PROPOSAL PENELITIAN

KUALIFIKASI

PENGEMBANGAN DAN IMPLEMENTASI METODE *HYBRID DEEP LEARNING* MENGGUNAKAN CNN+LSTM DENGAN *FASTTEXT WORD EMBEDDING*

Antonius Angga Kurniawan
99219025

PROGRAM DOKTOR TEKNOLOGI INFORMASI

UNIVERSITAS GUNADARMA

2021

DAFTAR ISI

BAB I
PENDAHULUAN

1.1. Latar Belakang

Perkembangan internet menyebabkan banyak pengguna internet semakin mudah dalam mengkonsumsi informasi. Kemp (2021) dalam artikelnya di Data Reportal menyebutkan bahwa pada Januari 2021, jumlah pengguna internet di Indonesia mencapai 202,6 juta. Jumlah penduduk Indonesia adalah 274,9 juta dan tingkat penetrasi Internet adalah 73,7% (Kemp, 2021).

Berdasarkan data penggunaan internet di Indonesia, secara tidak langsung menunjukkan besarnya potensi konsumen informasi, termasuk konsumsi berita palsu. Berita palsu atau hoaks merupakan serangkaian informasi yang sesungguhnya tidak benar, tetapi sengaja dibuat seolah-olah benar adanya. Politik, ketertiban umum, bisnis, ilmu pengetahuan, kesehatan, bencana alam, dan sosial adalah beberapa bidang yang banyak digunakan untuk menyebarkan berita palsu (*Hasil Survey Wabah HOAX Nasional 2019 | Website Masyarakat Telematika Indonesia*, 2019). Dampak negatif yang sering ditimbulkan akibat adanya hoaks diantaranya adalah buang-buang waktu, pengalihan isu, penipuan publik, dan pemicu kepanikan publik (Berghel, 2017).

Menurut survey yang dilakukan oleh MasTel Indonesia pada tahun 2019, hanya 16.20% responden yang dapat langsung membedakan berita palsu, sedangkan masyarakat yang menerima berita hoaks setiap hari mencapai 34,60% (*Hasil Survey Wabah HOAX Nasional 2019 | Website Masyarakat Telematika Indonesia*, 2019). Masyarakat yang memiliki tingkat literasi yang rendah akan rentan terpapar berita hoaks. Hal itu ditandai dengan sikap masyarakat yang tidak mempertimbangkan dan memeriksa kebeneran suatu berita. Masyarakat dengan mudahnya menyebarkan suatu berita tanpa mengetahui sumber yang sebenarnya (Maulana, 2017; Witro, 2020).

Oleh karena itu perlu adanya gerakan sosial dan suatu cara untuk memeriksa kebenaran fakta dari suatu berita. Salah satu gerakan sosial yang muncul untuk melawan hoaks adalah MAFINDO (Masyarakat Anti Hoax Indonesia). MAFINDO

mendirikan sebuah situs yang bernama turnbackhoax.id, di mana pada situs tersebut terdapat banyak kumpulan-kumpulan berita palsu yang tersedia setiap bulan. Namun, metode identifikasi atau klasifikasi yang dilakukan oleh MAFINDO di dalam situs turnbackhoax.id masih menggunakan proses manual (Panjaitan & Santoso, 2021). Saat MAFINDO menerima berita, MAFINDO akan melakukan verifikasi terhadap berita tersebut dari berbagai sumber untuk mengidentifikasi apakah berita tersebut asli atau palsu, setelah terverifikasi maka berita tersebut dimasukkan ke dalam situs turnbackhoax.id dengan memberikan penjelasan di dalam berita tersebut.

Berdasarkan uraian di atas dapat disimpulkan bahwa dibutuhkan suatu metode khusus untuk mengidentifikasi suatu berita palsu dengan lebih baik dan lebih cepat agar para pengguna internet dapat membedakan berita yang palsu dan yang asli pada saat mengkonsumsi sebuah informasi.

Berita atau informasi mengandung kumpulan teks yang banyak, sehingga dalam proses pengolahannya tidak mudah dilakukan secara manual. Salah satu cara yang banyak digunakan untuk pengolahan teks adalah teknik *deep learning*. *Deep learning* merupakan sub-bidang dari *machine learning* di mana algoritma yang digunakan terinspirasi dari struktur otak manusia yang terdiri dari jaringan saraf. *Deep learning* mampu belajar dan beradaptasi terhadap sejumlah besar data serta menyelesaikan berbagai permasalahan yang sulit diselesaikan dengan algoritma *machine learning* lainnya (Setiawan, 2021).

Khasanah (2021) melakukan penelitian dengan menyelidiki bagaimana *embedding FastText* mempengaruhi kinerja model klasifikasi sentimen. Peneliti mengusulkan dua model klasifikasi sentiment dengan arsitektur sederhana. Model pertama adalah model Bidirectional Gated Recurrent Unit (BiGRU) satu layer dengan *embedding FastText*, dan yang kedua adalah model Convolutional Neural Network (CNN) satu layer dengan *embedding FastText*. Akurasi terbaik dihasilkan oleh model fastText + CNN, dengan akurasi 80% untuk dataset MR dan 84% akurasi untuk dataset SST2. Hal ini dikarenakan CNN dapat melatih model lebih cepat daripada kedua metode lainnya karena CNN secara komputasi lebih efisien. Hasil penelitian juga menunjukkan bahwa penggunaan CNN untuk klasifikasi

sentimen dapat memberikan hasil yang kompetitif dibandingkan dengan model BiLSTM dan BiGRU. Berdasarkan hasil yang didapat juga menunjukkan bahwa penggunaan *embedding FastText* dapat meningkatkan kinerja model BiLSTM, BiGRU, dan CNN dibandingkan dengan *embedding Glove* dengan *single-layer*. Sehingga dapat disimpulkan bahwa peneliti dapat menghasilkan model dengan arsitektur sederhana untuk masalah klasifikasi sentimen tetapi tetap memberikan kinerja yang kompetitif.

Peneliti Fesseha et al., (2021) mempelajari CNN untuk digunakan pada bahasa Tigrinya yang merupakan keluarga bahasa Semit dengan *resource* yang rendah dan bahasa yang kompleks. Peneliti mengeksplorasi CNN dengan dua model *word embedding*, yaitu *word2vec* dan *festText* yang digunakan memprediksi sebuah berita masuk ke dalam salah satu dari enam kategori berita. Peneliti membangun CNN dengan metode *continuous bag-of-word* (CBOW), CNN dengan metode *skip-gram* menggunakan *word2vec* dan juga *fasText*. Selain itu, peneliti juga membangun CNN tanpa *word2vec* dan *fastText.* Evaluasi yang digunakan adalah akurasi, *precision, recall* dan *f1-score*. Peneliti menyatakan teknik *word2vec* dan *FastText* adalah salah satu teknik *word embedding* terbaik di bidang penelitian NLP. Hal ini ditunjukkan dengan hasil akurasi pada CNN + CBOW dengan *word2vec* adalah 93.41% dan CNN + CBOW dengan *FastText* adalah 90.41%, di mana dengan adanya *word embedding* dari *word2vec* dan *fastText* secara signifikan meningkatkan akurasi untuk klasifikasi berita Tigrinya.

Nasir et al., (2021) melakukan penelitian dengan mengusulkan model hybrid dari *deep learning* yaitu CNN-RNN untuk mendeteksi berita palsu. Metode yang dilakukan adalah pengumpulan data, *word embedding*, melakukan tes hanya dengan CNN, melakukan tes hanya dengan RNN dan kemudian membuat model hybrid CNN-RNN. Hasilnya adalah model hybrid CNN-RNN berhasil divalidasi pada dua kumpulan data berita palsu (ISOT dan FA-KES), di mana hasil deteksi yang di dapat lebih baik dari metode dasar non-hybrid lainnya. Hasil akurasi pada dataset FA-KES adalah 0.60 dan hasil akurasi pada dataset ISOT adalah 0.99.

Kurniawan & Mustikasari (2021) melakukan penelitian dengan membandingkan dua model dari *deep learning* yaitu CNN dan LSTM untuk

menentukan berita palsu dalam bahasa Indonesia. Metode yang digunakan adalah pengumpulan data, pelabelan data, *preprocessing data*, *word2vec word embedding*, *splitting data*, melakukan pembuatan model CNN dan LSTM, kemudian menguji model dengan data baru yang belum pernah dilakukan *training*. Hasil yang didapatkan adalah tingkat akurasi dari CNN sebesar 0.88 dan LSTM sebesar 0.84.

Hermanto et al., (2021) melakukan penelitian yang bertujuan untuk melakukan pengklasifikasian judul berita berbahasa Indonesia berdasarkan sentiment positif, negatif dengan menggunakan metode LSTM, LSTM-CNN, CNN-LSTM. Data yang diambil adalah data judul artikel berbahasa Indonesia yang diambil dari situs Detik Finance. Metode yang dilakukan adalah *preprocessing*, *word2vec embedding*, pembuatan model LSTM, LSTM-CNN, CNN-LSTM, evaluasi model dengan akurasi, presisi dan *recall*. Berdasarkan hasil pengujian memperlihatkan bahwa metode LSTM, LSTM-CNN, CNN-LSTM memiliki hasl akurasi sebesar 62%, 65%, dan 74%. Jumlah dataset yang digunakan hanya 1200 dengan data training sebesar 900 dan data testing sebesar 300. Kemudian *Word2vec embedding* yang digunakan belum bisa meningkatkan hasil akurasi dari ketiga model yang dibuat karena rata-rata hasil akurasinya masih sekitar 67% dari ketiga model tersebut.

Mojumder et al., (2020) melakukan penelitian untuk *document classification* dalam bahasa Bangla. Peneliti berusaha untuk mengetahui dampak penggunaan dari *word embedding* khususnya dengan *fastText* terhadap kinerja dari tiga teknik *deep learning*, yaitu CNN, BiLSTM dan CBi-LSTM. Dalam modul klasifikasi, telah dilakukan upaya untuk mengklasifikasikan 40 ribu sampel berita ke dalam 12 kategori. Hasil dari penelitian menunjukkan kinerja yang signifikan dalam klasifikasi Bangla menggunakan *fastText embedding* tanpa *preprocessing* seperti *lemmatization, stemming*, dan lain-lain. Dari ketiga teknik *deep learning* yang digunakan bersama dengan *fastText* menunjukkan bahwa teknik BiLSTM adalah teknik yang paling menjanjikan untuk tugas ini. Teknik *fastText* + BiLSTM mendapatkan akurasi *testing* sebesar 85.5%, sedangkan *fastText* + CBiLSTM sebesar 84.3%, dan *fastText* + CNN sebesar 80.2%.

Menurut Nurdin et al., (2020) karakteristik teks yang tidak terstruktur menjadi tantangan dalam ekstraksi fitur pada bidang pemrosesan teks. Dalam penelitiannya Nurdin et al., (2020) bertujuan untuk membandingkan kinerja dari *word embedding Word2Vec*, *Glove*, dan *FastText* dan melakukan klasifikasi dengan algoritma CNN. Peneliti memilih ketiga *word embedding* tersebut karena dapat menangkap makna semantik, sintatik, dan urutan bahkan konteks di sekitar kata jika dibandingkan dengan *feature engineering* tradisional seperti *Bag of Words.* Proses *word embedding* dari metode tersebut akan dibandingkan kinerjanya untuk klasifikasi berita dari dataset 20 newsgroup dan Routers Newswire. Performa terbaik menunjukkan FastText unggul dibanding dua metode *word embedding* lainnya dengan nilai F-Measure sebesar 0.979 untuk dataset 20 Newsgroup dan 0.715 untuk Routers newswire. Namun, perbedaan kinerja yang tidak begitu signifikan antar ketiga *word embedding* tersebut menunjukkan bahwa ketiga *word embedding* tersebut memiliki kinerja yang kompetitif. Penggunaannya sangat bergantung pada dataset yang digunakan dan permasalahan yang ingin diselesaikan.

Berdasarkan uraian di atas dapat disimpulkan bahwa penambahan teknik *word embedding* pada model *deep learning* dapat memberikan tingkat akurasi yang lebih baik khususnya menggunakan *FastText word embedding*. Beberapa model *deep learning* seperti CNN dan LSTM juga cocok digunakan untuk melakukan identifikasi atau klasifikasi teks. Oleh karena itu, teknik *FastText word embedding*, model CNN dan LSTM masih dapat dikembangkan, salah satunya dengan memanfaatkan kumpulan berita palsu dengan jumlah besar pada situs turnbackhoax.id. Penelitian ini mengusulkan pengembangan model *deep learning* dengan cara hybrid antara CNN dan LSTM serta penggunaan teknik *word embedding* seperti *FastText* yang memiliki hasil cukup baik pada penelitian sebelumnya. Pada penelitian ini juga mengusulkan perancangan dan implementasi situs web yang dibangun menggunakan model yang sudah dikembangkan tersebut sehingga para pengguna internet dapat mengidentifikasi berita atau informasi yang di dapat merupakan berita palsu atau fakta secara langsung.

1.2. Rumusan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan rumusan masalah sebagai berikut:

1. Bagaimana membuat dataset corpus berita palsu dan berita fakta dalam bahasa Indonesia dari situs turnbackhoax.id?
2. Bagaimana mengembangkan model *deep learning* dengan cara hybrid antara CNN dan LSTM serta penggunaan teknik *FastText word embedding*?
3. Bagaimana cara mengimplementasikan pengembangan model yang sudah dibuat ke dalam situs web untuk mengidentifikasi berita palsu dan berita fakta secara langsung?

1.3. Batasan Masalah

Berdasarkan latar belakang yang telah diuraikan, maka dapat dirumuskan batasan masalah sebagai berikut :

1. Data yang digunakan menggunakan teks berita dalam bahasa Indonesia.
2. Model *deep learning* yang digunakan untuk hybrid adalah CNN dan LSTM.
3. Teknik *word embedding* yang digunakan adalah *FastText word embedding*.
4. Model yang sudah dibuat diimplementasikan ke dalam situs web yang digunakan untuk mengidentifikasi berita palsu dan berita fakta secara langsung.

1.4. Tujuan Penelitian

Sesuai dengan masalah penelitian yang telah diuraikan sebelumnya, maka tujuan yang ingin dicapai dalam penelitian ini adalah :

1. Menghasilkan dataset corpus berita palsu dan berita fakta dalam bahasa Indonesia.
2. Menghasilkan pengembangan model *deep learning* dengan cara hybrid pada CNN dan LSTM serta penggunaan teknik *word embedding* menggunakan *FastText word embedding*.
3. Mengimplementasikan hasil dari rancangan pengembangan model yang sudah dibuat ke dalam situs web yang digunakan untuk mengidentifikasi berita palsu dan berita fakta secara langsung.

1.5. Manfaat dan Kontribusi Penelitian

Dari segi keilmuan, penelitian ini memberikan kontribusi berupa *dataset corpus* berita palsu dan berita fakta dalam bentuk teks berbahasa Indonesia. Selain itu, usulan pengembangan model yang dilakukan diusahakan untuk menemukan cara baru atau penambahan atau modifikasi dari model *deep learning* dan teknik *word embedding* yang diusulkan agar didapatkan hasil yang optimal dalam mengidentifikasi atau mengklasifikasi sebuah teks khususnya pada berita palsu. Dari sisi teknologi, penelitian ini menghasilkan suatu rancangan model *deep learning* yang diimplementasikan ke dalam situs web sehingga dapat membantu dan memudahkan pengguna dalam mengidentifikasi suatu berita atau informasi yang palsu dan yang asli dalam bentuk teks.

BAB II
TINJAUAN PUSTAKA

Bab ini menguraikan tentang studi literatur terkait dengan identifikasi atau klasifikasi teks dan perkembangan penelitian mengenai proses klasifikasi khususnya yang membahas penelitian-penelitian tentang identifikasi atau klasifikasi dengan menggunakan metode *deep learning* serta *word embedding* khususnya model CNN, LSTM dan *FastText word embedding*.

2.1. Berita Palsu (Hoaks)

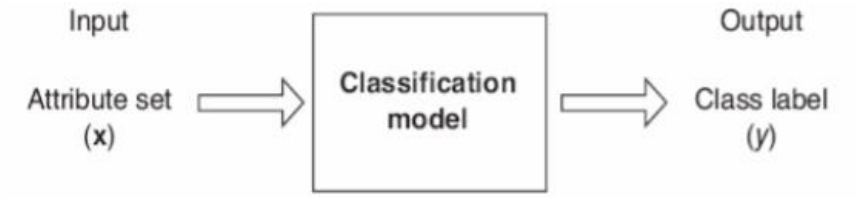
Dalam bahasa Inggris hoaks berarti menipu, berita bohong, tipuan, kabar burung, atau berita palsu (Sutantohadi, 2018). Menurut Firmansyah (2017) dalam mengartikan berita bohong (hoaks) merupakan kesengajaan dalam membuat suatu berita dengan tujuan memperdaya pembaca. Siswoko (2017) menyatakan bahwa hoaks dikenal juga dengan istilah berita palsu (*fake news*). Rahadi (2017) menyebutkan bahwa fake news bertujuan untuk memalsukan suatu informasi dan berupaya untuk menggantikan berita yang benar. Dari beberapa pengertian hoaks yang dikemukakan di atas, dapat disimpulkan bahwa hoaks adalah berita bohong yang dengan sengaja disebarkan dengan tujuan menggiring opini dan kemudian membentuk persepsi terhadap suatu informasi.

2.2. Klasifikasi

Ramageri (2010) mendeskripsikan klasifikasi menggunakan serangkaian contoh pra-klasifikasi untuk mengembangkan model yang dapat mengklasifikasikan record dalam populasi besar. Klasifikasi juga merupakan penempatan objek-objek ke salah satu dari beberapa kategori yang telah ditetapkan sebelumnya. Klasifikasi telah banyak ditemui dalam berbagai aplikasi. Sebagai contoh, pendeteksian pesan email spam berdasarkan header dan isi. Proses klasifikasi data melibatkan Learning dan Classification. Dalam Learning, data *training* dianalisis dengan algoritma klasifikasi. Dalam Classification data

digunakan untuk memperkirakan ketepatan aturan klasifikasi. Jika akurasi dapat diterima, aturan dapat diterapkan ke data *record* baru.

Data input untuk klasifikasi adalah koleksi dari *record*. Setiap *record* dikenal sebagai *instance* atau contoh, yang ditentukan oleh sebuah *tuple* (x,y), di mana x adalah himpunan atribut dan y adalah atribut tertentu, yang dinyatakan sebagai label kelas (juga dikenal sebagai kategori atau atribut target). Klasifikasi adalah tugas pembelajaran sebuah fungsi target f yang memetakan setiap himpunan atribut x ke salah satu label kelas y yang telah didefinisikan sebelumnya. Fungsi target juga dikenal secara informal sebagai model klasifikasi. Model klasifikasi dapat digunakan untuk memprediksi label kelas dari *record* yang tidak diketahui.



Gambar 2.1 Klasifikasi sebagai pemetaan sebuah himpunan atribut input x ke dalam label kelasnya y Ramageri (2010).

Seperti yang ditunjukkan pada gambar 2.1, sebuah model klasifikasi dapat dipandang sebagai kotak hitam yang secara otomatis memberikan sebuah label kelas Ketika dipresentasikan dengan himpunan atribut dari *record* yang tidak diketahui.

2.3 Pemrosesan Teks

Data dalam bentuk teks biasanya didapatkan dalam bentuk yang tidak terstruktur. Dalam banyak kasus, teks tersebut mengandung elemen-elemen yang dapat mengkontaminasi, seperti tag Hypertext Markup Language (HTML), simbol, kesalahan eja, dan masih banyak lagi. Hal ini dapat diperbaiki dengan pemrosesan teks yang baik. Tahapan-tahapan pemrosesan teks adalah sebagai berikut.

2.3.1 *Case Folding*

Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu, peran Case Folding dibutuhkan dalam mengkonversi keseluruhan teks

dalam dokumen menjadi suatu bentuk standar (biasanya huruf kecil atau lowercase). Sebagai contoh, user yang ingin mendapatkan informasi "KOMPUTER" dan mengetik "Komputer", "KomPUter", atau "komputer", tetap diberikan hasil retrieval yang sama yakni "komputer". Case folding adalah mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf 'a' sampai dengan 'z' yang diterima. Karakter selain huruf seperti angka, tanda baca, dan uniform resource locator (url), serta karakter kosong (*whitespace*) dihilangkan dan dianggap delimiter (Indraloka & Santosa, 2017).

2.3.2 *Tokenization*

Tahap *tokenization* adalah tahap pemotongan string input berdasarkan tiap kata yang menyusunnya. Hasil dari proses ini adalah token yang nantinya akan menjadi input untuk pemrosesan teks selanjutnya. Kata, angka, simbol, tanda baca dan entitas penting lainnya dapat dianggap sebagai token. Pada NLP token diartikan sebagai "kata" meskipun *tokenize* juga dapat dilakukan pada paragraf maupun kalimat (Aggarwal, 2018).

2.3.3 *Stopword Removal*

Stopword removal adalah proses mengambil kata-kata penting dari hasil token dengan menggunakan algoritma *stoplist* (membuang kata kurang penting) atau *wordlist* (menyimpan kata penting) (Nugroho, 2019). *Stopword* adalah kata umum yang biasanya muncul dalam jumlah besar dan dianggap tidak memiliki makna atau istilah-istilah yang tidak relevan. Contoh *stopword* dalam bahasa Indonesia adalah "yang", "dan", "di", "dari", dan masih banyak lagi. Makna di balik penggunaan *stopword* yaitu dengan menghapus kata-kata yang memiliki informasi rendah dari sebuah teks, kita dapat fokus pada kata-kata penting sebagai gantinya.

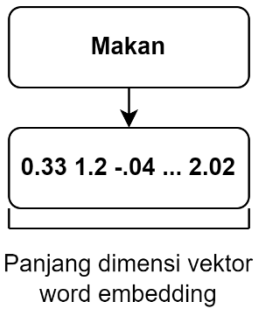
2.3.4 *Stemming*

Kata-kata yang muncul dalam teks yang digunakan sering kali memiliki banyak variasi morfologis. Oleh karena itu, setiap kata yang bukan *stopword* diubah menjadi kata dasarnya dengan menghilangkan awalan dan akhiran umum. Dengan

cara ini, kita dapat mengidentifikassi kumpulan kata di mana kata-kata tersebut bervariasi satu sama lain secara sintaks (Cios et al., 2007).

2.4 *Word Embedding*

Word embedding memetakan setiap kata dalam dokumen ke dalam dense vektor, di mana sebuah vektor merepresentasikan proyeksi kata di dalam ruang vektor. Posisi kata tersebut dipelajari dari teks atau berdasarkan kata-kata di sekitarnya. *Word embedding* ini dapat menangkap makna semantic dan sintaktik kata (Bengio et al., 2003; Nurdin et al., 2020). Mudahnya *word embedding* adalah istilah yang digunakan untuk teknik mengubah sebuah kata menjadi sebuah vektor atau array yang terdiri dari kumpulan angka.



Gambar 2.2 Contoh sederhana *word embedding*

Seperti pada gambar 2.2, dengan metode word embedding kita dapat mengubah kata menjadi sebuah vektor yang berisi angka-angka dengan ukuran yang cukup kecil untuk mengandung informasi yang lebih banyak.

2.4.1 *FastText*

FastText adalah metode *word embedding* yang merupakan pengembangan dari *word2vec*. Metode ini mempelajari representasi kata dengan mempertimbangkan informasi *subword*. Setiap kata direpresentasikan sebagai sekumpulan karakter *n-gram*. Dengan demikian dapat membantu menangkap arti kata-kata yang lebih pendek dan memungkinkan *embedding* untuk memahami sufiks dan prefiks dari kata. Representasi vektor dikaitkan dengan setiap karakter *n-gram*, sedangkan kata-kata direpresentasikan sebagai jumlah dari representasi

vektor tersebut. Setelah kata direpresentasikan dengan karakter *n-gram*, model Skip-gram dilatih untuk mempelajari *embedding* vektor dari kata (Bojanowski et al., 2017).

Pada umumnya model yang mempelajari representasi kata ke dalam vektor mengabaikan morfologi kata, setiap kata memiliki vektor yang berbeda. Hal ini menjadi keterbatasan untuk merepresentasikan kata dari bahasa dengan kosakata yang besar dan memiliki banyak katakata langka.

FastText memiliki kinerja yang baik, dapat melatih model pada dataset yang besar dengan cepat dan dapat memberikan representasi kata yang tidak muncul dalam data latih. Jika kata tidak muncul selama pelatihan model, kata tersebut dapat dipecah menjadi *n-gram* untuk mendapatkan *embedding* vektornya.

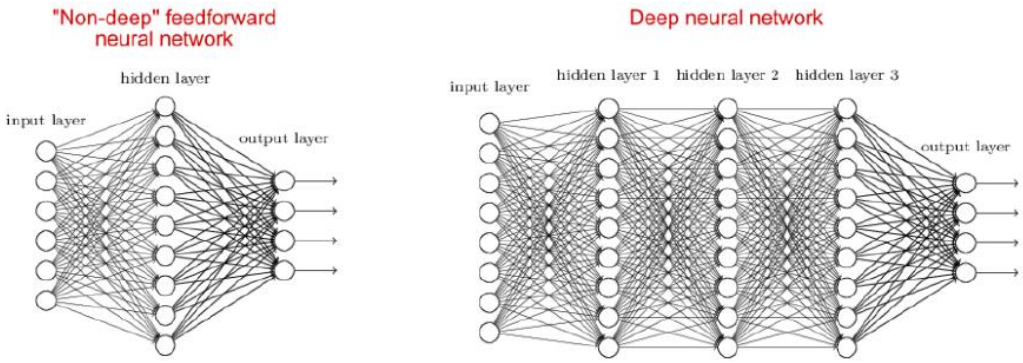
2.5 *Deep Learning*

Deep learning merupakan disiplin ilmu subset dari *machine learning*, tidak seperti teknik *machine learning* konvensional yang dibatasi oleh kemampuannya untuk memproses data mentah dan tergantung pada keahlian domain yang besar. Jika *machine learning* membutuhkan rekayasa yang cermat untuk merancang ekstrasi fitur, *deep learning* dapat mempelajari fitur dan memproses data secara langsung dalam bentuk mentah (LeCun, Y., Bengio, Y., Hinton, 2015).

Goodfellow et al., (2016) mendefinisikan *deep learning* sebagai berikut: "*Deep Learning* adalah jenis *machine learning* yang memiliki kekuatan besar dan fleksibilitas yang tinggi dengan belajar untuk mewakili dunia sebagai hierarki konsep bersarang, dengan masing-masing konsep didefinisikan dalam kaitannya dengan konsep yang lebih sederhana, dan lebih banyak representasi abstrak yang dihitung dalam hal yang kurang abstrak".

Sebagai contoh jika ada tugas untuk mengklasifikasikan gambar yang diberikan, jika itu mewakili kucing atau anjing, teknik *machine learning* konvensional harus mendefinisikan fitur wajah seperti telinga, mata, kumis, mulut dan sebagainya, maka perlu menulis metode untuk menentukan fitur mana yang lebih penting ketika mengklasifikasikan hewan tertentu, sedangkan *deep learning* tidak perlu menyediakan fitur secara manual, dengan *deep learning* fitur yang

paling penting akan diekstraksi secara otomatis, setelah menentukan fitur mana yang paling penting untuk mengklasifikasi foto (Zaccone & Karim, 2018).

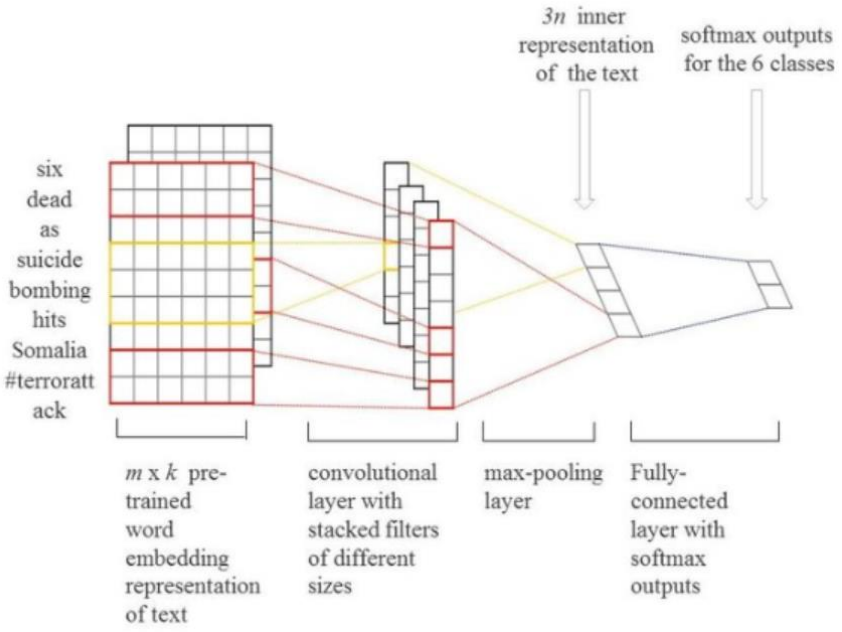


Gambar 2.3 Perbandingan standar *neural network* dan *deep neural network* (Arbones, 2017)

Implementasi dari *deep learning* biasanya menggunakan arsitektur *neural network* seperti yang ditunjukkan pada gambar 2.3, tetapi tidak seperti *neural network* tradisional yang biasanya hanya terdiri dari beberapa layer, *deep learning* biasanya terdiri dari ratusan bahkan ribuan layer untuk jaringan tersebut (Huang et al., 2016).

2.6 *Convolutional Neural Network* (CNN)

CNN adalah salah satu metode *machine learning* dari pengembangan Multi Layer Perceptron (MLP) yang didesain untuk mengolah data dua dimensi. CNN termasuk dalam jenis Deep Neural Network karena dalamnya tingkat jaringan dan banyak diimplementasikan untuk masalah klasifikasi (Goodfellow et al., 2016). CNN merupakan salah satu metode dari *deep learning* yang telah terbukti handal dalam sejumlah permasalahan klasifikasi khususnya di bidang *natural language processing (nlp)* karena kemampuannya secara efisien menangkap representasi bermakna dari kalimat (Nurdin et al., 2020). Gambar 2.4 menunjukan arsitektur sederhana dari CNN dalam melakukan klasifikasi teks.



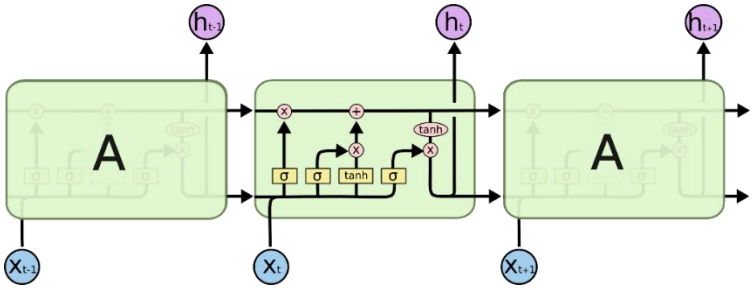
Gambar 2.4 CNN Arsitektur untuk klasifikasi teks

Layer pertama menyimpan kata-kata dalam sebuah *low-dimensional vector*. Layer selanjutnya menjalankan convolutions menggunakan *multiple filter sizes*. Selanjutnya, dilakukan *max-pool* hasil dari layer *convolutional* ke dalam sebuah *long feature vector*, menambahkan *dropout regularization*, dan mengklasifikasikan hasil menggunakan *softmax layer*.

2.7 *Long Short-Term Memory* (LSTM)

Jaringan LSTM (Colah, 2015) merupakan salah satu jenis jaringan RNN (*Recurrent Neural Network*) yang banyak digunakan untuk pembelajaran masalah prediksi data sekuensial. Sama seperti jaringan saraf lainnya, LSTM juga memiliki beberapa lapisan yang membantunya mempelajari dan mengenali pola untuk kinerja yang lebih baik. Operasi dasar LSTM dapat dianggap menyimpan informasi yang diperlukan dan membuang informasi yang tidak diperlukan atau berguna untuk prediksi lebih lanjut.

LSTM juga memiliki struktur berulang seperti RNN, namun LSTM memiliki struktur yang berbeda dalam melakukan pemrosesan. Biasanya RNN hanya memiliki 1 lapisan jaringan saraf, tetapi LSTM memiliki 4 lapisan dan berinteraksi dengan cara yang sangat istimewa.



Gambar 2.5 Modul berulang dalam LSTM berisi 4 lapisan yang saling berinteraksi (Colah, 2015)

Gambar 2.6 Notasi modul berulang LSTM (Colah, 2015)

Pada gambar 2.5 menunjukkan 4 lapisan yang dimaksud dan gambar 2.6 adalah keterangan dari notasi berulang LSTM, setiap garis membawa seluruh vektor, dari output satu simpul (*node*) ke input yang lain. Lingkaran merah muda mewakili operasi elemen, seperti penambahan atau perkalian elemen vektor, sedangkan kotak kuning adalah lapis jaringan saraf (mengandung parameter dan bias) yang bisa belajar. Dua garis yang bergabung menandakan penggabungan dua matriks atau vektor, sementara garis berpisah menandakan kontennya disalin dan salinannya pergi ke simpul yang berbeda (Colah, 2015).

2.8 *Evaluation Metrics*

Dalam klasifikasi data, *evaluation metrics* telah digunakan dalam dua tahap, yaitu tahap pelatihan (pembelajaran proses) dan tahap pengujian. Pada tahap pelatihan, *evaluation metrics* digunakan untuk mengoptimalkan algoritma klasifikasi. Dengan kata lain, *evaluation metrics* digunakan sebagai pembeda untuk mendiskriminasi dan memilih solusi optimal yang dapat menghasilkan prediksi yang lebih akurat dari evaluasi suatu pengklasifikasi tertentu. Sementara itu, pada tahap pengujian, *evaluation metrics* digunakan sebagai *evaluator* untuk mengukur efektivitas *classifier* yang dihasilkan ketika diuji dengan data yang tidak terlihat

(Hossin & Sulaiman, 2015). Terdapat beberapa jenis *evaluation metrics* yang sering digunakan untuk mengevaluasi suatu model, yaitu *accuracy* dan *confusion matrix*.

2.8.1 *Accuracy*

Akurasi adalah pengukuran seberapa sering model klasifikasi yang dibuat berhasil membuat prediksi yang benar. Akurasi merupakan rasio antara jumlah prediksi benar dan total jumlah prediksi seperti terlihat pada rumus (2.1).

$$Accuracy = \frac{Total\ Prediksi\ Benar}{Total\ Prediksi} \quad 2.1$$

2.8.2 *Confusion Matrix*

Confusion Matrix (atau *Confusion Table*) menunjukkan rincian yang lebih detil mengenai klasifikasi yang benar dan salah untuk setiap kelas. Baris matriks mewakili label benar, dan kolom mewakili prediksi. Misalkan set data tes berisi 100 contoh di kelas positif dan 200 contoh di kelas negatif; kemudian, *Confusion Matrix* mungkin terlihat seperti ini pada table 2.1 berikut.

Tabel 2.1 Contoh *Confusion Matrix*

	Terprediksi Positif	Terprediksi Negatif
Terlabel Positif	80	20
Terlabel Negatif	5	195

Sumber: (Zheng, 2015)

Berdasarkan Tabel 2.1, terlihat bahwa kelas positif memiliki akurasi yang lebih rendah (80 / (20 + 80) = 80%) daripada kelas negatif (195 / (5 + 195) = 97,5%). Sedangkan nilai akurasi jika dilihat berdasarkan tabel tersebut adalah (80 + 195) / (100 + 200) = 91,7% (Zheng, 2015).

2.9 Kajian Penelitian

Beberapa penelitian sebelumnya yang menjadi referensi penulis dalam melakukan penelitian ini, terutama penelitian tentang metode *deep learning* CNN dan LSTM dengan pemanfaatan *word embedding* khususnya *FastText* pada klasifikasi teks. Berikut kajian penelitian disajikan pada Tabel 2.1.

Tabel 2.1 Ringkasan Penelitian

Peneliti/Judul	Metode	Hasil	Keterbatasan
Isnaini Nurul Khasanah, 2021. "Sentiment Classification Using fastText Embedding and Deep Learning Model" (Khasanah, 2021)	BiDirectional Gated Recurrent Unit (BiGRU) satu layer + *FastText embedding*, CNN satu layer + *FastText embedding*, BiLSTM satu layer + *FastText embedding* menggunakan 2 dataset Movie Review (MR) dan Standford Sentiment Treebank (SST2)	Akurasi terbaik dihasilkan oleh *FastText* + CNN dengan akurasi 80% pada dataset MR dan 84% pada dataset SST2. Pada penelitian ini peneliti menyinpulkan bahwa penggunaan *FastText word embedding* dapat meningkatkan performa dari BiGRU, BiLSTM, dan CNN daripada *Glove embedding*.	Tahap *preprocessing* yang dilakukan hanya merubah semua teks menjadi huruf kecil, *remove stopword*, dan menghilangkan tanda baca. Tahap *tokenization* dan *stemming* tidak dilakukan. Nilai akurasi yang dihasilkan masih di angka sekitar 80% belum mencapai 90%.
Awet Fesseha, Shengwu Xiong, Eshete Derb Emiru, Moussa Diallo, Abdelghani Dahou, 2021. "Text Classification Based on Convolution Neural Networks and Word Embedding for Low-Resource Languages: Tigrinya" (Fesseha et al., 2021)	CNN + *word2vec embedding* dan CNN + *fastText embedding* menggunakan metode *continuous bag-of-word (CBOW)* dan *Skip-gram*. Selain itu peneliti juga membangun CNN tanpa *word2vec* dan *fastText*. Evaluasi yang digunakan adalah *accuracy, precision, recall* dan *f1-score*.	Peneliti menyatakan *word2vec* dan *fastText* adalah salah satu teknik *word embedding* terbaik di bidang penelitian NLP. Hasil akurasi pada CNN + CBOW dengan *word2vec* adalah 93.41% dan CNN+CBOW dengan *fastText* adalah 90.41%, di mana dengan adanya *word2vec* dan	Tahap *preprocessing* seperti *stemming* tidak dilakukan. Hanya menggunakan 1 metode uji coba yaitu CNN. Resources berita dengan bahasa Tigrinya rendah.

		fastText word embedding secara signifikan dapat meningkatkan akurasi untuk klasifikasi berita Tigrinya.	
Jamal Abdul Nasir, Osama Subhani Khan, Iraklis Varlamis, 2021. "Fake news detection: A hybrid CNN-RNN based deep learning approach" (Nasir et al., 2021)	Pengumpulan data, *preprocessing*, *Glove word embedding*, uji coba hanya dengan metode CNN, uji coba hanya dengan RNN dan uji coba dengan metode *hybrid* yang diusulkan yaitu CNN-RNN. Melakukan perbandingan dengan metode *machine learning* standar seperti *Logistic Regression*, *Random Forest*, KNN, *Decision Tree*.	Model *hybrid* CNN-RNN berhasil divalidasi pada dua kumpulan data berita palsu (ISOT dan FA-KES), di mana hasil deteksi yang didapat lebih baik dari metode *non-hybrid* lainnya. Hasil akurasi pada dataset FA-KES adalah 0.60 dan hasil akurasi pada dataset ISOT adalah 0.99.	Model *hybrid* CNN-RNN menggunakan *Glove embedding* memang cenderung bekerja baik namun hanya pada kumpulan data tertentu. Hal ini dibuktikan dengan perbandingan pada data Fa-Kes yang hanya mendapatkan akurasi 0.60 dibandingkan dengan dataset ISOT yang mendapatkan akurasi 0.99 untuk mengklasifikasikan berita palsu..
Antonius Angga Kurniawan, Metty Mustikasari, 2021. "Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita	Pengumpulan data, pelabelan data, *preprocessing data*, *word2vec embedding*, *splitting data*, membandingkan dua metode yaitu CNN dan LSTM,	Hasil dari *accuracy test* dengan metode CNN sebesar 0.88 dan metode LSTM sebesar 0.84.	Jumlah dataset yang digunakan tidak terlalu banyak, yaitu sebesar 1786 berita yang terdiri dari 802 berita fakta dan 984 berita palsu.

Palsu dalam Bahasa Indonesia" (Kurniawan & Mustikasari, 2021)	uji coba menggunakan data baru yang belum pernah dilatih dan dites.		Tidak melakukan *preprocessing stemming*. Model CNN dan LSTM menggunakan *word2vec word embedding* belum terlalu signifikan dari hasil akurasi karena masih di bawah 0.90.
Dedi Tri Hermanto, Arief Setyanto, Emha Taufiq Luthfi, 2021. "Algoritma LSTM-CNN untuk Sentiment Klasifikasi dengan Word2Vec pada media online" (Hermanto et al., 2021)	Pengumpulan data judul berita dari Finance.detik.com, *preprocessing*, *word2vec embedding*, pembuatan model LSTM, LSTM+CNN, CNN+LSTM, evaluasi model akurasi, presisi dan recall.	Berdasarkan pengujian didapatkan hasil akurasi dari LSTM sebesar 62%, LSTM+CNN sebesar 65%, dan CNN+LSTM sebesar 74%.	Jumlah dataset yang digunakan hanya 1200 dengan data training sebesar 900 dan data testing sebesar 300. *Word2vec embedding* yang digunakan belum bisa meningkatkan hasil akurasi dari ketiga model yang dibuat karena rata-rata hasil akurasinya masih sekitar 67% dari ketiga model.
Pritom Mojumder, Mahmudul Hasan, Md. Faruque Hossain, K.M. Azharul Hasan, 2020. "A Study of fastText Word Embedding Effects in Document	Menggunakan *FastText word embedding* pada 3 metode *deep learning* yaitu CNN, BiLSTM, dan Convolutional Bi-LSTM (CBiLSTM).	Hasil dari penelitian menunjukkan kinerja yang signifikan dalam klasifikasi berita Bangla menggunakan *FastText embedding* tanpa *preprocessing* seperti	Tidak melakukan teknik *preprocessing*. Hasil akurasi *testing* dari ketiga metode *deep learning* sudah bagus namun masih dibawah 90%.

Classification in Bangla Language" (Mojumder et al., 2020)	Sampel yang digunakan sebanyak 40 ribu sampel berita dengan 12 kategori berita.	*lemmatization, stemming,* dll. Hasil akurasi *testing* menggunakan *FastText*+BiLSTM sebesar 85.5%, *FastText*+CBiLSTM sebesar 84.3%, dan *FastText*+CNN sebesar 80.2%.	
Arliyanti Nurdin, Bernadus Anggo Seno Aji, Anugrayani Bustamin, Zaenal Abidin, 2020. "Perbandingan Kinerja Word Embedding Word2Vec, Glove, dan FastText pada Klasifikasi Teks" (Nurdin et al., 2020)	Membandingkan kinerja dari metode *word embedding* dari *Word2Vec, Glove, FastText* dan melakukan klasifikasi berita dari dataset 20 newsgroup dan Routers newswire dengan algortima CNN.	Hasil menunjukkan *FastText* memiliki performa terbaik dan lebih unggul dibandingkan dengan dua metode *word embedding* lainnya dengan nilai F-Measure sebesar 0.979 untuk dataset 20 Newsgroup dan 0.715 untuk dataset Routers Newswire.	Tidak dilakukan teknik *preprocessing*. Hasil akurasi dengan dataset Routers Newswire hanya sebesar 0.715.
Ernest Lim, Esther Irawati Setiawan, Joan Santoso, 2019. "Stance Classification Post Kesehatan di Media Sosial dengan FastText Embedding dan Deep Learning" (Lim et al., 2019)	Melakukan klasifikasi *stance* suatu judul *post* kesehatan di Facebook terhadap judul *post* lainnya. *Stance* dibagi menjadi 3 yaitu *for* (setuju), *observing* (netral), dan *against* (berlawanan). Metode yang digunakan adalah	Model dengan *FastText* pada penelitian ini mampu menghasilkan F1-*macro score* sebesar 64%. Peneliti menggunakan model *FastText* dikarenakan pada penelitian terdahulu teknik *word2vec* memiliki	Jumlah dataset yang digunakan hanya 3500. Hasil F1-*macro score* masih di bawah 70%.

	FastText word embedding dengan beberapa metode *deep learning* seperti CNN, LSTM dan BiLSTM.	keterbatasan pada pengenalan kata baru.	

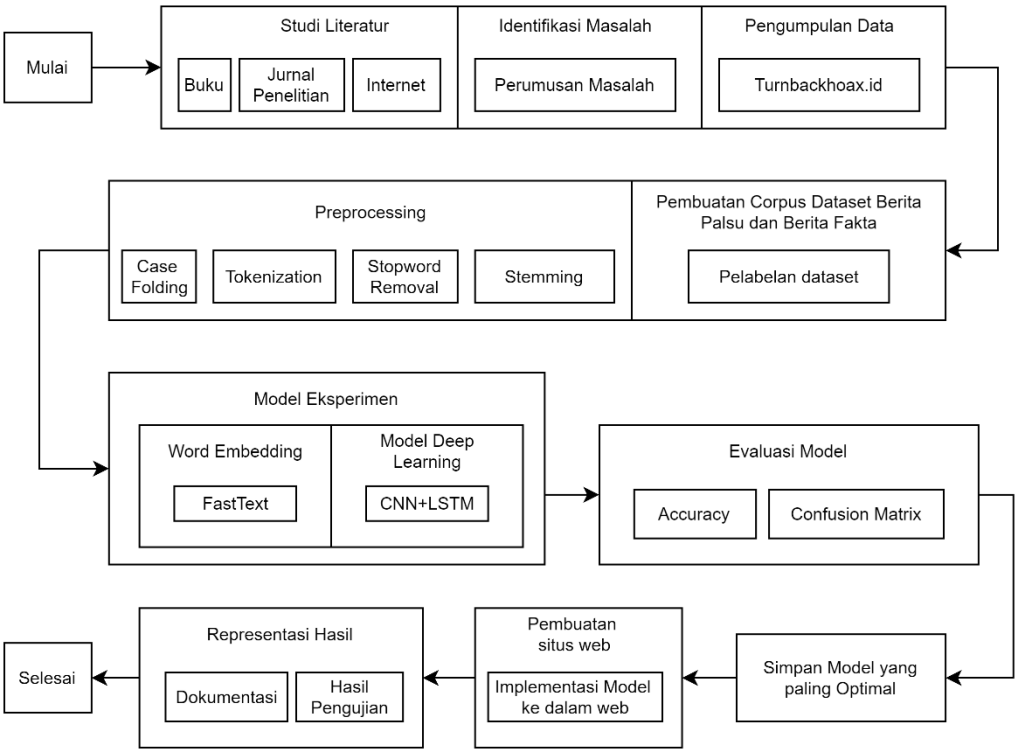
BAB III
METODOLOGI PENELITIAN

3.1. Tahapan Penelitian

Penelitian ini berusaha mengembangkan metode sebagai solusi dari masalah dan kekurangan dari teknik yang pernah dilakukan oleh peneliti terdahulu yang dapat memberikan optimasi akurasi dalam mengklasifikasikan teks dengan pendekatan baru. Metode pada penelitian ini mencoba untuk mengembangkan algoritma *deep learning* CNN dan LSTM disertai dengan teknik *FastText word embedding* agar mendapatkan nilai akurasi yang lebih baik dalam mengklasifikasikan teks khususnya berita palsu.

Pada penelitian ini mengusulkan pengembangan algoritma *deep learning* dengan cara *hybrid* antara metode CNN dan LSTM disertai dengan penggunaan *Fast Text word embedding* untuk klasifikasi teks pada berita palsu. Berdasarkan usulan tersebut, diharapkan metode klasifikasi teks dapat lebih efisien dan mengoptimalkan akurasi dalam mengklasifikasikan teks khususnya berita palsu. Selain itu, dalam penelitian ini mencoba untuk melakukan pembuatan *corpus* berita palsu dan berita fakta dalam bahasa Indonesia.

Metodologi yang digunakan pada penelitian ini terdiri dari beberapa tahapan yang selanjutnya dapat dilihat pada Gamar 3.1.



Gambar 3.1 Alur Penelitian

Penelitian ini tahap pertama dimulai dari tahapan studi literatur. Pada tahap ini peneliti melakukan studi literatur dengan membaca buku, jurnal penelitian dan pencarian artikel-artikel serta sumber-sumber lain di internet yang terkait dengan topik penelitian. Selanjutnya adalah mengidentifikasi masalah berdasarkan studi literatur yang sudah dilakukan dengan merumuskan masalah. Kemudian dilakukan pengumpulan data dan informasi berupa teks berita palsu dan teks berita fakta dari situs turnbackhoax.id. Setelah mendapatkan *raw* data sesuai dengan kebutuhan, tahap kedua adalah melakukan pembuatan *corpus dataset* untuk berita palsu dan berita fakta. Pada tahap ini dilakukan pelabelan data pada *dataset*. Berikutnya melakukan *preprocessing*, data dalam bentuk teks biasanya didapatkan dalam bentuk yang tidak terstruktur seperti adanya kesalahan eja, simbol, emotikon, *url* dan angka. Maka dari itu diperlukan teknik *preprocessing* untuk memperbaikinya. Tahap ketiga adalah model eksperimen, tahapan yang dilakukan adalah *word embedding* dengan teknik *FastText* yang berguna untuk merepresentasikan kata menjadi sebuah vektor. Pada tahap selanjutnya ini berisikan proses desain sistem dan *software* seperti pemodelan metode CNN+LSTM. Tahap keempat melakukan

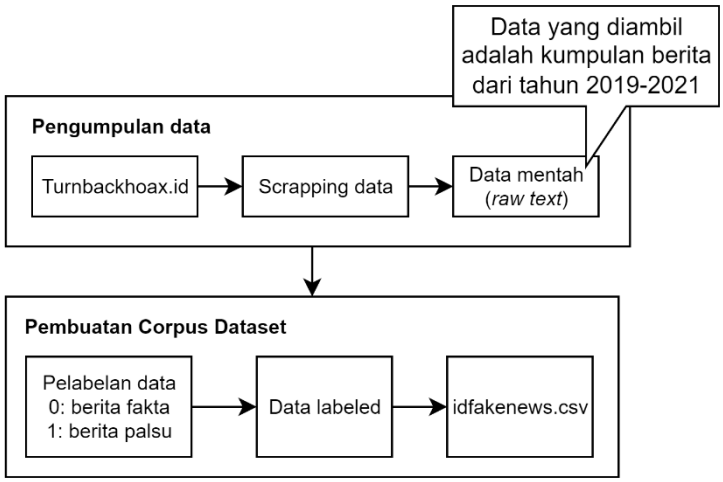
evaluasi model dengan dilakukan perhitungan, tingkat akurasi dan *confusion matrix* digunakan untuk mengukur kinerja yang memiliki beberapa parameter. Setelah model sudah dievaluasi dan didapatkan hasil yang paling optimal, pada tahap kelima dilakukan penyimpanan model agar model dapat diintegrasikan pada tahap selanjutnya. Tahap keenam adalah pembuatan situs web, model yang sudah disimpan diimplementasikan ke dalam web, sehingga web yang dibangun memiliki kemampuan dalam mengklasifikasikan teks khususnya berita palsu. Pada tahap akhir, tahap ketujuh dilakukan representasi hasil berupa hasil pengujian serta dokumentasi yang sudah dilakukan dalam penelitian.

3.2. Usulan Penelitian

Berdasarkan studi literatur yang sudah dilakukan, dalam penelitian ini diajukan tiga buah usulan yang terdiri dari pembuatan *corpus dataset*, model eksperimen, implementasi hasil model eksperimen ke dalam situs web.

3.2.1. Pembuatan Corpus Dataset

Dalam penelitian ini, peneliti mencoba untuk membuat sebuah *corpus dataset* berisikan berita palsu dan berita fakta dalam bahasa Indonesia yang dikumpulkan dari situs turnbackhoax.id. Gambar 3.2 menunjukkan tahap dalam pembuatan *corpus dataset*.



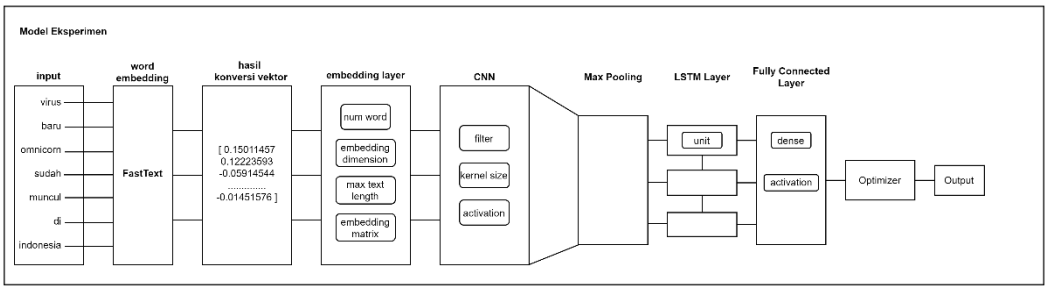
Gambar 3.2 Pembuatan *corpus dataset* berita palsu dalam bahasa Indonesia.

Pembuatan *corpus dataset* berita palsu dimulai dari pengumpulan data. Pengumpulan data dilakukan dengan mengumpulkan berita palsu dan berita fakta dari situs turnbackhoax.id. Proses *scrapping* data dilakukan untuk mengambil data mentah sesuai dengan kebutuhan, yaitu kumpulan teks berita yang ada di situs tersebut. Data mentah yang diambil untuk dijadikan dataset memiliki rentang waktu dari tahun 2019 sampai dengan tahun 2021. Pada situs turnbackhoax.id selalu memperbarui kumpulan berita yang disediakan sehingga situs tersebut memiliki kumpulan berita setiap bulannya.

Setelah proses pengumpulan data selesai tahap yang dilakukan berikutnya adalah pelabelan data. Pelabelan data dilakukan secara manual dengan memberikan label 0 untuk berita fakta dan label 1 untuk berita palsu. Kemudian data yang sudah dilabelkan disimpan ke dalam format .csv (idfakenews.csv) yang terdiri dari dua kolom, kolom pertama adalah label dan kolom yang kedua adalah teks.

3.2.2. Model Eksperimen

Dalam penelitian ini, peneliti mengusulkan pengembangan metode *deep learning* dengan cara *hybrid* CNN+LSTM disertai dengan teknik *word embedding* menggunakan *FastText word embedding*. Model eksperimen tersebut diusulkan karena berdasarkan penelitian terdahulu (Fesseha et al., 2021; Hermanto et al., 2021; Khasanah, 2021; Kurniawan & Mustikasari, 2021; Mojumder et al., 2020; Nasir et al., 2021; Nurdin et al., 2020) model CNN dan LSTM memiliki kemampuan yang baik terkait klasifikasi teks. Selain itu, teknik *word embedding* khususnya *FastText* juga memberikan dampak yang signifikan pada saat digunakan bersama dengan metode *deep learning* dalam hal klasifikasi teks . Gambar 3.3 menunjukkan arsitektur model eksperimen yang akan dibangun.



Gambar 3.3 Arsitektur model eksperimen

Arsitektur dari model eksperimen diawali dengan proses *input* yang berupa teks berita. Selanjutnya dilanjutkan ke dalam proses *word embedding* menggunakan *FastText word embedding*. *FastText word embedding* ini terdiri dari *pre-trained* model yang sudah pernah dilatih di mana berisikan kumpulan kata dalam bentuk vektor yang nantinya akan dicocokkan dengan *input* yang diberikan. Apabila tidak ada kata yang sesuai di dalamnya, maka *FastText* akan membuat sebuah vektor baru pada kata tersebut. Hasil dari proses *word embedding* berupa kumpulan vektor dari kata-kata yang sudah di*input*kan. Hasil konversi vektor yang didapat akan diteruskan ke dalam *embedding layer*. Pada *embedding layer* diatur agar jumlah kata, dimensi, panjang teks dan *embedding matrix* memiliki ukuran yang sama pada setiap *inputan* yang masuk. Model CNN+LSTM terdiri dari lapisan *convolutional* awal yang akan menerima *input embedding layer* untuk setiap token sebagai *input*. Dari lapisan *convolutional* akan dihasilkan sebuah *feature map* yang berisi fitur-fitur penting yang berdimensi lebih rendah di *hidden layer*. *Max Pooling Layer* akan mengambil nilai tertinggi dari elemen-elemen yang berada pada lingkup *window* satu dimensi dengan ukuran yang ditentukan di awal, sehingga diperoleh informasi paling penting dari *feature map* hasil konvolusi. Kemudian *output feature map* ini akan diumpankan ke dalam lapisan LSTM yang diharapkan akan mengekstrak fitur lokal. *Output* dari lapisan LSTM, berupa *feature map* yang telah di-*reshape* menjadi sebuah vektor, dihubungkan dengan *output layer* (*fully connected layer*) untuk diklasifikasikan. Pada layer ini, digunakan fungsi aktivasi dan *loss function*. Berikutnya digunakan sebuah *optimizer* untuk memberikan tingkat akurasi dan *lose* yang lebih baik pada sebuah model. Pada akhirnya *layer* terakhir akan didapatkan sebuah label berita palsu atau berita fakta.

3.2.3. Implementasi Model ke dalam Situs Web

Pada penelitian ini, model terbaik yang didapat akan diimplementasikan ke dalam situs web yang akan digunakan untuk mengklasifikasikan berita palsu dalam bentuk teks berbahasa Indonesia. Hal ini ditujukan agar para pengguna internet dapat dengan mudah mengidentifikasi berita atau informasi yang diterima

merupakan berita palsu atau bukan. Sehingga dengan adanya situs web yang dibangun diharapkan para pengkonsumsi informasi dapat lebih kritis dan lebih bijak terhadap berita yang diterima maupun yang akan disebarkan.

3.3. Rencana Kerja

Rencana penyelesaian waktu dalam penelitian ini selama 24 bulan, dengan rincian kerja yang terlihat pada tabel 3.1.

Tabel 3.1. Rencana Pelaksanaan Penelitian

Bulan	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24
Langkah Penelitian																								
Studi Literatur																								
Pengumpulan Data																								
Pelabelan *dataset* dan tahapan *preprocessing*																								
Splitting Data																								
Implementasi Model Eksperimen																								
Evaluasi Model dan Menyimpan Model																								
Implementasi model ke dalam situs web																								
Representasi Hasil																								

DAFTAR PUSTAKA

Aggarwal, C. C. (2018). *Machine Learning for Text*. New York: Springer Science & Business Media.

Arbones, M. (2017). *Deep learning: Creating bridges between dmps in auto encoders and recurrent neural networks*. Escola TÃšcnica Superior d'Enginyeria Industrial de Barcelona.

Bengio, Y., Réjean, D., Vincent, P., & Jauvin, C. (2003). A Neural Probabilistic Language Model. *Journal OfMachine Learning Research 3*, 19. https://doi.org/10.1080/1536383X.2018.1448388

Berghel, H. (2017). Alt-News and Post-Truths in the "Fake News" Era. *Computer*. https://doi.org/10.1109/MC.2017.104

Bojanowski, P., Grave, E., Joulin, A., & Mikolov, T. (2017). Enriching Word Vectors with Subword Information. *Transactions of the Association for Computational Linguistics*. https://doi.org/10.1162/tacl_a_00051

Cios, K. J., Pedrycz, W., Swiniarski, R. W., & Kurgan, L. A. (2007). Data mining: A knowledge discovery approach. In *Data Mining: A Knowledge Discovery Approach*. https://doi.org/10.1007/978-0-387-36795-8

Colah. (2015, August 27). *Understanding LSTM Networks -- colah's blog*. http://colah.github.io/posts/2015-08-Understanding-LSTMs/

Fesseha, A., Xiong, S., Emiru, E. D., Diallo, M., & Dahou, A. (2021). Text classification based on convolutional neural networks and word embedding for low-resource languages: Tigrinya. *Information (Switzerland)*, *12*(2), 1–17. https://doi.org/10.3390/info12020052

Firmansyah, R. (2017). Web Klarifikasi Berita untuk Meminimalisir Penyebaran Berita Hoax. *Jurnal Informatika*, *4*(2), 230–235.

Goodfellow, I., Bengio, Y., & Courville, A. (2016). Deep Learning. In *MIT Press*. MIT Press. https://www.deeplearningbook.org/

Hasil Survey Wabah HOAX Nasional 2019 | Website Masyarakat Telematika Indonesia. (2019). https://mastel.id/hasil-survey-wabah-hoax-nasional-2019/

Hermanto, D. T., Setyanto, A., & Luthfi, E. T. (2021). Algoritma LSTM-CNN

untuk Sentimen Klasifikasi dengan Word2vec pada Media Online. *Creative Information Technology Journal*, *8*(1), 64. https://doi.org/10.24076/citec.2021v8i1.264

Hossin, M., & Sulaiman, M. N. (2015). A Review on Evaluation Metrics for Data Classification Evaluations. *International Journal of Data Mining & Knowledge Management Process*. https://doi.org/10.5121/ijdkp.2015.5201

Huang, G., Sun, Y., Liu, Z., Sedra, D., & Weinberger, K. Q. (2016). Deep networks with stochastic depth. *Lecture Notes in Computer Science (Including Subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. https://doi.org/10.1007/978-3-319-46493-0_39

Indraloka, D. S., & Santosa, B. (2017). Penerapan Text Mining untuk Melakukan Clustering Data Tweet Shopee Indonesia. *Jurnal Sains Dan Seni ITS*. https://doi.org/10.12962/j23373520.v6i2.24419

Kemp, S. (2021). *Digital in Indonesia: All the Statistics You Need in 2021 — DataReportal – Global Digital Insights*. https://datareportal.com/reports/digital-2021-indonesia

Khasanah, I. N. (2021). Sentiment Classification Using fastText Embedding and Deep Learning Model. *Procedia CIRP*, *189*, 343–350. https://doi.org/10.1016/j.procs.2021.05.103

Kurniawan, A. A., & Mustikasari, M. (2021). Implementasi Deep Learning Menggunakan Metode CNN dan LSTM untuk Menentukan Berita Palsu dalam Bahasa Indonesia. *Jurnal Informatika Universitas Pamulang*, *5*(4), 544. https://doi.org/10.32493/informatika.v5i4.6760

LeCun, Y., Bengio, Y., Hinton, G. (2015). Deep learning. nature 521 (7553): 436. *Nature*, *521*, 436–444.

Lim, E., Setiawan, E. I., & Santoso, J. (2019). Stance Classification Post Kesehatan di Media Sosial Dengan FastText Embedding dan Deep Learning. *Journal of Intelligent System and Computation*, *1*(2), 65–73. https://doi.org/10.52985/insyst.v1i2.86

Maulana, L. (2017). Kitab Suci dan Hoax: Pandangan Alquran dalam Menyikapi Berita Bohong. *Wawasan: Jurnal Ilmiah Agama Dan Sosial Budaya*, *2*(2),

209–222. https://doi.org/10.15575/jw.v2i2.1678

Mojumder, P., Hasan, M., Hossain, M. F., & Hasan, K. M. A. (2020). A Study of fastText Word Embedding Effects in Document Classification in Bangla Language. In *Cyber Security and Cyber Science* (Issue 325, pp. 1–13). Springer Nature Switzerland AG 2020. https://doi.org/10.1007/978-3-642-03503-6

Nasir, J. A., Khan, O. S., & Varlamis, I. (2021). Fake news detection: A hybrid CNN-RNN based deep learning approach. *International Journal of Information Management Data Insights*, *1*(1), 100007. https://doi.org/10.1016/j.jjimei.2020.100007

Nugroho, K. S. (2019, January 18). *Dasar Text Preprocessing dengan Python | by Kuncahyo Setyo Nugroho | Medium*. https://medium.com/@ksnugroho/dasar-text-preprocessing-dengan-python-a4fa52608ffe

Nurdin, A., Anggo Seno Aji, B., Bustamin, A., & Abidin, Z. (2020). Perbandingan Kinerja Word Embedding Word2Vec, Glove, Dan Fasttext Pada Klasifikasi Teks. *Jurnal Tekno Kompak*, *14*(2), 74. https://doi.org/10.33365/jtk.v14i2.732

Panjaitan, A. T. B., & Santoso, I. (2021). Deteksi Hoaks Pada Berita Berbahasa Indonesia Seputar COVID-19. *Format : Jurnal Ilmiah Teknik Informatika*, *10*(1), 76. https://doi.org/10.22441/format.2021.v10.i1.007

Rahadi, D. R. (2017). Perilaku Pengguna Dan Informasi Hoax Di Media Sosial. *Jurnal Manajemen Dan Kewirausahaan*, *5*(1), 58–70. https://doi.org/10.26905/jmdk.v5i1.1342

Ramageri, B. M. (2010). Data Mining Techniques and Applications. *Indian Journal of Computer Science and Engineering*, *1*(4), 301–305.

Setiawan, R. (2021, October 9). *Mengenal Deep Learning Lebih Jelas - Dicoding Blog*. https://www.dicoding.com/blog/mengenal-deep-learning/

Siswoko, K. H. (2017). Kebijakan Pemerintah Menangkal Penyebaran Berita Palsu atau 'Hoax.' *Jurnal Muara Ilmu Sosial, Humaniora, Dan Seni*, *1*(1), 13. https://doi.org/10.24912/jmishumsen.v1i1.330

Sutantohadi, A. (2018). Bahaya Berita Hoax Dan Ujaran Kebencian Pada Media Sosial Terhadap Toleransi Bermasyarakat. *DIKEMAS (Jurnal Pengabdian*

Kepada Masyarakat), *1*(1).

Witro, D. (2020). URGENCY RIJALUL POSTING IN PREVENTING HOAX: QURANIC PERSPECTIVE. *Islamic Communication Journal*, *5*(1).

Zaccone, G., & Karim, M. R. (2018). *Deep Learning with TensorFlow: Explore neural networks and build intelligent systems with Python* (2nd ed.). Packt Publishing.

Zheng, A. (2015). Evaluating Machine Learning Models - O'Reilly Media. In *Oreilly*.