pada pada

nilai variabel *True Positive* (TP) dan variabel *True Negative* (TN) yang menunjukkan prediksi model yang benar. Sedangkan, nilai variabel *False Positive* (FP) dan variabel *False Negative* (FN) menunjukkan jumlah total prediksi salah yang dibuat oleh model. Perhitungan ini dilakukan dengan menghitung nilai *accuracy, precision, recall,* dan F1-*Score* (Fadli, 2019).

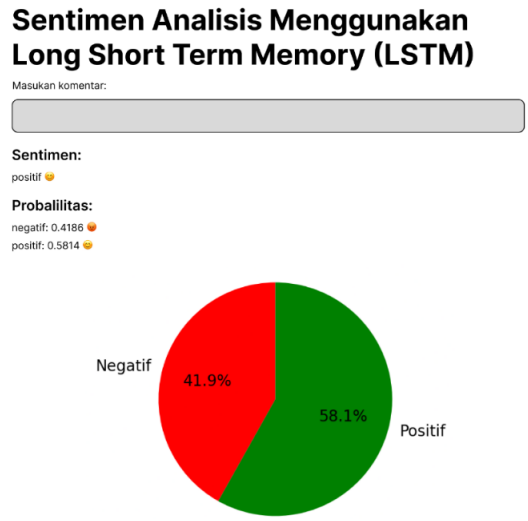
3.9. Desain *Interface*

Pada tahapan ini akan dijelaskan terkait dengan rancangan antarmuka pengguna (*user interface)* dari sistem yang akan dibuat oleh peneliti. Dapat dilihat pada gambar 3.5 berikut.



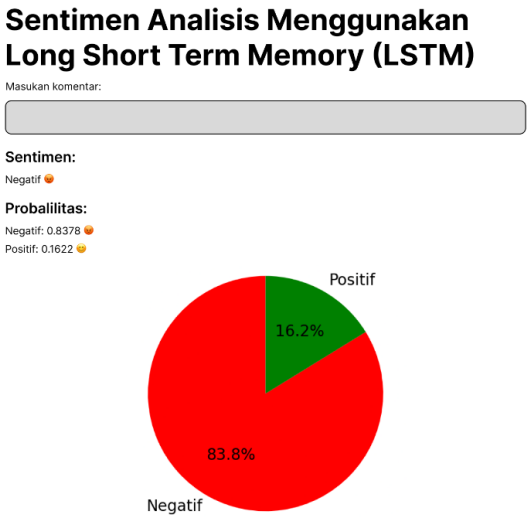
Gambar 3.5. Tampilan Awal Sistem

Gambar diatas merupakan halaman yang pertama kali muncul saat menjalankan sistem.



Gambar 3.6. Tampilan Sentimen Positif

Berdasarkan gambar 3.6 diatas merupakan tampilan dari sistem yang mengandung unsur positif atau *non hate speech.*



Gambar 3.7. Tampilan Sentimen Negatif

Berdasarkan gambar 3.7 diatas merupakan tampilan dari sistem yang mengandung unsur negatif atau *hate speech.*

BAB IV

HASIL DAN PEMBAHASAN

4.1. Analisis Data

Pada tahapan ini membahas terkait dengan pengklasifikasian komentar *hate speech* dan *non hate speech* dengan mencari nilai *recall*, *precision*, dan juga *accuracy* dari model yang dibuat. Komentar tersebut diambil secara manual pada *instagram* dan *facebook* terkait dengan komentar masyarakat terhadap pemerintah Maluku Utara. Selanjutnya data dari komentar tersebut disimpan dengan format CSV. Jumlah *dataset* yang digunakan sebanyak 1.200 komentar yang ada di *instagram* dan *facebook*. Kemudian dilakukan pembagian untuk data latih sebanyak 80% atau 960 komentar dan data test sebanyak 20% atau 240 Komentar. Berikut *dataset* pada penelitian ini dapat dilihat pada gambar 4.1.

	No	Sumber	Nama Pengguna	Komentar	Category	Unnamed: 5
0	1	Instagram	sayakhoko	bicara jaga iko mulu p bobou kong	Hate Speech	NaN
1	2	Instagram	at.abdrahmann	wakil rakyat bisa bapikir bagitu ii, padahal n...	Hate Speech	NaN
2	3	Instagram	dilahakim512	keadaan skrg ketua DPRD me butuh doi, kong bic...	Hate Speech	NaN
3	4	Instagram	lucky_achy	om,, ngoni skola tinggi2 sampe jadi ketua dprd...	Hate Speech	NaN
4	5	Instagram	aby_ammarie_36	mantap… itu baru wakil rakyat wakanda.. Member...	Hate Speech	NaN
...	...	...	...	...	...	...
1195	1196	Instagram	12bojczuk_	Haduhhhh anda di rugikan berapa ? Sampai doak...	Non Hate Speech	NaN
1196	1197	Instagram	12bojczuk_	Kalau diamkan tu bukan tidak di tangkap . Diam...	Non Hate Speech	NaN
1197	1198	Instagram	zawin_nonai	Azab yg maha kuasa	Non Hate Speech	NaN
1198	1199	Instagram	ad_1_nk	Anaknya blg, pak Gub dijebak dan mau mengajuka...	Non Hate Speech	NaN
1199	1200	Instagram	ulendoankaja123	Pak gani Kasuba ini trg p kepala pondok dulu d...	Non Hate Speech	NaN

1200 rows × 6 columns

Gambar 4.1. *Dataset*

Pada gambar 4.1. diatas terdapat 1.200 baris yaitu jumlah komentar dan 6 kolom yaitu no, sumber, nama pengguna, komentar, *category, unnamed*. Pada kolom komentar adalah hasil dari pengumpulan komentar masyarakat terhadap pemerintah Maluku Utara pada postingan di *instagram* dan *facebook* yang peneliti lakukan secara manual. Kemudian pada kolom *category* adalah label *category* yang pelabelannya dilakukan oleh

ahli bahasa Indonesia. Data yang digunakan sebanyak 1.200 data komentar dengan rincian 600 data *hate speech* dan 600 data *non hate speech.* Untuk lebih jelasnya dapat dilihat pada gambar 4.2. berikut.

Gambar 4.2. Rincian *Dataset*

4.2. Data *Preprocessing*

Pada tahapan ini, data *preprocessing* bertujuan untuk membersihkan dan mengolah data teks sehingga menjadi data yang lebih baik dan terstruktur sebelum mamasuki tahapan selanjutnya. *Dataset* yang digunakan masih banyak mengandung *noise* sehingga data perlu dilakukan *preprocessing* data sebagai berikut.

4.2.1. Case Folding

Pada tahapan ini, *dataset* seperti pada gambar 4.1 akan dilakukan proses *case folding* dalam pemrosesan teks adalah membantu mengatasi masalah variasi huruf besar dan kecil dalam kata-kata yang sama. Dapat dilihat pada gambar berikut 4.3. berikut.

```python
def clean_text(text):
    # Remove non-alphanumeric characters
    cleaned_text = re.sub(r'[^a-zA-Z0-9\s]', '', text)
    # Convert text to lowercase
    cleaned_text = cleaned_text.lower()
    # Remove extra whitespaces
    cleaned_text = re.sub(r'\s+', ' ', cleaned_text).strip()
    return cleaned_text

# Assuming df is your DataFrame with columns "Komentar" and "Category"
# Replace "df" with the name of your DataFrame if it's different
df["Komentar"] = df["Komentar"].apply(clean_text)
```

Gambar 4.3. *Script Case Folding*

Pada *script* diatas menggunakan fungsi *lower* untuk mengubah huruf kapital dalam teks menjadi huruf kecil dan juga dilakukan pembersihan lainnya menggunakan fungsi *regular expression (re)* atau lebih dikenal dengan *RegEx*. Pembersihan tersebut akan mengubah semua huruf kapital menjadi huruf kecil, menghapus angka dan spasi yang

berlebihan pada teks sehingga menghasilkan teks yang lebih mudah untuk diproses. Lebih jelasnya dapat dilihat pada tabel 4.1 berikut.

Tabel 4.1. *Case Folding*

No	*Case Folding*
1	bicara jaga iko mulu p bobou kong
2	wakil rakyat bisa bapikir bagitu ii padahal nakes tuh dorang cm tuntuk dorang p hak ih sp pilih pa paitua ini kh
3	keadaan skrg ketua dprd me butuh doi kong bicara bagitu hafal saja
4	om ngoni skola tinggi2 sampe jadi ketua dprd kg bukang tamba pinta tpi silahkan di isi sandiri kt me bingung
5	mantap itu baru wakil rakyat wakanda memberikan maslah tanpa menyelesaikan masalah
…	…..
1196	haduhhhh anda di rugikan berapa sampai doakan seperti itu
1197	kalau diamkan tu bukan tidak di tangkap diamkan itu orang tra akan proses itu kan sdh di proses dan sdh menerima konsukwensi mo apa lg
1198	azab yg maha kuasa
1199	anaknya blg pak gub dijebak dan mau mengajukan tuntutan balik
1200	pak gani kasuba ini trg p kepala pondok dulu di madrasah aliyah alkhairat kalumpang ternate tahu 2001-2006

4.2.2. Tokenizing

Tokenizing adalah metode yang digunakan untuk memisahkan setiap kata pada teks sehingga tidak membentuk kalimat yang utuh. Proses pemecahan kalimat dilakukan berdasarkan spasi kata antar kalimat sehingga menjadi kumpulan kata-kata dalam sebuah daftar yang disebut *token.* Dapat dilihat pada gambar berikut 4.4. berikut.

```
import pandas as pd
import nltk
from nltk.tokenize import word_tokenize

# Make sure to download the NLTK data for tokenization
nltk.download('punkt')

# Define the text preprocessing function (without stemming)
def tokenize_text(text):
    # Tokenization
    tokens = word_tokenize(text)
    return tokens


# Apply the function and create a new column for tokenized words
df['Tokenized_Komentar'] = df['Komentar'].apply(tokenize_text)
```

Gambar 4.4. Script *Tokenizing*

Berdasarkan hasil *script* di atas, digunakan *phyton* untuk melakukan praproses teks dengan membuat kumpulan token pada data teks dengan mengimpor *library* yaitu *pandas* untuk manipulasi dan mengolah data dalam bentuk data *frame*. Kemudian digunakan nltk.*tokenize import word_tokenize* untuk memecah teks menjadi kata-kata *(token).* Dan untuk menampilkan kolom dari data *frame* menggunakan (df['*Tokenized*_Komentar']). Dapat dilihat pada tabel 4.2. berikut.

Tabel 4.2. *Tokenizing*

No	*Tokenizing*
1	['bicara', 'jaga', 'iko', 'mulu', 'p', 'bobou', 'kong']
2	['wakil', 'rakyat', 'bisa', 'bapikir', 'bagitu', 'ii', 'padahal', 'nakes', 'tuh', 'dorang', 'cm', 'tuntuk', 'dorang', 'p', 'hak', 'ih', 'sp', 'pilih', 'pa', 'paitua', 'ini', 'kh']
3	['keadaan', 'skrg', 'ketua', 'dprd', 'me', 'butuh', 'doi', 'kong', 'bicara', 'bagitu', 'hafal', 'saja']
4	['om', 'ngoni', 'skola', 'tinggi2', 'sampe', 'jadi', 'ketua', 'dprd', 'kg', 'bukang', 'tamba', 'pinta', 'tpi', 'silahkan', 'di', 'isi', 'sandiri', 'kt', 'me', 'bingung']
5	['mantap', 'itu', 'baru', 'wakil', 'rakyat', 'wakanda', 'memberikan', 'maslah', 'tanpa', 'menyelesaikan', 'masalah']
…	…..
1196	['haduhhhh', 'anda', 'di', 'rugikan', 'berapa', 'sampai', 'doakan', 'seperti', 'itu']
1197	['kalau', 'diamkan', 'tu', 'bukan', 'tidak', 'di', 'tangkap', 'diamkan', 'itu', 'orang', 'tra', 'akan', 'proses', 'itu', 'kan', 'sdh', 'di', 'proses', 'dan', 'sdh', 'menerima', 'konsukwensi', 'mo', 'apa', 'lg']
1198	['azab', 'yg', 'maha', 'kuasa']
1199	['anaknya', 'blg', 'pak', 'gub', 'dijebak', 'dan', 'mau', 'mengajukan', 'tuntutan', 'balik']
1200	['pak', 'gani', 'kasuba', 'ini', 'trg', 'p', 'kepala', 'pondok', 'dulu', 'di', 'madrasah', 'aliyah', 'alkhairat', 'kalumpang', 'ternate', 'tahu', '2001-2006']

4.2.3. Stopword Removal

Stopwords Removal adalah proses menghilangkan kata-kata yang sering muncul dalam teks tetapi tidak memberikan makna yang signifikan atau informasi kontekstual

penting. Tahapan ini dilakukan penghapusan kata-kata yang tidak diperlukan pada suatu kalimat yang ada pada *dataset*. Lebih jelasnya dapat dilihat pada gambar 4.5. berikut

```
from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory
from Sastrawi.Stemmer.StemmerFactory import StemmerFactory
import re

def preprocess_text(text):

    # Sastrawi stopword removal
    factory = StopWordRemoverFactory()
    stopword_remover = factory.create_stop_word_remover()
    text = stopword_remover.remove(text)

    return text

df["Komentar"] = df["Komentar"].apply(preprocess_text)
```

Gambar 4.5. *Script Stopword Removal*

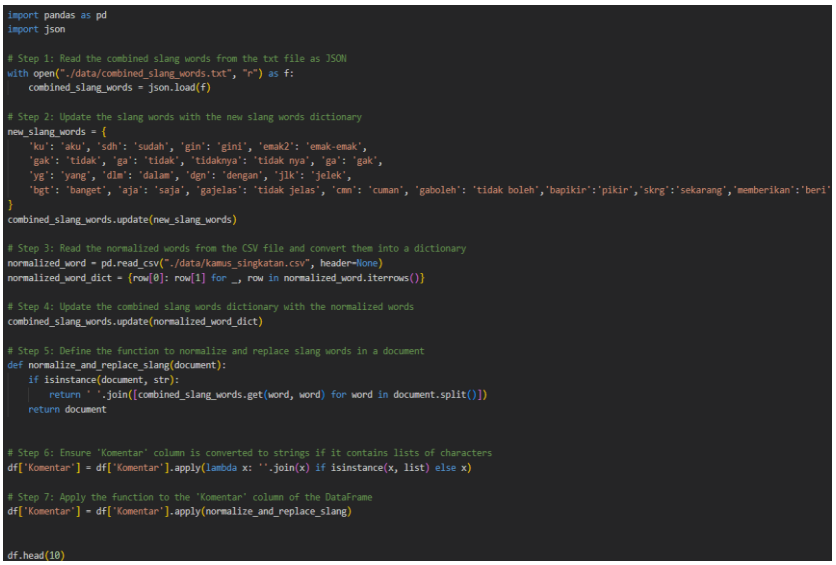
Berdasarkan *script* diatas bertujuan untuk membersihkan teks komentar dari data frame dengan menghapus *stopword* mengguakan Sastrawi di *python.* Kemudian digunakan fungsi *StopwordRemovalFactory* untuk menghilangkan kata-kata yang tidak memiliki arti penting dalam teks. Lebih jelasnya dapat dilihat pada tabel 4.3. berikut.

Tabel 4.3. *Stopword Removal*

No	*Stopword Removal*
1	bicara jaga iko mulu p bobou kong
2	wakil rakyat bapikir bagitu ii padahal nakes tuh dorang cm tuntuk dorang p hak ih sp pilih pa paitua kh
3	keadaan skrg ketua dprd me butuh doi kong bicara bagitu hafal
4	om ngoni skola tinggi sampe jadi ketua dprd kg bukang tamba pinta tpi silahkan isi sandiri kt me bingung
5	mantap baru wakil rakyat wakanda memberikan maslah menyelesaikan masalah
…	…..
1196	haduhhhh di rugikan berapa doakan itu
1197	kalau diamkan tu bukan tangkap diamkan orang tra proses kan sdh di proses sdh menerima konsukwensi mo apa lg
1198	azab yg maha kuasa
1199	anaknya blg pak gub dijebak mau mengajukan tuntutan balik
1200	pak gani kasuba trg p kepala pondok dulu madrasah aliyah alkhairat kalumpang ternate tahu

4.2.4. Stemming

Tahapan selanjutnya adalah melakukan *stemming*, *stemming* adalah proses mengubah kata-kata yang terinfleksi atau memiliki variasi bentuk kata (seperti kata-kata berimbuhan) ke bentuk dasar atau "akar" kata. Proses *stemming* membutuhkan waktu yang lama daripada proses yang lainnya. Proses *stemming* ini membutuhkan waktu hingga 18 menit. Lebih jelasnya dapat dilihat pada gambar 4.6. berikut.



```
import pandas as pd
import json

# Step 1: Read the combined slang words from the txt file as JSON
with open("./data/combined_slang_words.txt", "r") as f:
    combined_slang_words = json.load(f)

# Step 2: Update the slang words with the new slang words dictionary
new_slang_words = {
    'ku': 'aku', 'sdh': 'sudah', 'gin': 'gini', 'emak2': 'emak-emak',
    'gak': 'tidak', 'ga': 'tidak', 'tidaknya': 'tidak nya', 'ga': 'gak',
    'yg': 'yang', 'dlm': 'dalam', 'dgn': 'dengan', 'jlk': 'jelek',
    'bgt': 'banget', 'aja': 'saja', 'gajelas': 'tidak jelas', 'cmn': 'cuman', 'gaboleh': 'tidak boleh','bapikir':'pikir','skrg':'sekarang','memberikan':'beri'
}
combined_slang_words.update(new_slang_words)

# Step 3: Read the normalized words from the CSV file and convert them into a dictionary
normalized_word = pd.read_csv("./data/kamus_singkatan.csv", header=None)
normalized_word_dict = {row[0]: row[1] for _, row in normalized_word.iterrows()}

# Step 4: Update the combined slang words dictionary with the normalized words
combined_slang_words.update(normalized_word_dict)

# Step 5: Define the function to normalize and replace slang words in a document
def normalize_and_replace_slang(document):
    if isinstance(document, str):
        return ' '.join([combined_slang_words.get(word, word) for word in document.split()])
    return document


# Step 6: Ensure 'Komentar' column is converted to strings if it contains lists of characters
df['Komentar'] = df['Komentar'].apply(lambda x: ''.join(x) if isinstance(x, list) else x)

# Step 7: Apply the function to the 'Komentar' column of the DataFrame
df['Komentar'] = df['Komentar'].apply(normalize_and_replace_slang)


df.head(10)
```

Gambar 4.6. *Script Stemming*

Berdasarkan *script* diatas, bertujuan untuk membersihkan dan menormalkan teks dalam kolom “komentar” dari data frame dengan mengganti kata-kata *slang* menggunakan kombinasi dari kamus kata-kata slang dan kata-kata yang dinormalisasi. Mengimpor *pandas* dan membaca *file combinated_slang_worts.txt* yang berisi kamus kata-kata slang format json dan menyimpan dalam variabel. Kemudian untuk menambahkan kata-kata slang menggunakan metode *‘update’* dengan membaca *file* kamus_singkatan.csv yang berisi kata *slang* dan kata normal yang dikonversi menjadi *normalized_word_dict* kedalam *combined_slang_words*. Lebih jelasnya dapat dilihat pada tabel 4.4 berikut.

Tabel 4.4. *Stemming*

No	*Stemming*
1	bicara jaga iko melulu p bobou kong
2	wakil rakyat pikir bagitu ii padahal nakes tuh dorang cm tuntuk dorang p hak ih sp pilih pa paitua kh
3	keadaan sekarang ketua dprd me butuh doi kong bicara bagitu hafal
4	om ngoni skola tinggi sampai jadi ketua dprd kg bukang tamba pinta tapi silahkan isi sandiri kita me bingung
5	mantap baru wakil rakyat wakanda beri maslah menyelesaikan masalah
…	…..
1196	haduhhhh di rugikan berapa doakan itu
1197	kalau diamkan tu bukan tangkap diamkan orang tra proses sudah proses sudah menerima konsukwensi mau apa lagi
1198	azab yang maha kuasa
1199	anaknya bilang pak gub dijebak mau mengajukan tuntutan balik
1200	pak gani kasuba trg p kepala pondok dulu madrasah aliyah alkhairat kalumpang ternate tahu

4.3. Pembagian Data *Training* dan Data *Testing*

Pada tahapan ini *dataset* dibagi menjadi 2 yaitu data *train* dan data test yang menggunakan fungsi *train_test_split* dari *library sklearn*.model untuk membagi data. Dari 1.200 data dibagi menjadi 80% atau 960 komentar data *train* dan 20% atau 240 untuk data test. Setiap data yang diperoleh dipisahkan secara acak menggunakan nilai *random_state* untuk mengontrol *randomization* agar hasil yang konsisten. Lebih jelasnya dapat dilihat pada gambar 4.7. berikut.

```
(960, 89) (960, 2)
(240, 89) (240, 2)
```

Gambar 4.7. Pembagian *Dataset*

4.4. Implementasi Model *Long Short-Term Memory*

Setelah data komentar yang diperoleh sudah melalui tahapan *preprocessing* data dan juga dilakukan pembagian data *train* dan data test, maka selanjutnya masuk pada

tahapan *indexing, sequence*, dan *padding* sebelum diproses dalam model LSTM. Model *deep learning* tidak dapat mengolah data dalam bentuk format teks/*string* maka harus dilakukan proses *indexing* untuk mengonversi setiap token menjadi bilangan numerik integer.

4.4.1. *Indexing*

Pada tahap berikutnya data komentar tersebut dilakukan proses i*ndexing* dan mengonversi setiap teks menjadi bilangan numerik. Selanjutnya diperlukan parameter *'num_words'* yang memungkinkan kita untuk mengatur jumlah kosakata yang akan digunakan sesuai dengan *max_features*. Pada penelitian ini dengan jumlah komentar sebanyak 1.200 dengan parameter *num_word* berjumlah 4591 kata. Dapat dilihat pada tabel 4.5 berikut.

Tabel 4.5. *Indexing*

Indexing
{'di':1, 'yg':2, 'itu':3, 'ini';4, 'dan':5, 'sampah':6, 'ada':7, 'ada':8, 'p':9, 'pak':10, 'masyarakat':11, 'dari':12, 'pejabat':13, 'tra':14, 'so':15, 'rakyat':16, 'jadi':17, 'orang':18, 'tu':19, 'bayar':20, 'korupsi':21, 'lagi':22, 'tidak':23, 'deng':24, 'saja':25, 'juga':26, 'buang':27, 'bisa':28, 'dong':29, 'jalan':30, 'me':31, 'parkir':32, 'kita':33, 'doi':34, 'dia':35, 'ke':36, 'pemerintah':37, 'pe':38, 'tapi':39, 'apa':40, 'bukan':41, 'ya':42, 'kalo':43, 'nya'44, 'ngoni':45, 'sampe':46, 'tong':47, 'sama':48, 'ternate':49, 'dg':50, 'kase':51, 'untuk':52, 'kong':53, 'ni':54, 'tau':55, 'sudah':56, 'klo':57, 'salah':58, 'masih':59, 'mau':60, 'bupati':61, 'saya':62, 'banyak':63, 'baru':64, 'tara':65, 'uang':66, 'semua':67, 'harus':68, 'd':69, 'kalau':70, 'semoga':71, 'jang':72, 'kpk':73, 'allah':74, 'buat':75, 'sehat':76, 'smpe':77, 'cuma':78, 'beliau':79, 'barangka':80, 'akan':81, 'kota':82, 'mo':83, 'karcis':84, 'dalam':85, 'diri':86, 'coba':87, 'bapak':88, 'kan':89, 'dapa':90, 'gub':91, 'pasar':92, 'biar':93, 'kadang':94, 'dengan':95, 'maluku':96, 'utara':97, 'dorang':98, 'org':99, 'punya':100 .....'kalumpang':4591}

4.4.2. *Sequences* dan *Padding*

Setelah setiap kata atau token diubah menjadi bilangan numerik, maka setiap baris data akan direpresentasikan kedalam *sequence* yang terdiri dari kumpulan teks dari setiap

dokumen. Untuk lebih jelasnya dapat dilihat pada gambar 4.8. berikut.

```
max_fatures = 2000
tokenizer = Tokenizer(num_words=max_fatures, split=' ')
tokenizer.fit_on_texts(df['Komentar'].values)
X = tokenizer.texts_to_sequences(df['Komentar'].values)
X = pad_sequences(X)
print("index pad sequence")
X[:2]
```

Gambar 4.8. *Script Sequences* dan *Padding*

Berdasarkan *script* diatas digunakan fungsi *max_fatures* untuk menetapkan batas maksimum untuk jumlah kata dalam tokenisasi sebanyak 2000 kata yang paling sering muncul dan kata-kata lain akan diabaikan. Kemudian membuat objek *(num_word=max_fatures)* untuk mempertimbangkan kata yang paling sering muncul dan *split* untuk mengatur pemisahan kata berdasarkan spasi. Dan untuk mengonversi setiap komentar teks menjadi urutan angka berdasarkan kamus menggunakan fungsi *texts_to_sequences*. Selanjutnya *pad_sequences* untuk menambahkan *padding* pada urutan angka agar memiliki panjang yang sama karena diperlukan pada model *learning*. Untuk lebih jelasnya dapat dilihat pada tabel 4.6 berikut.

Tabel 4.6. *Sequences*

No	*Sequences*
1	[101, 160, 444, 445, 5, 1077, 45]
2	[170, 17, 446, 274, 612, 126, 275, 118, 79, 384, 1629, 79, 5, 250, 523, 613, 119, 276, 161, 524]
3	[786, 149, 144, 177, 28, 251, 29, 45, 101, 274, 1078]
4	[208, 36, 1630, 525, 34, 4, 144, 177, 91, 787, 788, 1631, 46, 1079, 447, 448, 193, 28, 789]
5	[312, 26, 170, 17, 1080, 449, 1081, 1632, 102],
6	[144, 177, 5, 171, 1082, 450]
7	[101, 33, 42, 15, 12, 1633, 16, 144, 177, 1634, 11, 790]
8	[1078, 161, 5, 137, 18, 3, 1083, 332, 43, 444, 1084, 1085]
9	[194, 1, 46, 119, 91, 4, 451, 177, 16, 120, 39, 101]
10	[614, 1, 1635, 144, 177, 1, 21, 791, 1636, 275, 1, 127, 791]

Kemudian dilakukan proses *padding* untuk menyamakan panjang dari setiap *sequences*. Setiap kalimat memiliki jumlah kata yang berbeda-beda sehingga untuk mengisi kekosongan yang ada dalam sistem yang panjang, maka akan disamakan dengan menggunakan *pad_sequences* pada keras untuk mengisi angka 0 secara otomatis pada setiap kata. Pada penelitian ini digunakan *max_features*=60

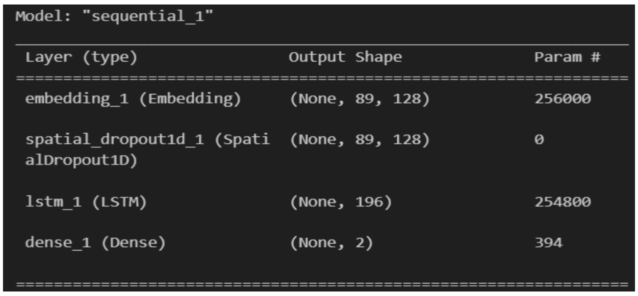
Tabel. 4.7. *Padding*

No	*Padding*
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, 101, 160, 444, 445, 5, 1077, 45]
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, 170, 17, 446, 274, 612, 126, 275, 118, 79, 384, 1629, 79, 5, 250, 523, 613, 119, 276, 161, 524]
3	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 786, 149, 144, 177, 28, 251, 29, 45, 101, 274, 1078]
4	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 208, 36, 1630, 525, 34, 4, 144, 177, 91, 787, 788, 1631, 46, 1079, 447, 448, 193, 28, 789]
5	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 312, 26, 170, 17, 1080, 449, 1081, 1632, 102]
6	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 144, 177, 5, 171, 1082, 450],
7	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 101, 33, 42, 15, 12, 1633, 16, 144, 177, 1634, 11, 790]
8	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1078, 161, 5, 137, 18, 3, 1083,

	332, 43, 444, 1084, 1085]
9	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 194, 1, 46, 119, 91, 4, 451, 177, 16, 120, 39, 101]
10	[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 614, 1, 1635, 144, 177, 1, 21, 791, 1636, 275, 1, 127, 791]

4.5. Arsitektur Model

Arsitektur model dirancang untuk memproses *input* teks hingga akurasi optimal. Arsitektur model memiliki lapisan-lapisan seperti *embedding layer* untuk mengkonversi *input* teks ke dalam vektor *embedding,* Setelah menerima *output* dari *embedding layer* masuk pada *SpatialDropout1D Layer* yang bertujuan menghapus fitur untuk mencegah *overfitting.* Pada LSTM *Layer* memiliki jumlah *neuron* dan menerapkan *dropout* untuk *input.* Setelah semuanya dibuat, model terlebih dahulu dibangun menggunakan optimasi Adam dan biner *cross entropy* untuk mengetahui nilai *loss* dari model yang terbentuk. Untuk lebih jelasnya dapat dilihat pada gambar 4.9. berikut.

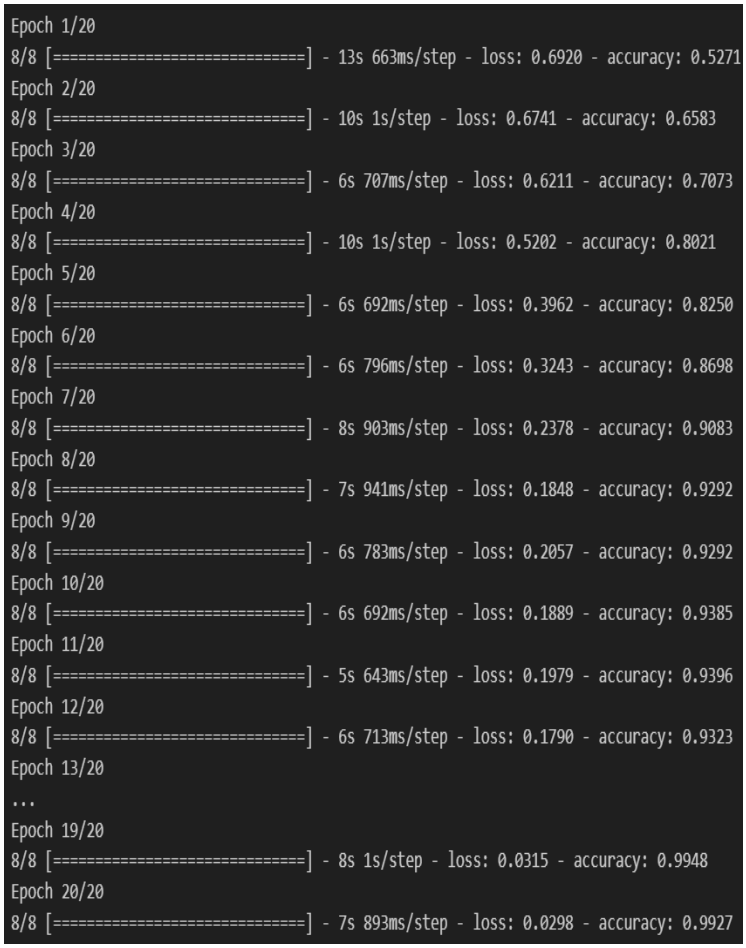


```
Model: "sequential_1"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 embedding_1 (Embedding)     (None, 89, 128)           256000

 spatial_dropout1d_1 (Spati  (None, 89, 128)           0
 alDropout1D)

 lstm_1 (LSTM)               (None, 196)               254800

 dense_1 (Dense)             (None, 2)                 394

=================================================================
```

Gambar 4.9. Arsitektur Model

4.6. Pengujian Data

Pada pengujian ini dilakukan dengan menggunakan 1200 data yang telah melewati proses *preprocessing* data dengan *epoch* yang sudah ditentukan sebanyak 20 untuk membuat model dan mengetahui akurasi tertinggi. Dapat dilihat pada gambar 4.10 berikut.



Gambar 4.10. Pengujian Data

Pada gambar diatas, diperoleh nilai *loss* pada data latih sebesar 0.6920. Dapat dilihat bahwa data latih terjadi penurunan nilai loss yang cukup banyak sampai epoch ke 20.

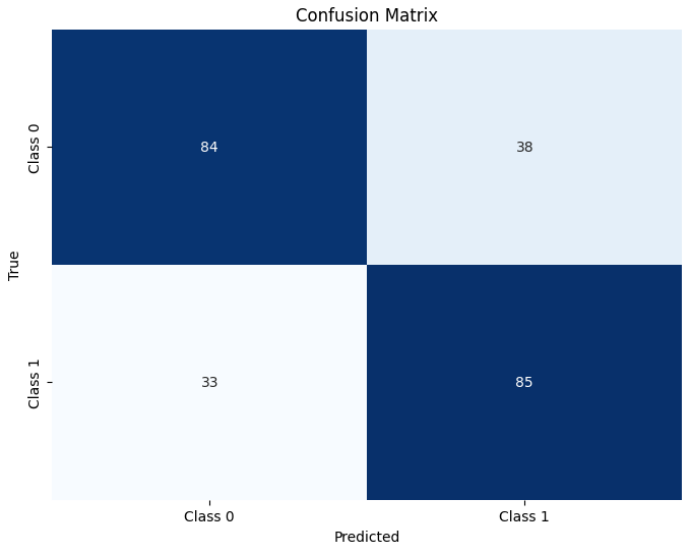
4.7. Evaluasi Data

Pada tahapan ini akan dilakukan pengujian model dengan menggunakan data komentar. Data yang di uji sebanyak 1.200 data dengan pembagian 600 komentar *hate speech* dan 600 komentar *non hate speech.* Model ini mencapai kinerja evaluasi dengan rata-rata nilai *precision, recall*, dan f1-*score* sebesar 70%. Lebih jelasnya dapat dilihat pada gambar 4.11. berikut.

```python
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.metrics import confusion_matrix, classification_report

def plot_confusion_matrix(conf_matrix, labels):
    plt.figure(figsize=(8, 6))
    sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues', cbar=False,
                xticklabels=labels,
                yticklabels=labels)
    plt.title('Confusion Matrix')
    plt.xlabel('Predicted')
    plt.ylabel('True')
    plt.show()

# Assuming Y_test contains the true labels and Y_pred contains the predicted probabilities

# Convert Y_test and Y_pred to lists
Y_test_list = Y_test.tolist()
Y_pred_list = Y_pred.tolist()

# Convert the true labels and predicted probabilities to class labels
true_labels = [np.argmax(x) for x in Y_test_list]
predicted_labels = [np.argmax(x) for x in Y_pred_list]

# Create a DataFrame to compare true labels with predicted labels
df_test = pd.DataFrame({'true': true_labels, 'pred': predicted_labels})

# Generate confusion matrix
conf_matrix = confusion_matrix(df_test.true, df_test.pred)
print("Confusion Matrix:\n", conf_matrix)

# Generate classification report
class_report = classification_report(df_test.true, df_test.pred)
print("Classification Report:\n", class_report)

# Assuming labels are ['Class 0', 'Class 1', ...]
labels = ['Class {}'.format(i) for i in range(len(conf_matrix))]

# Plot confusion matrix
plot_confusion_matrix(conf_matrix, labels)
```

Gambar 4.11. *Script* Evaluasi Data

Berdasarkan *script* diatas bertujuan untuk membuat dan memvisualisasikan *confusion matrix* dari hasil prediksi model klasifikasi. Hasil pengujian *confusion matrix* dapat dilihat pada gambar 4.12. berikut.



Gambar 4.12. *Confusion Matrix*

Berdasarkan hasil evaluasi *confusion matrix* pada gambar diatas dapat diketahui bahwa terdapat 240 data test dimana 84 data test dianggap *non hate speech* dan kenyataannya memang *non hate speech*, banyaknya data yang dideteksi salah sebagai *non hate speech* adalah 38, banyaknya data yang salah dideteksi sebagai *hate speech* namun sebenarnya *non hate speech* adalah 33, dan banyaknya data yang dideteksi benar sebagai ujaran kebencian adalah 84.

Berdasarkan penelitian sebelumnya dari (Talita, 2019) model ini dengan nilai *accuracy* sebesar 70% sudah cukup baik dalam mendeteksi komentar *hate speech.* Namun masih terdapat beberapa kesalahan pada pelabelan yang dilakukan secara manual dengan hasil perhitungan pada model dalam klasifikasi LSTM yang mencakup bahasa informal dari media sosial diantaranya terdapat penulisan kata-kata yang seringkali disingkat dan cenderung mengganti huruf tertentu menjadi angka sehingga tidak dapat terdeksi sebagai *hate speeh* sesuai dengan model yang dibangun. Semakin banyak komentar yang dilabeli secara manual benar maka semakin tinggi *accuracy, precision, recall*, dan f1-*score* yang diperoleh. Untuk hasil dari nilai *precision, recall*, dan f1-*score* dapat dilihat pada Tabel 4.8 berikut.

Tabel. 4.8. Hasil Evaluasi Data

Category	*Precision*	*Recall*	*F1-Score*	*Support*
Hate Speech	0,72	0,69	0,70	122
Non Hate Speech	0,69	0,72	0,71	118

Berdasarkan gambar diatas didapatkan *accuracy* dari perhitungan model klasifikasi yang dibangun adalah 70%, nilai perhitungan tingkat keakuratan dari data yang diminta dengan hasil dari sistem atau *precision*-nya adalah 0,69 nilai perhitungan dari keberhasilan

sistem dalam menemukan kembali sebuah informasi atau *recall* 0,72 dan *f1-score* dari perhitungan dari perbandingan antara nilai *precision* dan *recall* adalah 0,71.

4.8. Analisis Perbandingan

Pada penelitian sebelumnya oleh Fadli (2019) dengan judul "Identifikasi *Cyberbullying* pada media sosial twitter menggunakan metode LSTM dan BiLSTM", sedangkan pada penelitian ini dengan judul "Klasifikasi Hate Speech Menggunakan *Long Short-Term Memory* (LSTM)" menggunakan Algoritma yang sama namun memiliki sumber data dan tujuan yang berbeda.

Penelitian selanjutnya yang dilakukan Isnain (2022) menganalisis pebandingan Algoritma LSTM dan *Naïve Bayes* untuk analisis sentimen dengan dataset 1.590 data dengan hasil LSTM menghasilkan nilai akurasi 83,33% yang lebih baik dibandingkan *Naïve Bayes* dengan nilai akurasi 82%. Sedangkan penelitian ini menganalisis *dataset* sebanyak 1.200 komentar yang ada di *facebook* dan *instagram* dengan nilai akurasi sebesar 70%, yang dimana data yang digunakan lebih sedikit maka nilai presentasi akurasi yang dihasilkan juga lebih sedikit. Hal ini dikarenakan walaupun menggunakan metode yang sama, tetapi objek dan tujuan yang berbeda serta jumlah data yang digunakan juga berbeda.

Kemudian Penelitian berikutnya yang dilakukan oleh Ihsan (2021*)* pada penelitian tersebut menggunakan metode *Algoritme Decision Tree* untuk mengetahui ujaran kebencian/*hate speech* pada Twiiter dengan rata-rata akurasi 69,77%. Sedangkan pada penelitian ini menggunakan metode LSTM pada media sosial *facebook* dan *instagram* memiliki nilai akusari sebesar 70% yang berarti bahwa metode LSTM menghasilkan nilai akurasi lebih tinggi dari *Algoritme Decision Tree*.

Berikutnya penelitian yang dilakukan Nurvania (2021) dengan judul "Analisis Sentimen Pada Ulasan di *TripAdvisor* Menggunakan Metode *Long Short-Term Memory* (LSTM)." Hasil pengujian pada penelitian ini menggunakan *confusion matrix* dan mendapatkan nilai akurasi sebanyak 71,67%, *precision* sebesar 72,97%, *recall* sebesar 95,3% dan F1-*Score* sebesar 82,7% pada media sosial *facebook* dan *instagram*. Namun pada penelitian ini "Klasifikasi *Hate Speech* Menggunakan *Long Short-Term Memory* (LSTM)", LSTM memperoleh akurasi sedikit lebih rendah yaitu sebesar 70%, presisi sebesar 69%, *recall* sebesar 72% dan F1-*Score* sebesar 71% pada media sosial *facebook* dan *instagram.*

Pada penelitian yang dilakukan oleh Bert (2023) yaitu deteksi *Hate Speech* pada *Twiiter* menggunakan Algoritma *Bert* dengan hasil akurasi sebesar 78,69%, presisi sebesar 78,90%, *recall* sebesar 78,69, dan F1-*Score* sebesar 78,77%. Sedangkan penelitian ini mengklasifikasikan *Hate Speech* menggunakan Algoritma yang berbeda yaitu *Long Short-Term Memory* (LSTM) dengan jumlah akurasi lebih kecil yaitu sebesar 70%, presisi sebesar 69%, *recall* sebesar 72% dan F1-*Score* sebesar 71% pada media sosial *facebook* dan *instagram.*

4.9. Tampilan *Interface*

Pada sistem ini terdapat beberapa tampilan halaman *interface* diantaranya yaitu tampilan awal dan tampilan sentiment.

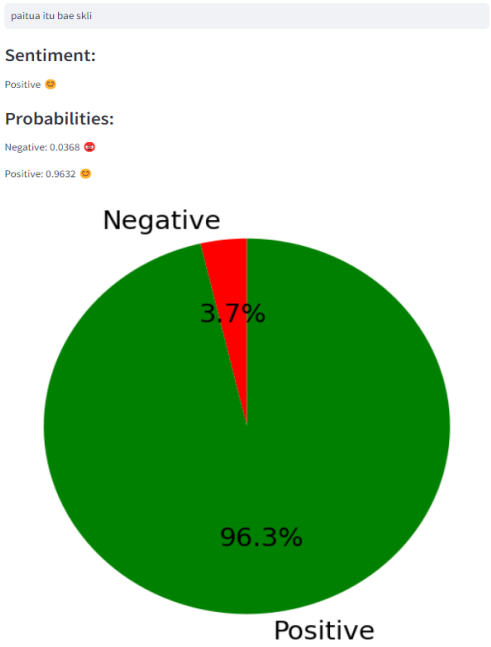
1. Tampilan Awal

   Tampilan awal pada sistem yaitu akan muncul sebuah kotak yang diwajibkan untuk memasukkan komentar dari user pada sistem yang telah dibuat. Dapat dilihat pada gambar berikut.

Gambar 4.13. Tampilan Awal

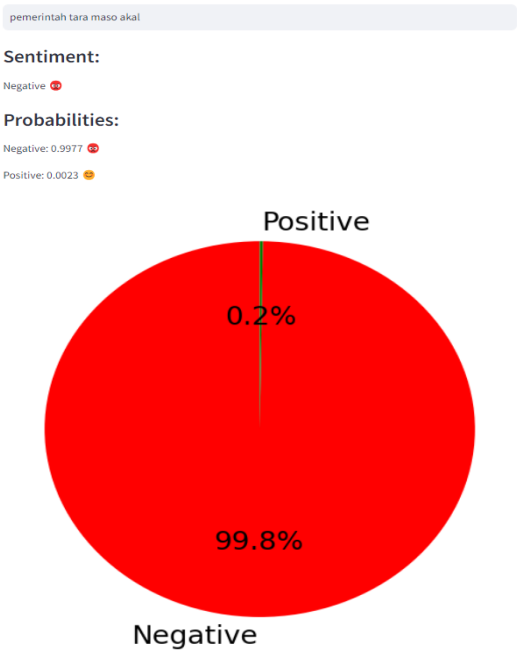
2. Tampilan Sentimen

Pada halaman ini berfungsi untuk menentukan komentar yang telah dimasukkan oleh *user* termasuk pada kategori *hate speech* ataupun *non hate speech.* Dapat dilihat pada gambar berikut.



Gambar 4.14. Tampilan Sentimen Positif

Berdasarkan gambar diatas dapat diketahui bahwa komentar tersebut mengandung unsur positif sebesar 96,3% dan 3,7% unsur negatif. Dengan presentasi tinggi yaitu 96,3% yang berarti bahwa komentar tersebut merupakan komentar *non hate speech*.



Gambar 4.15. Tampilan Sentimen Negatif

Berdasarkan gambar diatas dapat diketahui bahwa komentar tersebut mengandung unsur negatif sebesar 99,8% dan 0,2% unsur negatif. Dengan presentasi tinggi yaitu 99,8% yang berarti bahwa komentar tersebut merupakan komentar *hate speech*.

4.10. Tabel Pengujian Sistem

Berikut merupakan data komentar dari *facebook* dan *instagram* yang dilakukan pengujian oleh sistem sebagai berikut.

Tabel 4.9. Hasil Pengujian Sistem

No	Komentar	Sentimen	
		Positif	Negatif
1	bicara jaga iko mulu p bobou kong	0,0004	0,9996
2	pak kuntu ni so jadi wakil rakyat sada baru bicara tra tau diri tu	0,0003	0,9997
3	iyo nnti gubernur mkn makanan enak deng korupsi uang negara	0,0000	1,0000
4	batagi trus me trada perubahan cuaksssssss	0,0004	0,9996
5	tagih karcis elit jln bagus sulit	0,0002	0,9998

6	definisi provinsi paling bahagia pak gub nya full senyummm	0,9978	0,0022
7	pak kiyai baik cuma salah bergaul	0,9762	0,0238
8	sehat selalu pak gub mari tng tahan jari selagi blm ada putusan pengadilan asas praduga tak bersalah	0,0000	1,0000
9	trust issue sama kyai tokoh agama yg jdi politikus	0,8705	0,1295
10	daripada bahujat beliau lebih bagus hujat isriwil	0,9853	0,0147
11	sehat selalu ustad yang namanya manusia pasti punya salah dan khilaf	0,9027	0,0973
12	masyaallah kasiang di hari tua yang harusnya dihabiskan bersama keluarga	0,9960	0,0040
13	diluar kesalahan yang telah diperbuat beliau ingatlah adab	0,8561	0,1439
14	kecewa liat gubernur model kayak begini semoga gubernur selanjutnya lebih baik bersyukur dengan gajinya dan selalu memperhatikan masyarakat yg miskin	0,9995	0,0005
15	sehat selalu pak semua manusia punya kesalahan	0,9825	0,0175
16	batul bikin sengsara masok bayar parkir doi abis deng bayar karcis	0,0007	0,9993
17	nae kamari kg bikin rusak sj	0,0000	1,0000
18	halibiru sajaaaaa	0,0010	0,9990
19	bupati laef	0,0391	0,9609
20	bupati pe komunikasi politik mungkin waktu kuliah nilai e ka apa eeh	0,0006	0,9994
21	bkp kg jadi bupati k model itu tu	0,0016	0,9984
22	lebe bae tong bunuh p dorang dari pada bkin bibit tra bae lebe bae brenti jd bupaati dari pada tamba bibit yg tra bae	0,3980	0,6020
23	tong p kualitas pemimpin cuma bagini2 saja ceeiii	0,0003	0,9997
24	anggapan kyk bagitu ose stop dari jabatan itu sdh bupati ee	0,2843	0,7157
25	manyasal pilih dia	0,2802	0,7198
26	layani sesuai dengan sop foya saja	0,0003	0,9997
27	jgn mau senang sendiri apalagi pake uang negara	0,0005	0,9995
28	nama nya juga aliong mus panipu	0,0020	0,9980
29	kegagalan otonomi daerah hasilkan pejabat2 tdk becus	0,0000	1,0000
30	pejabat rata2 bacot doank gede	0,0000	1,0000

BAB V

KESIMPULAN DAN SARAN

5.1. Kesimpulan

Berdasarkan hasil penelitian yang telah dilakukan dapat disimpulkan bahwa:

1. Peneliti ini berhasil dilakukan dengan jumlah *dataset* yang digunakan sebanyak 1.200 komentar yang dikumpulkan dari *instagram* dan *facebook* secara manual dengan rincian 600 komentar *hate speech* dan 600 komentar *non hate speech.*
2. Pelabelan data dilakukan oleh ahli bahasa Indonesia, kemudian data dibagi kedalam data latih sebanyak 80% atau 960 komentar dan data test sebanyak 20% atau 240 komentar.
3. Data tersebut kemudian dilakukan tahapan *preprocessing* yang bertujuan untuk membersihkan dan mengolah data teks sehingga menjadi data yang lebih baik dan terstruktur sebelum memasuki tahapan selanjutnya. Tahapan data *preprocessing* yaitu *case folding*, *tokenizing*, *stopword removal*, dan *stemming.* Selanjutnya implementasi ke model LSTM melalui tahapan *indexing*, *sequences*, dan *padding*. Hasil dari model ini memiliki *accuracy* sebesar 70% dan sudah cukup baik dalam mendeteksi komentar *hate speech*.
4. Berdasarkan hasil evaluasi *confusion matrix* diketahui bahwa terdapat 240 data test dimana 84 data test dianggap *non hate speech* dan memang kenyataannya *non hate speech*, banyaknya data yang dideteksi salah sebagai *non hate speech* sebanyak 34, banyaknya data yang salah dideteksi sebagai *hate speech* namun sebenarnya *non hate speech* sebanyak 33, dan banyaknya data yang dideteksi benar sebagai *hate speech* adalah 84.

5. Dengan demikian dapat dikategorikan peneliti ini berhasil dalam melakukan pengklasifikasian komentar *hate speech* dan *non hate speech* melalui *Metode Long Short-Term Memory* (LSTM).

5.2. Saran

Berdasarkan klasifikasi *hate speech* dan *non hate speech* dalam penelitian ini masih ada kekurangan yang perlu diperbaiki dari hasil yang sudah didapatkan dari hasil penelitian ini. Adapun saran yang dapat diberikan untuk mengembangkan penelitian ini diantaranya adalah sebagai berikut:

1. Dari hasil sistem ini perlu diadakan penelitian lebih lanjut yaitu dengan menambahkan jumlah *dataset* khususnya untuk kalimat yang bermakna ambigu dan singkatan yang biasa digunakan pada media sosial.
2. Pada penelitian ini peneliti menyarankan agar dapat menambahkan metode evaluasi lain untuk melihat kinerja yang lebih optimal.

DAFTAR PUSTAKA

Universitas Muhammadiyah Yogyakarta, C. O. 2021. Analisis Yuridis Tindak Pidana Ujaran Kebencian Dalam Media Sosial. Al-Adl : Jurnal Hukum, 13(1), 168. https://doi.org/10.31602/al-adl.v13i1.3938

Al-Faruq, M. N. M., Nur'aini, S., & Aufan, M. H. 2022. Perancangan Ui/Ux Semarang *Virtual Tourism* Dengan *Figma.* Walisongo *Journal of Information Technology*, 4(1), 43–52. https://doi.org/10.21580/wjit.2022.4.1.12079

Antariksa, K., Purnomo WP, Y. S., & Ernawati, E. 2019. Klasifikasi Ujaran Kebencian pada Cuitan dalam Bahasa Indonesia. Jurnal Buana Informatika, 10(2), 164. https://doi.org/10.24002/jbi.v10i2.2451

APJII. 2023. *Survei* APJII Pengguna Internet di Indonesia Tembus 215 Juta Orang. Ipjii.or.Id. https://apjii.or.id/berita/d/*survei*-apjii-pengguna-internet-di-indonesia-tembus-215-juta-orang#:~:*text*=*Survei* APJII Pengguna Internet di,yang sebesar 275.773.901 jiwa.

Ariyanti, D., & Iswardani, K. 2020. Teks Mining untuk Klasifikasi Keluhan Masyarakat Pada Pemkot Probolinggo Menggunakan Algoritma *Naïve Bayes*. Jurnal IKRA-ITH Informatika, 4(3), 125–132.

Astari, Y. yuli, Afiyati, A., & Rozaqi, S. W. 2021. Analisis Sentimen *Multi-Class* Pada Sosial Media Menggunakan Metode *Long Short-Term Memory* (LSTM). Jurnal *Linguistik* Komputasional, 4(1), 8–12. http://inacl.id/journal/index.php/jlk/article/view/43

Bert, M. A. 2023. Deteksi *Hate Speech* Pada Twitter. 10(1), 256–262.

Fadli, H. F., & Hidayatullah, A. F. 2019. Identifikasi *Cyberbullying* Pada Media Sosial Twitter Menggunakan Metode Klasifikasi *Random Forest*. Automata.

Hernawati.2023.Pengaruh Media Sosial Terhadap Perilaku Masyarakat. Sulselprov.Go.I. https://sulselprov.go.id/*s*/post/pengaruh-media-sosial-terhadap-perilaku-masyarakat

Ihsan, F., Iskandar, I., Harahap, N. S., & Agustian, S. 2021. *Decision tree algorithm for multi-label hate speech and abusive language detection in Indonesian Twitter*. Jurnal Teknologi Dan Sistem Komputer, 9(4), 199–204. https://doi.org/10.14710/jtsiskom.2021.13907

Isnain, A. R., Sulistiani, H., Hurohman, B. M., Nurkholis, A., & Styawati. 2022. Analisis

Perbandingan Algoritma LSTM dan *Naive Bayes* untuk Analisis Sentimen. JEPIN (Jurnal Edukasi Dan Penelitian Informatika), 8(2), 299–303. https://jurnal.untan.ac.id/index.php/jepin/article/view/54704

Jubilee Enterprise. 2019. *Python* untuk *Programmer* Pemula.

Kerahasiaan, P., Perspektif, B., Perpajakan, H., Jan, T. S., Muttaqin, Z., & Sugiharti, D. K. 2020. Amanna Gappa. 28(2), 64–76.

Malut *center*. 2023. Masa Depan Maluku Utara : 93,76% Pengguna Internet Akan Apatis. Malut*center*.Com. https://malutcenter.com/2023/09/07/masa-depan-maluku-utara-9376-pengguna-internet-akan-apatis/

Nurvania, J., Jondri, & Lhaksamana, K. M. 2021. Analisis Sentimen Pada Ulasan di *TripAdvisor* Menggunakan Metode *Long Short-Term Memory* (LSTM). *E-Proceeding of Engineering*, 8(4), 4124–4135.

Parulian, H., & Putranto, R. D. 2022. Pidana Ujaran Kebencian Melalui Media Sosial Ditinjau dalam Perspektif Undang-Undang Nomor 19 Tahun 2016 tentang Perubahan Atas Undang Undang Nomor 11 Tahun 2008 tentang Informasi dan Transaksi Elektronik (UU ITE). Jurnal Pendidikan Dan Konseling (JPDK), 4(4), 4909–4919.

Permatasari, D. I., & Subyantoro2, S. 2020. Ujaran Kebencian *Facebook* Tahun 2017-2019. Jurnal Sastra Indonesia, 9(1), 62–70. https://doi.org/10.15294/jsi.v9i1.33020

Pramiyati, T., Jayanta, J., & Yulnelly, Y. 2017. Peran Data Primer Pada Pembentukan Skema Konseptual Yang Faktual (Studi Kasus: Skema Konseptual Basisdata Simbumil). Simetris : Jurnal Teknik Mesin, Elektro Dan Ilmu Komputer, 8(2), 679. https://doi.org/10.24176/simet.v8i2.1574

Prasetyo, V. R., Benarkah, N., & Chrisintha, V. J. 2021. Implementasi *Natural Language Processing* Dalam Pembuatan *Chatbot* Pada Program *Information Technology* Universitas Surabaya. Teknika, 10(2), 114–121. https://doi.org/10.34148/teknika.v10i2.370

Puspitarini, D. S., & Nuraeni, R. 2019. Pemanfaatan Media Sosial Sebagai Media Promosi (Studi Deskriptif pada *Happy Go Lucky House*). Jurnal *Common*, 3(1), 71–80. https://doi.org/10.34010/COMMON.V3I1.1950

Ratna, S. 2020. Pengolahan Citra Digital Dan Histogram Dengan *Phyton* Dan *Text Editor Phycharm. Technologia:* Jurnal Ilmiah, 11(3), 181.

https://doi.org/10.31602/tji.v11i3.3294

Renaldo Yosia Rafael, F. A. 2023. Pengimplmentasian algoritma *long short-term memory* untuk mendeteksi ujaran kebencian pada aplikasi twitter. 8(2), 551–560.

Rosaly, R., & Prasetyo, A. 2019. Pengertian *Flowchart* Beserta Fungsi dan Simbol-simbol *Flowchart* yang Paling Umum Digunakan. Https://Www.Nesabamedia.Com, 2, 2. https://www.nesabamedia.com/pengertian-*flowchart/https*://www.nesabamedia.com/pengertian-*flowchart/*

Salendah, J., Kalele, P., Tulenan, A., & ... 2022. Penentuan Beasiswa Dengan Metode *Fuzzy Tsukamoto* Berbasis *Web Scholarship Determination Using Web Based Fuzzy Tsukamoto Method. Proceeding* Seminar …, (nama), 81–90. https://proceeding.unived.ac.id/index.php/snasikom/*article/view/8*0%0A*https://proceeding.unived.ac.id/index.php/snasikom/article/download/80/70*

Talita, A. S., & Wiguna, A. 2019. Implementasi Algoritma *Long Short-Term Memory* (LSTM) Untuk Mendeteksi Ujaran Kebencian (*Hate Speech)* Pada Kasus Pilpres 2019. MATRIK : Jurnal Manajemen, Teknik Informatika Dan Rekayasa Komputer, 19(1), 37–44. https://doi.org/10.30812/matrik.v19i1.495

Udin, I. La. 2022. Klasifikasi Teks Pada Artikel Berita Menggunakan; Metode *Long Short Term Memory* (Lstm). 27.

Wahyono, T. 2018. *Fundamental of Python for Machine Learning*: Dasar-Dasar Pemrograman *Python* untuk *Machine Learning* dan Kecerdasan Buatan. Gava Media, September 2018, 49.

Widi Hastomo, Nur Aini, Adhitio Satyo Bayangkari Karno, & L.M. Rasdi Rere. 2022. Metode Pembelajaran Mesin untuk Memprediksi Emisi *Manure Management*. Jurnal Nasional Teknik Elektro Dan Teknologi Informasi, 11(2), 131–139. https://doi.org/10.22146/jnteti.v11i2.2586

Yusuf Sukman, J. (2017). Tindak Pidana Ujaran Kebencian *(Hate Speech)* Dalam Jejaring Media Sosial (Vol. 4).

LAMPIRAN