BAB II

LANDASAN TEORI

2.1 Penelitian Terdahulu

Dari pencarian yang dilakukan melalui media internet tentang analisis sentimen yang menggunakan media sosial sebagai sarana penelitian, maka ditemukan beberapa jurnal dan penelitian sebagai berikut:

Tabel 2.1 Penelitian Terdahulu

No	Penulis	Nama Jurnal	Judul	Metode	Kesimpulan
1	Fahmi Mahmuji Cholis, Muchammad Chandra Cahyo Utomo, Nisa Rizqiya Fadhliana (2023)	*Journal of Mathematics & Information Technology* (EQUIVA), Vol 1 No. 2, 2023 [10]	Analisis Sentimen Pada Twitter Terhadap Isu Penundaan Pemilu 2024 Dengan Membandingkan Metode *Long Short-Term Memory* Dan *Naïve Bayes Classifier*	*Long-Short Term Memory*	Akurasi, presisi, *recall,* dan *f1-score* kelas positif dan negatif sama yaitu 92%
2	Yuliya Astari, Afiyanti dan Saddam Wahib Rozaqi (2021)	Jurnal Linguistik Komputasional, vol. 4, no. 1, pp. 8-12 [11]	Analisis Sentimen Multi-Class Pada Sosial Media Menggunakan Metode *Long Short-Term Memory*	*Long Short-Term Memory* dengan nilai *epoch* 40	Akurasi 91.9% dan nilai rata-rata *multiclass* mendapatkan hasil 89.45%
3	Yuliana Romadhoni, Khadijah Fahmi Hayati Holle (2022)	Jurnal Informatika: Jurnal pengembangan IT (JPIT), Vol.7, No.2 [12]	Analisis sentimen terhadap PERMENDIKBUD No. 30 pada media sosial Twitter menggunakan metode LSTM	*Long-Short Term Memory*	Akurasi 77%, presisi 84%, *recall* 75%, dan *f1-score* 80%

No	Penulis	Nama Jurnal	Judul	Metode	Kesimpulan
4	Muhammad Zaini Rahman, Yuita Arum Sari, dan Novanto Yudistira (2021)	Jurnal Pengembangan Teknologi Informasi dan Ilmu Komputer, vol. 5, no. 11, pp. 5120-5127 [13]	Analisis Sentimen COVID-19 Menggunakan *Word Embedding* dan Metode *Long Short-Term Memory*	*Long Short-Term Memory*	Akurasi 81%, presisi 80%, *recall* 80%, dan *f-measure* 81%
5	Ni Putu Sintia Wati dan Cokorda Pramartha (2022)	Jurnal Nasional Teknologi Informasi dan Aplikasinya, Volume 1, Nomor 1, pp. 755-762 [14]	Penerapan *Long Short-Term Memory* dalam Mengklasifikasi Jenis Ujaran Kebencian Pada Bahasa Indonesia	*Long Short-Term Memory*	Akurasi 74%, presisi 74%, *recall* 77% dan *f1-score* 75%
6	Dinar Ajeng Kristiyanti dan Sri Hardani (2023)	JURNAL RESTI (Rekayasa Sistem dan Teknologi Informasi), vol. 7, no. 3, pp. 722 - 732 [15]	*Sentiment Analysis of Public Acceptance of Covid-19 Vaccines Types in Indonesia using Long Short-Term Memory*	*Long Short-Term Memory*	Akurasi 82.97%, presisi 84%, *recall* 89%, dan *f1-score* 91%
7	H Tsaniya, R Rosadi, dan A S Abdullah (2021)	*Journal of Physics: Conference Series, Volume 1722, Tenth International Conference and Workshop on High Dimensional Data Analysis* [8]	*Sentiment Analysis Towards Jokowi`s Government using Twitter Data with Convolutional Neural Network Method*	*Convolutional Neural Network* dengan CBOW	Akurasi CNN 57%, sentimen positif 38%, sentimen negatif 51%.

No	Penulis	Nama Jurnal	Judul	Metode	Kesimpulan
8	Ahmad Fathan Hidayatullah, Siwi Cahyaningtya, Anisa Miladya Hakim (2020)	*IOP Conference Series: Materials Science and Engineering, Volume 1077, The 5th International Conference on Information Technology and Digital Applications* [9]	*Sentiment Analysis on Twitter using Neural Network: Indonesian Presidential Election 2019 Dataset*	*Convolutional Neural Network, Long Short-Term Memory,* CNN-LSTM, *Gated Recurrent Unit* LSTM dan *Bidirectional* LSTM	Akurasi LSTM 84.20%, akurasi CNN 84.05%, akurasi CNN+LSTM 84.30%, akurasi GRU+LSTM 84.50%, dan akurasi *Bidirectional* LSTM 84.60%
9	Putri Arta Aritonang, Monika Evelin Johan, dan Iwan Prasetiawan (2022)	Ultima InfoSys: Jurnal Ilmu Sistem Informasi vol. 13, no. 1, pp. 54-61 [16]	*Aspect-Based Sentiment Analysis on Application Review using CNN (Case Study: Peduli Lindungi Application)*	*Convolutional Neural Network* dan *Adam optimizer* dengan *learning rate* 0.0001	Akurasi 95.10%, presisi 95.14%, *recall* 95.10%, *dan f1-score* 95.13%, sentimen negatif *f1-score* 97%, dan sentimen positif *f1-score* 86.82%
10	Brescia Ayundina Yuniarossy, Kartika Maulida Hindrayani, Aviolla Terza (2024)	Lebesgue: Jurnal Ilmiah Pendidikan Matematika, Matematika dan Statistika, Vol. 5, No. 1 [17]	Analisis Sentimen Terhadap Isu Feminisme di Twitter Menggunakan Model *Convolutional Neural Network* (CNN)	*Convolutional Neural Network*	Akurasi 86%, kelas negatif nilai presisi 86%, *recall* 95%, *f1-score* 90%, sedangkan kelas positif presisi 84%, *recall* 66%, *f1-score* 74%

Pada tabel 2.1 berisikan penelitian terdahulu yang dijadikan referensi pada pembuatan laporan ini. Pada penelitian pertama yang dilakukan oleh oleh F. M. Cholis, M. C. Utomo. dan N. R. Fadhliana terkait analisis sentimen data X isu pemerintah khususnya pemilu didapatkan akurasi terbaik oleh metode LSTM

dengan nilai kurasi, presisi, *recall*, dan *f1-score* pada kelas positif dan negatif sama yaitu 92% [10]. Pada penelitian kedua oleh Y. Astari, Afiyanti dan S.W. Rozaqi menggunakan metode LSTM dengan nilai *epoch* 20 diperoleh akurasi data *training* 98% dan data *testing* 66%. Sedangkan untuk hasil akurasi uji coba *multiclass* dengan delapan label yang berbeda menggunakan nilai *epoch* 40 menghasilkan akurasi data *training* 91.9% dan data *testing* 25.5%, sehingga nilai rata-rata akurasi sebesar 89.45% [11]. Pada penelitian ketiga oleh Y. Romadhoni, K. F. H. Holle menggunakan metode pembobotan kata TF-IDF baru dilanjutkan dengan LSTM. dan menghasilkan akurasi sebesar 77%, presisi sebesar 84%, *recall* sebesar 75%, dan *f1-score* sebesar 80% [12]. Pada penelitian keempat oleh M. Z. Rahman, Y. A. Sari, dan N. Yudistira menggunakan metode LSTM dengan penambahan *Word Embedding* menghasilkan akurasi 81%, presisi 80%, *recall* 80%, dan *f-measure* 81% [13]. Pada penelitian kelima oleh N. P. S. Wati dan C. Pramartha juga menggunakan LSTM dengan *Word Embedding* menghasilkan nilai akurasi 74%, presisi 74%, *recall* 77%, dan *f1-score* 75% [14]. Pada penelitian keenam oleh D. A. Kristiyanti dan S. Hardani menggunakan metode LSTM dengan pembobotan kata TF-IDF menghasilkan akurasi 82.97%, presisi 84%, *recall* 89%, dan *f1-score* 91% [15]. Pada penelitian ketujuh oleh H. Tsaniya, R. Rosadi, dan A. S. Abdullah menggunakan metode CNN dengan *Word Embedding* CBOW mencapai akurasi 57%, dengan kelas sentimen positif 38%, sentimen negatif 51% [8]. Penelitian kedelapan oleh A.F. Hidayatullah, S. Cahyaningtya, A. M. Hakim pada metode LSTM menghasilkan akurasi 84.2%, CNN 84.05%, CNN+LSTM 84.30%, GRU+LSTM 84.50%, dan *Bidirectional* LSTM 84.60% [9]. Pada penelitian kesembilan oleh P A. Aritonang, M. E. Johan, dan I. Prasetiawan menggunakan metode CNN dan pada model klasifikasi aspek menghasilkan akurasi dan *recall* 92.24%, presisi 92.34%, dan *f1-score* 92.23%, sedangkan pada model klasifikasi sentimen menghasilkan akurasi dan *recall* 95.10%, presisi 95.14%, dan *f1-score* 95.13% [16]. Pada penelitian kesepuluh menggunkan metode CNN menghasilkan

akurasi 86%, pada kelas negatif nilai presisi 86%, *recall* 95%, *f1-score* 90%, sedangkan pada kelas positif presisi 84%, *recall* 66%, *f1-score* 74% [17].

Berdasarkan penelitian terdahulu dapat disimpulkan bahwa analisis sentimen menggunakan *deep learning* metode LSTM dan CNN berhasil memberikan hasil yang baik. Penelitian ini berbeda dengan penelitian terdahulu dimana pada salah satu tahap *preprocessing,* yaitu *normalization* meggunakan tiga kamus bahasa tidak baku Indonesia dan pada tahap *labeling* menggunakan kamus *InSet Lexicon*. Pada tahap *features extraction* membandingkan penggunaan TF-IDF dan *Word Embedding* pada model LSTM dan CNN serta membandingkan penggunaan *optimizer Adam, RMSprop, SGD, Adagrad, Adadelta, Adamax,* dan *Nadam* untuk mendapatkan hasil terbaik. Melalui penelitian ini, diperoleh pemahaman lebih jelas mengenai respon publik pada X terhadap kinerja pemerintahan termasuk prioritas kerja Jokowi. Selain itu, memberi pemahaman tentang kinerja dan keefektifan LSTM dan CNN dengan metode-metode yang digunakan dan paramater yang disesuaikan pada masing-masing model.

2.2 Analisis Sentimen

Analisis Sentimen, sering dikenal sebagai penambangan opini, adalah studi komputasi mengenai emosi dan pandangan yang direpresentasikan dalam teks tertulis [18]. Analisis Sentimen adalah metode yang digunakan untuk memahami, menganalisis, dan secara otomatis mengekstrak data tekstual untuk mendapatkan informasi tentang sentimen atau makna yang mendasari data dalam kalimat yang menyatakan pendapat. Analisis Sentimen adalah teknik komputasi dan linguistik yang digunakan untuk menganalisis opini atau sentimen individu terhadap topik atau aktivitas tertentu. Analisis sentimen melibatkan pengkategorian teks ke dalam kalimat individual atau titik data untuk mengidentifikasi apakah setiap kalimat itu positif, negatif, atau netral [19]. Oleh karena itu, analisis sentimen merupakan suatu metode yang digunakan untuk memastikan sudut pandang atau pandangan individu yang disampaikan melalui informasi tekstual, yang dapat dikategorikan sebagai emosi

positif, negatif, atau netral. Berdasarkan sumber datanya, analisis sentimen dapat dikategorikan menjadi dua kelompok utama [20] :

a) *Coarse-grained Sentiment Analysis*

Analisis sentimen dilakukan per dokumen. Biasanya, bentuk analisis sentimen ini terutama memeriksa keseluruhan isi dokumen, mengkategorikannya sebagai sentimen positif atau negatif.

b) *Fined-grained Sentiment Analysis*

Analisis sentimen dilakukan berdasarkan per kalimat. Analisis sentimen sebagian besar melibatkan pemeriksaan data dari kalimat individual.

2.2.1 *Natural Language Processing (NLP)*

Sebuah subbidang ilmu komputer dan kecerdasan buatan, *Natural Language Processing* (NLP) meneliti hubungan antara komputer dan bahasa alami manusia. *Natural Language Processing* (NLP) terutama berkaitan dengan memungkinkan komputer untuk secara efektif memahami, menganalisis, dan menghasilkan bahasa manusia [21]. *Natural Language Processing* (NLP) adalah domain penting dalam kecerdasan buatan yang terutama meneliti interaksi manusia-mesin / komputer melalui penggunaan bahasa alami [22]. Pemrosesan bahasa alami (NLP) diimplementasikan dalam banyak konteks untuk mengatasi beragam tantangan, diantaranya [23]:

a) Mencari serangkaian karakter.

Pemanfaatan pemrosesan bahasa alami otomatis memungkinkan identifikasi elemen tekstual tertentu. Misalnya hal ini digunakan dalam mencari dokumen pada contoh sebutan tertentu atau, lebih umum, urutan karakter.

b) Pengenalan entitas.

Berguna untuk mencari atau mengekstrak nama lokasi, individu, organisasi, maupun produk melalui teks.

c) Analisis sentimen

Dengan menggunakan metode ini, sentimen dan perspektif individu mengenai suatu produk atau layanan dapat dipastikan. Hal ini bermanfaat untuk mendapatkan umpan balik mengenai cara konsumen atau pengguna merasakan produk atau layanan. Data jejaring sosial dapat dimanfaatkan oleh bisnis di semua industri untuk mendapatkan pemahaman yang lebih dalam tentang sentimen konsumen. Hambatan utama adalah secara mekanis mengubah ulasan konsumen ke dalam format teks untuk digunakan dalam menentukan produk atau merek terbaik.

d) Mesin pencari

Berbagai aplikasi menggunakan terjemahan bahasa alami (NLP), misalnya *query understanding, query expansion, question answering*, pencarian informasi (*information retrieval*), pemeringkatan, dan pengelompokan hasil (*clustering of results*).

e) Pesan elektronik

Platform email juga menggunakan pemrosesan bahasa alami (NLP) yang dapat diimplementasi untuk melakukan klasifikasi spam email, prioritas kotak masuk, dan pelengkapan otomatis.

Analisis sentimen telah muncul sebagai bidang studi yang sangat dinamis dalam pemrosesan bahasa alami (NLP) sejak awal tahun 2000-an. Selain itu, penambangan data, penambangan web, penambangan teks, dan pengambilan informasi semuanya mencurahkan perhatian yang cukup besar untuk itu. Saat ini, karena signifikansinya bagi bisnis dan masyarakat luas, NLP sedang diimplementasikan dalam ilmu manajemen, ilmu sosial,

komunikasi, ilmu kesehatan, pemasaran, keuangan, ilmu politik, dan bahkan sejarah, selain ilmu komputer [22].

2.2.2 *Deep Learning*

Penerapan jaringan saraf tiruan untuk pembelajaran dengan jaringan beberapa lapisan dikenal sebagai *Deep Learning* [22]. *Deep Learning* adalah cabang *Machine Learning* dan *Artificial Intelligence* yang memerlukan pembangunan jaringan saraf berlapis-lapis untuk menawarkan akurasi dalam pekerjaan seperti terjemahan bahasa, pengenalan suara, dan deteksi objek [24]. Metode menggunakan konsep hierarki untuk memecahkan masalah dalam sistem pembelajaran komputer disebut *Deep Learning*. Komputer dapat memperoleh ide-ide rumit dengan mengintegrasikan yang lebih sederhana berkat prinsip hierarki [25]. Adapun kategori *deep learning* yakni sebagai berikut [24]:

a) *Deep Learning* untuk Pembelajaran Tanpa Pengawasan (*Unsupervised Learning*)

   Pembelajaran mendalam tipe ini diterapkan ketika analisis pola membutuhkan nilai korelasi yang lebih baik untuk dihitung dari unit yang diamati dan label variabel target tidak dapat diakses.

b) *Hybrid Deep Networks* (*Deep Learning* gabungan)

   Menggunakan pembelajaran yang diawasi atau mungkin pembelajaran tanpa pengawasan, metode semacam ini berusaha memberikan hasil yang baik dalam analisis pola..

Adapun tiga kategori teknik dalam *Deep Learning*, yaitu *supervised, semi-supervised*, dan *unsupervised*. Kategori lain, seperti *Reinforcement Learning* (RL) atau *Deep RL* (DRL) [26]:

a) *Deep Supervised Learning*

b) Metode pembelajaran yang diterapkan dalam kategori ini

memanfaatkan data berlabel (labeled data). *Deep Neural Networks* (DNN), CNN, RNN, LSTM, dan *Gated Recurrent Unit* (GRU) merupakan contoh metode populer dari *Deep Supervised Learning.*

c) *Deep Semi-Supervised Learning*

Metode pembelajaran yang menggunakan data berlabel sebagian (partially labeled data) digunakan dalam pendekatan pembelajaran *Semi-supervised learning*. DRL, RNN, LSTM, GRU, serta *Generative Adversarial Networks* (GAN adalah contoh di kelas ini.

d) *Deep Unsupervised Learning*

Data tanpa label (unlabeled data) digunakan pada *deep unsupervised learning*. Dalam kategori ini termasuk *Auto Encoders* (AE), *Restricted Boltzmann Machines* (RBM), hingga generasi GAN terbaru.

e) *Deep Reinforcement Learning*

Pada teknik *deep reinforcement learning* menggunakan pengaturan atau lingkungan yang tidak diketahui (*unknown environments*).

2.2.3 *Web Scraping*

Web scraping prosedur sistematis di mana dokumen semi-terstruktur yang diperoleh dari internet, biasanya halaman web yang diformat dalam bahasa markup seperti HTML atau XHTML, dianalisis untuk mendapatkan data spesifik yang berlaku untuk konteks yang berbeda [27]. *Scraping web*, dikenal juga dengan ekstraksi web atau pemanenan, adalah metode yang digunakan untuk mengambil atau menganalisis data dari *World Wide Web* (WWW) dengan menyetornya ke dalam *database* atau sistem file [28]. Tahapan umum *web scraping* yaitu [29]:

a) Memanfaatkan HTTP untuk memperoleh sumber daya yang ditargetkan
b) Server akan menjalankan permintaan dalam bentuk URL yang terdiri dari permintaan GET / HTTP yang mencakup tindakan POST
c) Sumber daya yang diminta akan diambil dan dikembalikan dalam berbagai format setelah diterima

Web scraping adalah proses yang melibatkan pengambilan dan ekstraksi data. Keuntungan dari proses *web scraping* adalah menghasilkan informasi yang lebih terkonsentrasi, yang memfasilitasi inti pertanyaan. Tujuan utama dari aplikasi *web scraping* adalah untuk memperoleh, mengambil, dan mengekstrak data sesuai dengan besarnya data [30]. *Web scraping* terdiri dari tahap-tahap berikut [31]:

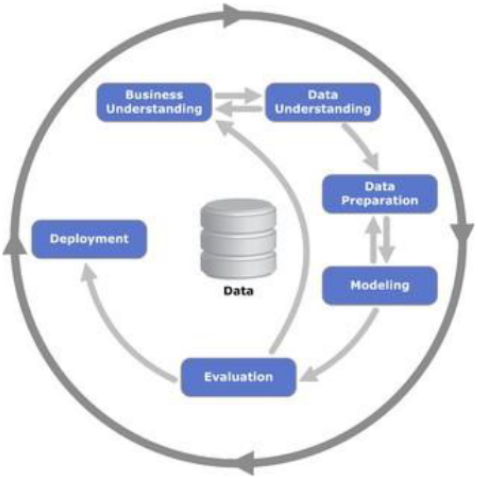
a) *Create scraping template* melalui pemeriksaan dokumen HTML yang ada di situs web target (di mana informasi tersebut akan diekstraksi).
b) *Explore site navigation* pada *web scraper*, mereplikasi navigasi situs web dari mana informasi akan diambil.
c) *Automate navigation and extraction* memanfaatkan informasi yang diperoleh. Tujuan pengembangan aplikasi *web scraper* adalah untuk mengotomatiskan proses penggalian data dari situs web yang dipilih.
d) *Extracted data and package history*, informasi yang diperoleh dan data yang diekstraksi disimpan dalam tabel *database*.

2.3 *Cross Industry Standard Process for Data Mining* (CRISP-DM)

CRISP-DM didirikan pada tahun 1996 untuk memfasilitasi penerapan proses industri bisnis untuk penyelidikan ilmiah [32]. Para analis di Daimler Chrysler, SPSS, dan NCR, antara lain, berkontribusi pada pengembangan CRISP-

DM. Sebagai pendekatan umum untuk pemecahan masalah bagi organisasi dan perusahaan riset, CRISP-DM membakukan prosedur penambangan data. Pola dan signifikansi eksplorasi dicari dalam data yang digunakan menggunakan metode CRISP-DM [33].

Umumnya digunakan untuk tujuan *data mining*, CRISP-DM adalah model proses industri-independen. Diimplementasikan secara luas dalam proyek analitik, *data mining*, serta *data science* [34]. CRISP-DM dianggap sebagai metodologi penambangan data paling komprehensif dalam hal memenuhi persyaratan proyek industri. Terdapat enam tahapan, yaitu *Business Understanding, Data Understanding, Data Preparation, Modeling, Evaluation*, dan *Deployment* [32]. Penjelasan tentang enam fase CRISP-DM disediakan dalam metodologi CRISP-DM di bawah ini:



Gambar 2.1 Metodologi CRISP-DM [33]

Berdasarkan gambar metodologi CRISP-DM, berikut ini penjelasan enam tahapan CRISP-DM:

2.3.1 *Business Understanding*

Business Understanding adalah prosedur mendefinisikan tujuan penelitian yang dilakukan terhadap masalah yang diselesaikan oleh *data*

mining, pemahaman situasi dan kondisi saat penelitian [32]. Langkah pertama dalam fase ini adalah memahami masalah bisnis yang harus diselesaikan atau tujuan yang seharusnya dicapai oleh proyek penambangan data. Tahap ini dimulai dengan memahami masalah bisnis yang ingin diselesaikan atau tujuan yang ingin dicapai dengan proyek *data mining.*

2.3.2 *Data Understanding*

Data understanding adalah langkah persiapan yang melibatkan verifikasi data yang digunakan, pengumpulan data awal, dan menentukan kualitas data. Setiap karakteristik data yang digunakan dalam pemahaman data akan dijelaskan [33]. Langkah pertama dalam fase ini adalah mengumpulkan, mengkarakterisasi, dan menilai kualitas data. Pengumpulan data adalah salah satu aspek pemahaman data, juga mencakup persiapan dan penilaian kebutuhan [34].

2.3.3 *Data Preparation*

Data preparation merupakan proses yang dilakukan setelah data telah dikumpulkan. Pada tahap ini, data akan melalui proses identifikasi, pemilihan data, pembersihan data dan transformasi data. Pada tahap *data preparation* pertama akan dilakukan *data selection* untuk melakukan seleksi terhadap data yang dibutuhkan, *data preprocessing* untuk mempersiapkan data mentah sehingga siap untuk menjadi data yang dapat digunakan pada pemodelan, dan data *transformation* untuk melakukan transformasi data [32].

2.3.3.1 *Preprocessing*

Data preprocessing adalah fase di mana operasi pengantar dilakukan pada data. Pada titik ini, data yang menjalani pemrosesan berusaha untuk menghilangkan data yang mengganggu (noise) atau informasi yang tidak konsisten. Tujuan dari *data preprocessing*

adalah untuk mengubah data yang tidak diproses menjadi data berkualitas tinggi, sehingga mempersiapkannya untuk tahap pemrosesan selanjutnya [34]. Sejumlah langkah terdiri dari prapemrosesan data, termasuk pembersihan, *case folding*, *tokenizing*, *filtering*, *stemming*, dan koreksi kata-kata tidak baku atau normalisasi bahasa.

a) *Cleaning*

Cleaning mengacu pada proses menghilangkan karakter yang tidak patuh, seperti huruf atau karakter di luar rentang alfabet 'a' sampai 'z' (termasuk tanda baca), serta menghapus tautan atau URL, tagar, simbol, emoji, dan nama pengguna [35].

b) *Case Folding*

Case Folding mengacu pada proses mengonversi semua karakter dalam teks menjadi huruf kecil atau huruf besar. Prosedur mengubah semua karakter dalam teks yang diberikan menjadi huruf kecil dan hanya huruf 'a' hingga 'z' yang ditangani sebagai karakter [35].

c) *Tokenizing*

Tokenizing merupakan tindakan membagi urutan kalimat atau *string* menjadi kata atau segmen individual, yang dikenal sebagai token [35]

d) *Filtering*

Filtering atau *stopword* adalah Proses menghapus elemen yang tidak diinginkan atau tidak perlu dari sekumpulan data atau informasi. Pemfilteran, juga dikenal sebagai penghapusan *stopword*, adalah proses mengekstraksi kata-kata penting dari hasil tokenization. Algoritma ini

mencakup stoplist yang menghilangkan kata-kata yang tidak perlu, dan *wordlist* yang menyimpan kata-kata penting [36]. Selama tahap penyaringan, akan ada proses menghilangkan istilah yang kurang bermakna atau kurang penting dalam data [37].

e) *Stemming*

Stemming mengacu pada proses mengurangi kata-kata ke bentuk dasar atau akarnya. Ini biasanya digunakan dalam pemrosesan bahasa alami untuk menyederhanakan variasi kata dan meningkatkan analisis teks. *Stemming* adalah mengubah kata-kata yang berasal dari penyaringan ke dalam bentuk dasarnya dengan menghilangkan imbuhan dalam kata-kata yang termasuk dalam dokumen [36].

f) Normalisasi bahasa.

Normalisasi yaitu melibatkan pengubahan dan perbaikan kata-kata yang disingkat menjadi istilah yang memiliki arti yang sama menurut KBBI, sehingga mengubahnya menjadi informasi yang mudah diproses. Prosedur normalisasi dilakukan untuk mengatasi masalah berbagai singkatan kata, penggunaan *slang*, dan masalah ejaan [18].

2.3.3.2 *Sentimen Labeling*

Langkah selanjutnya adalah memberi label kelas sentimen setiap bagian data setelah langkah pembersihan data. Dalam penelitian ini, *InSet Lexicon* digunakan untuk memberi label pada data. Kamus leksikon ini adalah cara untuk mengurutkan data teks yang telah diterjemahkan ke dalam bahasa Indonesia [18] Dengan menggunakan kamus kata-kata opini untuk mencari tahu kategori

kelas dari setiap kalimat, apakah termasuk positif atau negatif. *InSet Lexicon* digunakan sebagai kamus kata kunci untuk karya ini.

Tujuan pelabelan data adalah untuk menetapkan sebutan positif dan negatif untuk semua data yang telah mengalami prapemrosesan. Dalam membangun model, formasi pelabelan ini akan berfungsi sebagai dasar untuk algoritma klasifikasi. Prosedur pelabelan data menggunakan metode berbasis leksikon karena kepraktisan, kesederhanaan, dan kesesuaiannya dalam analisis sentimen yang memanfaatkan data media sosial [18].

2.3.3.3 *Feature Extraction*

Data teks memerlukan persiapan khusus sebelum mulai menggunakannya untuk pemodelan. Teks harus diuraikan untuk menghapus kata-kata, yang disebut *tokenization*. Kemudian kata-kata perlu dikodekan sebagai bilangan bulat atau nilai *floating point* untuk digunakan sebagai input ke algoritma *deep learning*, yang disebut ekstraksi fitur (vectorization). *Feature Extraction* atau ektrasi fitur adalah salah satu teknik yang berguna untuk memperoleh fitur yang paling representatif sehingga membantu dalam analisis data [35]. Ekstraksi fitur memiliki peran dalam menghilangkan fitur asing untuk meningkatkan akurasi klasifikasi dengan mengurangi *noise.*

a) *Term Frequency Inverse Document Frequency* (TF-IDF)
Pendekatan TF-IDF adalah teknik yang digunakan untuk menentukan pentingnya istilah dalam pencarian informasi berdasarkan frekuensi kemunculannya. Metode ini menghitung bobot hubungan antara kata (istilah) dan kata dalam dokumen. Ini adalah ukuran statistik yang dapat digunakan untuk menilai kata-kata penting dalam

dokumen. Frekuensi istilah dalam dokumen adalah ukuran seberapa sering kata tersebut digunakan [38]. Metode ini menggunakan rumus matematika untuk menghitung bobot kata yaitu:

$$wdt = tfdt \times idft \quad (2.1)$$

Keterangan:

wdt : bobot dokumen ke-d terhadap kata ke-t

$tfdt$: banyak kata yang dicari dalam dokumen

$idft$: *Inversed Document Frequency* (log (N/df))

N : total dokumen

df : banyak dokumen yang mengandung kata yang dicari

b) *Word Embedding*

Word embedding diugnakan dalam pemrosesan bahasa alami (Natural Language Processing) sebagai salah satu teknik mengubah kata-kata menjadi representasi numerik dalam bentuk vektor berdimensi tinggi [35]. Proses ini dikenal sebagai *feature extraction*, di mana informasi dari teks mentah diekstraksi menjadi fitur-fitur yang dapat digunakan oleh algoritma *machine learning*. *Word embedding* bertujuan untuk memetakan kata-kata ke dalam ruang vektor kontinu sehingga setiap kata direpresentasikan sebagai sebuah vektor. Vektor-vektor ini memungkinkan algoritma *machine learning*

untuk memahami hubungan semantik antar kata berdasarkan posisi mereka dalam ruang vektor tersebut [13].

Beberapa teknik dan model yang populer digunakan dalam *word embedding* adalah:

- *Word2Vec*:
  - *Continuous Bag of Words* (CBOW): Memperkirakan kata target berdasarkan konteks sekelilingnya.
  - *Skip-gram*: Memperikirakan konteks kata berdasarkan kata target. Model ini dilatih menggunakan korpus teks yang besar untuk menangkap hubungan statistik antara kata-kata.
- *GloVe* (Global Vectors for Word Representation):
  - Menggunakan matriks ko-occurrence yang mencerminkan frekuensi kemunculan bersama kata-kata dalam korpus teks.
  - Menyediakan representasi vektor yang menggabungkan informasi global dari teks.
- *FastText*:
  - Memperluas *Word2Vec* dengan mempertimbangkan sub-kata (n-gram), sehingga lebih efektif dalam menangkap informasi dan menangani kata-kata yang jarang atau baru.

2.3.3.4 *Data Spliting*

Pemisahan data adalah fitur penting dari ilmu data, terutama ketika mengembangkan model berbasis data. Teknik ini membantu memastikan bahwa model data yang dikembangkan akurat dan dapat digunakan dalam tahap selanjutnya, seperti *machine learning* [20]. *Split data* terbagi menjadi dua set data, sebagai berikut:

a) *Data training* atau pelatihan mmengacu pada data yang digunakan untuk melatih model.
b) *Data testing* atau pengujian mengacu pada data yang digunakan untuk mengetes/menguji performa model yang dihasilkan oleh data *training*.

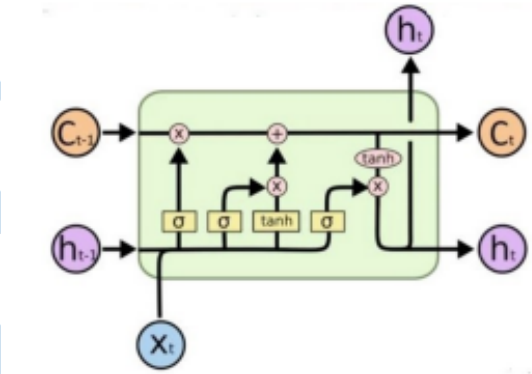
2.3.4 *Modeling*

Pemodelan adalah tahap implementasi algoritma di mana pola dicari, diidentifikasi, dan diproduksi untuk digunakan dalam data penelitian [21].

2.3.4.1 *Long Short-Term Memory* (LSTM)

Hochreiter dan Schmidhuber adalah orang pertama yang berbicara tentang *Long Short-Term Memory* (LSTM) pada tahun 1997 [39]. *Long Short-Term Memory* (LSTM) menjadi salah satu metode *Deep Learning* yang sering dimanfaatkan pada *Natural Language Processing* (NLP) seperti pengenalan suara, translasi teks, termasuk analisis sentimen. LSTM merupakan jenis dari metode *Recurrent Neural Network* (RNN), tetapi kekurangan dari arsitektur RNN konvensional ialah tidak dapat mengingat hal-hal yang terjadi di jaringan dalam jangka waktu yang lama. Oleh karena itu, metode LSTM dikembangkan untuk menyelesaikan permasalahan *vanishing gradient* yang kerap ditemukan pada RNN [13]. Sel memori LSTM dapat menyimpan data dan mengakses

informasi yang diurutkan ke dalam kelompok meskipun pada jangka waktu yang lama [25]. Sel memori LSTM mampu menyimpan data dan mengakses informasi dalam jangka waktu yang lebih lama sehingga LSTM tediri dari tiga *gate*, yaitu *forget gate*, *input gate* dan *ouput gate, gate* mengontrol bagaimana informasi teks masa lalu digunakan dan diperbarui di LSTM dengan desain arsitekur LSTM yang ditunjukan pada Gambar 2.2.



Gambar 2.2 Arsitektur LSTM [40]

Dalam hal ini, *gates* memutuskan apakah informasi cukup penting untuk diingat atau tidak [13]. *Forget gate* adalah gerbang pertama dan terjadinya pengurutuan informasil ke dalam sel, kemudian lapisan *sigmoid* membuat pilihan. Ketika angka keputusannya adalah 1, itu berarti "bisa lulus." Ketika angka keputusan adalah 0, itu berarti "lupakan informasi". Persamaan berikut dapat digunakan untuk mencari nilai *forget gate* (f_t) [40]:

$$f_t = \sigma\left(W_f\left[h_{t-1,}X_t\right] + b_t\right) \quad (2.2)$$

Keterangan:

f_t	:	*Forget gate*
X_t	:	Input variabel x ke t
σ	:	Fungsi Sigmoid
W_f	:	Bobot *forget gate*
$h_{t-1,}$	:	*Hidden State* sebelumnya
b_t	:	Bias *forget gate*

Setelah perhitungan *forget gate* selesai, langkah selanjutnya adalah mencari tahu data baru apa yang disimpan dalam status sel. *Input gate*, lapisan pertama *sigmoid*, memutuskan bagian mana yang akan diperbarui. *Layer tanh* kemudian membuat vektor nilai kandidat baru. Nilai c_t dapat ditambahkan ke status sel, lalu keduanya dapat disatukan untuk memperbarui status. Persamaan berikut dapat digunakan untuk menemukan nilai *input gate* [40]:

$$i_t = \sigma(W_i[h_{t-1}, X_t] + b_i) \quad (2.3)$$

$$\hat{C}_t = \tanh(W_c\,[h_{t-1}, X_t] + b_c) \quad (2.4)$$

Keterangan:

i_t	:	*Input gate*
$\hat{C}_t$	:	*Cell* aktivasi
σ	:	Fungsi Sigmoid
$tanh$	:	Fungsi aktivasi
X_t	:	Input variabel x ke t
W_i	:	Bobot *input gate*

W_c	:	Bobot *cell* aktivasi
$h_{t-1,}$	:	*Hidden State* sebelumnya
b_i	:	Bias *input gate*
b_c	:	Bias *cell* aktivasi

Langkah selanjutnya adalah memperbaharui *cell state* yang lama C_{t-1} ke *cell state* yang baru yakni C_t setelah itu mengalihkan *cell state* yang lama dengan *forget gate* dan ditambah $i_t * \check{C}_t$. Nilai C_t atau *state* baru dapat dihitung dengan persamaan berikut ini [40]:

$$C_t = f_t * C_{t-1} + i_t * \check{C}_t \quad (2.5)$$

Keterangan:

f_t	:	*Forget gate*
i_t	:	*Input gate*
C_t	:	*Cell* gate
$\check{C}_t$	:	*Cell* aktivasi
X_t	:	Input variabel x ke t
$C_{t-1,}$	:	*Cell gate* sebelumnya

Setelah kalkulasi C_t *sigmoid layer* mentukan sel lain untuk dijadikan sebagai *output* (h_t), kemudian penempatan *cell state* melewati *tanh* memperbesar jumlah *output* dari *sigmoid gate* (O_t). Nilai *output gate* dapat dihitung dengan persamaan berikut ini [40]:

$$O_t = \sigma(W_0[h_{t-1}, X_t] + b_0) \quad (2.6)$$

$$h_t = O_t * \tanh(C_t) \quad (2.7)$$

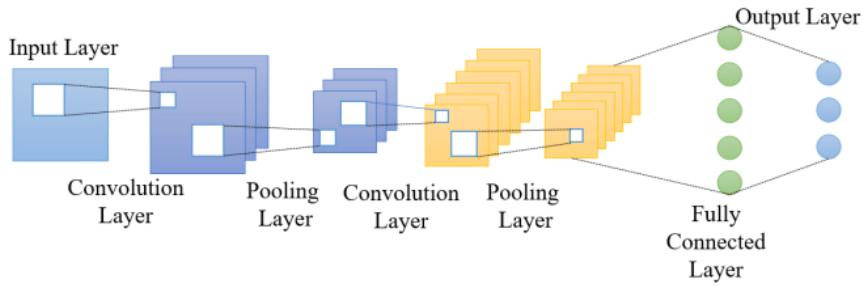
Keterangan:

O_t	:	*Output gate*
W_0	:	Bobot *output gate*
σ	:	Fungsi Sigmoid
$tanh$	:	Fungsi aktivasi
X_t	:	Input variabel x ke t
$h_{t-1,}$	:	*Hidden State* sebelumnya
b_0	:	Bias *output gate*
C_t	:	*Cell* gate

Fungsi $tanh$ membmengevaluasi nilai-nilai yang dilewati dan menentukan suatu tingkat kepentingannya (-1 hingga 1). Status sel baru diperbarui dengan mengalikan dua nilai, Kemudian, memori C_{t-1} lama ditambahkan, menghasilkan C_t. Parameter C_{t-1} dan C_t adalah kondisi sel pada waktu $(t-1)$ dan (t) [11].

2.3.4.2 *Convolutional Neural Network* (CNN)

Convolutional Neural Network (CNN) adalah arsitektur pembelajaran mendalam yang biasa digunakan untuk memproses input gambar dan teks. CNN adalah konstituen dari *Deep Neural Network* (DNN). CNN memanfaatkan operasi konvolusi sebagai metode perkalian matriks, khususnya dalam satu lapisan tertentu [41].



Gambar 2.3 Arsitektur CNN [42]

Convolutional neural network diklasifikasikan sebagai *deep neural network* karena kedalaman jaringannya yang besar dan aplikasi yang luas untuk data teks dan gambar. Secara teknis, CNN adalah arsitektur yang dapat dilatih yang terdiri dari banyak fase. Setiap tahap proses melibatkan penggunaan banyak *array* yang dikenal sebagai peta fitur untuk *input* dan *output*. Tahapan dalam proses ini terdiri dari tiga lapisan berbeda: lapisan konvolusi, lapisan aktivasi, dan lapisan pooling [58]. CNN terdiri dari tiga jenis lapisan saraf utama, yaitu, *convolutional layers*, *pooling layers*, dan *fully connected layers* [36] :

a) Convolutional layer

CNN menggabungkan seluruh gambar dan peta fitur menengah menggunakan berbagai parameter pada lapisan konvolusional, menghasilkan berbagai peta fitur. *Array* konvolusi mengekstrak objek dari gambar input melalui *filter*. *Filter* terdiri dari bobot yang digunakan untuk menentukan fitur objek, termasuk warna, tepi, dan kurva [43]. Ketika input memasuki lapisan ini, operasi konvolusi yang melibatkan *filter* diterapkan ke jendela kata untuk menghasilkan fitur baru. *Filter* diterapkan berulang kali ke setiap jendela kata dalam kalimat untuk menghasilkan peta fitur [16].

b) Pooling Layers

Mengurangi dimensi spasial (lebar × tinggi) dari volume input untuk lapisan konvolusional berikutnya adalah tugas dari *pooling layers*. Lapisan penggabungan tidak berpengaruh pada dimensi kedalaman volume. Proses yang dijalankan oleh lapisan ini secara alternatif disebut sebagai *subsampling* atau *downsampling* karena fakta bahwa penurunan dimensi mengakibatkan hilangnya data secara bersamaan. Lapisan ini secara bertahap mengurangi jumlah parameter, kompleksitas komputasi model, dan kontrol *overfitting*. *Max-over-time pooling* sering diterapkan pada peta fitur untuk mengambil fitur yang paling penting (fitur dengan nilai tertinggi) untuk setiap peta [16].

c) Fully Connected Layers

Pada lapisan *fully connected* dibentuk *neuron* satu dimensi dan mencakup *neuron* yang berkesinambungan dengan neuron di lapisan sebelumnya juga sesudahnya. Regularisasi dapat dilakukan pada lapisan ini menggunakan fungsi *dropout* guna *neuron* tetap pada nilai probabilitas antara 0 dan 1, sehingga lebih mudah untuk mengklasifikasikan kelas *output*. Lapisan ini juga akan menampilkan jumlah kelas yang ditentukan menggunakan aktivasi *softmax* [16].

Dikarenakan strukturnya yang kompleks, CNN mencakup banyak lapisan representasi. Oleh karena itu, CNN dapat secara otomatis memperkirakan dan mendapatkan atribut representasi dari data melalui transformasi nonlinier. Struktur CNN terdiri dari lapisan konvolusional untuk mengekstraksi fitur dari gambar, sedangkan lapisan pooling dan pengklasifikasi *softmax* mengurangi dimensi dan waktu komputasi [44].

Convolutional Neural Network (CNN) adalah algoritma untuk pembelajaran mendalam yang berasal dari *Multilayer Percepton* (MLP) dan dirancang untuk memproses data dua dimensi, termasuk gambar dan suara. CNN digunakan untuk mengklasifikasikan data berlabel melalui pemanfaatan metode *supervised learning*. Ketika menggunakan *supervised learning*, data pelatihan dikombinasikan dengan variabel yang ditargetkan, yang kemudian dianalisis untuk mengklasifikasikan data [8].

2.3.5 Evaluation

Evaluasi adalah proses kuantifikasi hasil penilaian yang dilakukan pada model yang sebelumnya diimplementasikan di fase *modeling*. Hasil evaluasi menggambarkan prosedur penambangan data yang dilaksanakan dan menilai model optimal untuk implementasi [32]. Tujuan melakukan tahap evaluasi adalah untuk memastikan kinerja model akhir dan dapat dilakukan dengan melalui *confusion matrix*. Dalam *machine learning*, *confusion matrix* adalah metode perhitungan yang sering digunakan untuk mengkategorikan kinerja model klasifikasi. *Confusion matrix* digunakan untuk menilai keefektifan algoritma klasifikasi pada dataset tertentu. Pada Tabel 2.2 ditampilkan contoh *confusion matrix* hasil dari memprediksi dua kelas [36]:

Tabel 2.2 *Confusion Matrix*

		Kelas sebenarnya	
		1	2
Kelas prediksi	1	True positive	False negative
	2	False positive	True negative

Keterangan:

- TP (*true positive*): contoh data bernilai positif yang diprediksi benar sebagai positif
- TN (*true negative*): contoh data bernilai negatif yang diprediksi benar sebagai negatif
- FP (*false positive*): contoh data bernilai negatif yang diprediksi salah sebagai positif
- FN (*false negative*): contoh data bernilai positif yang diprediksi salah sebagai negatif.

Pada prakteknya *confusion matrix* digunakan untuk menghitung nilai akurasi, presisi, *recall* dan *f1- score* dengan rumus sebagai berikut [36]:

- Nilai akurasi, menggambarkan seberapa akurat model dalam mengklasifikasikan dengan benar.

$$Akurasi = \frac{TP + TN}{TP + FN + FP + TN} \times 100\% \quad (2.8)$$

- Presisi, menggambarkan akurasi antara data yang diminta dengan hasil prediksi yang diberikan oleh model.

$$Presisi = \frac{TP}{TP + FP} \times 100\% \quad (2.9)$$

- *Recall* atau *sensitivity*, menggambarkan keberhasilan model dalam menemukan kembali sebuah informasi.

$$Recall = \frac{TP}{TP + FN} \times 100\% \quad (2.10)$$

- *F1-Score*, merupakan perbandingan rata-rata presisi dan *recall* yang dibobotkan.

$$F1 - Score = 2 \times \frac{Precision \times Recall}{Precsion + Recall} \times 100\% \quad (2.11)$$

2.4 *Tools*

2.4.1 X

X atau yang sebelumnya bernama Twitter di bawah kepemilikan dan perusahaan Twitter Inc., sebuah *platform* yang memberi pengguna kemampuan untuk mengirim dan membaca pesan *tweet* melalui *microblog* yang merupakan jejaring sosial. *Microblog* diklasifikasikan sebagai bentuk alat komunikasi *online* di mana pengguna dapat memberikan pembaruan status mengenai pikiran dan tindakannya, serta pendapat seseorang tentang suatu objek atau fenomena tertentu [62].

Tweet adalah teks maksimum 140 karakter yang ditampilkan di halaman profil pengguna. Meskipun *tweet* dapat diakses oleh publik, pengguna memiliki opsi untuk membatasi pengiriman pesan ke kontak terverifikasi atau khusus pengikut. *Tweet* dapat ditampilkan di beranda kenalan yang mengikuti pengguna (follower) melalui profilnya. Pengguna dapat melihat *tweet* dari akun rekan-rekan yang mereka ikuti (following), selain milik mereka sendiri. X menjadi salah satu *platform* yang kerap ditemukan konteks berunsur ujaran kebencian atau bahasa yang menyinggung. Halaman X berisikan beragam topik dan jenis postingannya, seperti berita, kehidupan sehari-hari sampai kepada topik politik pemerintahan [1]. Pengguna X juga dapat melakukan pencarian kata kunci untuk menemukan *tweet* tentang topik tertentu, misalnya memasukkan kata "Jokowi" akan mengembalikan daftar beberapa *posting* pengguna yang berisi kata "Jokowi".

2.4.2 Python

Python dikembangkan oleh Guido van Rossum dan diluncurkan pada tahun 1991, adalah bahasa pemrograman yang kuat dan ramah pengguna. Python memiliki beberapa keunggulan, termasuk keterbacaan, efisiensi, multifungsi, interoperabilitas, dan dukungan komunitas yang

substansial. *Source code* Python untuk keterbacaan sangat mudah, membuatnya mudah untuk menulis, mengingat, dan menggunakan kembali. *Library* komprehensif Python membuatnya lebih efisien dibandingkan dengan bahasa pemrograman lain seperti Java, C, C #, dan C ++. Hal ini menghasilkan kode Python yang lebih sederhana dan termasuk bahasa pemrograman serbaguna yang dapat digunakan untuk mengembangkan situs web, aplikasi jaringan, aplikasi robotika, dan aplikasi kecerdasan buatan [45].

Python menawarkan berbagai modul pra-bangun yang dapat dengan mudah digunakan untuk pengembangan aplikasi sesuai kebutuhan. Python menunjukkan tingkat interoperabilitas yang tinggi karena kemampuannya untuk berinteraksi secara mulus dengan berbagai bahasa pemrograman. Program Python dapat dijalankan di berbagai sistem operasi seperti Windows, Linux, Mac OS, Unix, serta sistem operasi berbasis seluler seperti Android atau iOS. Python memiliki dukungan komunitas yang kuat karena sifatnya yang *open-source*. Komunitas yang kuat memfasilitasi kolaborasi pengguna yang mulus dan meningkatkan keandalan bahasa pemrograman Python [45].

2.4.3 Visual Studio Code

Microsoft mengembangkan editor teks ringan dan ampuh bernama Visual Studio Code (VS Code), yang tersedia dalam versi untuk Linux, Mac, dan Windows. Dengan bantuan *plugin* yang tersedia untuk diunduh melalui *marketplace* Visual Studio Code, editor teks ini secara langsung mendukung Javascript, Typescript, dan Node. Js selain bahasa pemrograman lain termasuk C++, C#, Python, Go, Java, PHP [46]. Membedakan warna berdasarkan fungsi dalam kumpulan kode, Visual Studio Code dapat mendeteksi bahasa pemrograman yang digunakan dan sudah terintegrasi dengan Gihub.

Fungsi lainnya adalah sistem ekstensi, yang memungkinkan pengembang untuk menyertakan kemampuan yang tidak tersedia di Visual Studio Code, Di antara berbagai fitur yang ditawarkan oleh Visual Studio Code adalah ekstensi yang meningkatkan kemampuan editor teks, integrasi Git, *Intellisense*, dan *Debugging*. Versi Visual Studio Code akan membawa lebih banyak fungsionalitas. Visual Studio Code berbeda dari editor teks lain dan pembaruan juga rutin dilakukan [47].

2.4.4 *Tweet-harvest*

Tweet-harvest, sebuah *command line tools* dengan memanfaat *Playwright* dalam mengumpulkan data *tweet* pencarian di X berdasarkan *keyword* pada suatu rentan waktu. Dikarenakan tidak menggunakan API resmi X, *Tweet-harvest* dapat menjadi pilihan ketika API X tidak tersedia, atau ketika fungsi API tidak memenuhi kebutuhan pengguna. Namun, penting untuk memperhatikan bahwa penggunaan *web scraping* untuk mengakses data dari situs *web*, termasuk X, harus dilakukan dengan memperhatikan kebijakan penggunaan yang berlaku dan etika penggunaan yang baik. *Tweet-harvest* memiliki kelebihan, yaitu memungkinkan pengguna untuk mengumpulkan data tambahan, berjalan sebagai *Command Line Interface* (CLI) hanya dengan menggunakan satu *auth_token* [48].