**BAB I**

**PENDAHULUAN**

* 1. **Latar Belakang Masalah**

Penyebaran berita *hoax* saat ini sedang marak terjadi, bahkan sudah masuk fase yang cukup mengkhawatirkan. Menteri Komunikasi dan Informatika (Menkominfo), Rudiantara mengatakan tidak kurang dari 900 ribu situs yang melakukan penyebaran berita *hoax* (Fajri, 2019). Penyebaran berita *hoax* ini bertujuan untuk menggiring opini para pembaca sehingga memiliki persepsi yang sama dengan pelaku. Berdasarkan pengamatan terhadap berita hoax yang beredar, ternyata berita hoax memiliki pola-pola tertentu dalam penulisan beritanya yang dilakukan dan disebarkan secara masif (Assidik, 2018). Apabila penyebaran berita *hoax* dibiarkan akan berdampak buruk bagi masyarakat seperti pengaburan fakta, hilangnya kepercayaan masyarakat hingga dapat mengakibatkan perpecahan antar masyarakat.

Ada beberapa media yang sering digunakan untuk melakukan penyebaran berita *hoax* seperti media cetak, email, situs web dan media sosial. Dari beberapa media penyebaran *hoax* tersebut, menurut hasil survei Masyarakat Telematika Indonesia (Mastel) menunjukkan bahwa media sosial menjadi sumber utama peredaran berita *hoax* (Librianty, 2017). Hal tersebut karena tingginya pengguna media sosial di Indonesia yang mencapai 150 juta orang atau sebesar 56% dari total populasi. Dari data tersebut juga menujukkan bahwa pengguna media sosial meningkat 15% dari tahun sebelumnya (Wearesocial, 2019). Ada beberapa jenis berita *hoax* yang sering diterima masyarakat seperti isu di bidang sosial politik, SARA, kesehatan, IPTEK, bencana alam, dan lain-lain. Banyaknya berita *hoax* yang tersebar membuat sulitnya masyarakat dalam memilah dan membedakan mana berita *hoax* dan tidak. Ada beberapa masyarakat yang kritis terhadap berita yang tersebar, lalu melakukan pengecekan apakah berita tersebut *hoax* atau tidak. Namun, tidak sedikit pula masyarakat yang langsung percaya terhadap berita yang disampaikan. Maka dari itu, perlu dikembangkan sebuah sistem yang dapat digunakan untuk mengklasifikasikan suatu konten berita sehingga dapat diketahui apakah berita tersebut *hoax* atau bukan.

Penelitian yang berkaitan dengan *hoax* pernah dilakukan menggunakan metode *naïve bayes* dengan tingkat akurasi 80% untuk mengklasifikasikan 3 kategori yaitu berita palsu (*hoax*), berita asli (*fact*) dan berita yang tidak berindikasi (Tanjung, 2018). Selain itu, ada penelitian lain yang membandingkan antara algoritma *rocchio* dengan *multinomial naïve bayes* dengan akurasi yang didapatkan sebesar 83,501% untuk *rocchio* dan 65,835% untuk *multinomial naïve bayes* (Afriza & Adisantoso, 2018). Algoritma *support vector machine* pun pernah digunakan untuk klasifikasi artikel *hoax* dengan tingkat akurasi yang didapatkan sebesar 95,8333% (Maulina & Sagara, 2018). Ada beberapa metode lain yang pernah digunakan seperti *gradient descent backpropagation* dengan tingkat akurasi 72,04% sedangan *lavenberg-marquad backpropagation* memiliki akurasi 72,19% dalam klasifikasi pengguna media sosial (Lhaksmana, Nhita, & Budhiarto, 2017). Penerapan *deep learning* juga pernah dilakukan dalam klasifikasi sentimen analisis *tweet* dengan menggunakan metode *deep belief network* dengan tingkat akurasi sebesar 93,31% (Zulfa & Winarko, 2017).

Ada tantangan yang harus dihadapi dalam proses klasifikasi berita *hoax* yaitu banyaknya berita *hoax* yang memiliki tingkat kemiripan yang sangat tinggi dengan berita aslinya. Bahkan dalam beberapa kasus, penambahan atau perubahan beberapa kata dari berita asli akan merubah maksud dan tujuan dari berita tersebut. Seperti kendala dari penggunaan metode *naïve bayes*, dimana kemiripan data membuat sulitnya menemukan pola berita *hoax* sehingga mempengaruhi tingkat error dan akurasinya (Tanjung, 2018). Selain itu, banyaknya data yang digunakan membuat waktu yang diperlukan untuk melakukan *preprocessing* dan *training* semakin lama. Untuk meminimalisisr hal tersebut, dapat dilakukan dengan menggunakan metode *deep learning*.

Kelebihan dari penggunaan metode *deep learning* tersebut karena memiliki algoritma dengan struktur dan jumlah jaringan syaraf yang sangat banyak sebagai jaringan tersembunyi (*hidden layer*) yang dapat digunakan untuk berbagai hal, salah satunya klasifikasi teks. Metode *deep learning* yang dapat digunakan untuk klasifikasi teks yaitu *Convolutional Neural Network (CNN), Recurrent Neural Network (RNN),* dan *Hierarchical Attention Network (HAN)*. Ketiga metode tersebutmemiliki layer hirarki banyak/bertumpuk-tumpuk sehingga kendala yang berkaitan dengan kemiripan antar data dapat diminimalisir dan dapat menemukan pola dari berita *hoax*. Dengan hal tersebut, diharapkan hasil yang didapat pun menjadi lebih akurat. Kelebihan lain yang dimiliki metode-metode tersebut yaitu menghilangkan ekstraksi fitur secara manual, tidak memerlukan informasi terhadap data yang akan dipelajarinya, dan dapat dikombinasikan dengan *Graphics Processing Unit* (GPU) untuk mempercepat proses *preprocessing* maupun *training*. Algoritma dari *deep learning* dapat melakukan *tuning* (penyetelan) dan pemilihan model yang paling optimal secara mandiri. Selain itu, metode tersebut terbukti memiliki tingkat akurasi yang tinggi dan efisien dalam proses pengklasifikasian. Seperti penelitian yang pernah dilakukan oleh Razi (2017) tentang Klasifikasi Artikel Berita Berbahasa Indonesia Menggunakan *Convolutional Neural Network* dengan nilai akurasi sebesar 96,70% dan presisi, recall serta f-measure mencapai 96,60%. Ketiga metode tersebut juga pernah dibandingkan untuk klasifikasi teks dengan hasil dari CNN, RNN, dan HAN secara berturut turut yaitu 96,12%, 95,76%, 95,6% menggunakan dataset pertama, 87,88%, 82,63%, 87,91% menggunakan dataset kedua, dan dengan dataset ketiga menghasilkan 91,6%, 89,54%, 92,16% (Maheshwari, 2018).

Oleh karena itu, penelitian ini akan mengkomparasikan metode *convolutional neural network, recurrent neural network* dan *hierarchical attention network* untuk klasifikasi berita *hoax.* Penelitian ini akan menggunakan data yang bersumber dari turnbackhoax.id yang telah memvalidasi berita *hoax* yang beredar di media sosial. Data tersebut akan menjadi data latih dan juga data uji. Penelitian ini diharapkan dapat mengetahui metode *deep learning* mana yang memiliki tingkat performa yang lebih baik. Selain itu, penelitian ini juga diharapkan dapat menghasilkan sebuah sistem yang dapat dimanfaatkan untuk melakukan klasifikasi berita berdasarkan pola-pola tertentu sehingga dapat diketahui apakah berita tersebut termasuk kategori hoax atau bukan.

* 1. **Perumusan Masalah**

Berdasarkan latar belakang yang telah dipaparkan, maka dapat diambil rumusan masalah sebagai berikut:

1. Bagaimana tingkat akurasi algoritma *convolutional neural network, recurrent neural network* dan *hierarchical attention network* dalam klasifikasi berita *hoax*?
2. Bagaimana tingkat efisiensi algoritma *convolutional neural network, recurrent neural network* dan *hierarchical attention network* dalam klasifikasi berita *hoax*?
   1. **Batasan Masalah**

Menghindari terjadinya topik permasalahan yang lebih luas, maka diberikan batasan masalah sebagai berikut:

1. Data penelitian didapatkan dari situs web turnbachoax.id yang telah memvalidasi berita *hoax* yang beredar di media sosial.
2. Data penelitian yang digunakan dari tahun 2015 sampai dengan 2019.
3. Data penelitian yang digunakan akan dikategorikan menjadi dua kelas yaitu *hoax* dan fakta.
4. Data penelitian yang digunakan berbahasa Indonesia.
5. Metode klasifikasi yang digunakan adalah metode *Convolutional Neural Network* (*CNN*)*, Recurrent Neural Network* (*RNN*)dan *Hierarchical Attention Network* (*HAN*).
6. Penelitian ini akan melakukan klasifikasi berdasarkan bentuk atau pola dari berita.
   1. **Tujuan Penelitian**

Tujuan pada penelitian ini adalah sebagai berikut:

1. Dapat mengetahui tingkat akurasi algoritma *convolutional neural network, recurrent neural network* dan *hierarchical attention network* untuk klasifikasi berita *hoax*.
2. Dapat mengetahui tingkat efisiensi algoritma *convolutional neural network, recurrent neural network* dan *hierarchical attention network* untuk klasifikasi berita *hoax*.
   1. **Manfaat Penelitian**

Dengan dilakukannya penelitian ini dapat mengetahui algoritma *deep learning* mana yang memiliki tingkat performa yang lebih tinggidalam menyelesaikan permasalahan tentang klasifikasi berita *hoax*. Selain itu, diharapkan dengan adanya sistem ini dapat memudahkan masyarakat khususnya pengguna media sosial dalam memilah dan menyaring berita yang tersebar. Sehingga dampak dari penyebaran berita *hoax* dapat diminimalisir.

* 1. **Metodologi Penelitian**

Metodologi penelitian yang digunakan pada penelitian ini adalah sebagai berikut:

* + 1. **Metodologi Pengumpulan Data**

Alur metodologi pengumpulan data yang digunakan pada penelitian ini adalah sebagai berikut:

1. Studi Literatur

Studi literatur dilakukan dengan cara mencari dan menghimpun data ataupun sumber-sumber pustaka yang dapat mendukung penelitian serta memberikan informasi untuk menyelesaikan permasalahan pada penelitian ini. Studi kepustakaan yang digunakan bersumber dari buku, jurnal, artikel dan paper yang berkaitan pada penelitian ini.

1. Pengumpulan Data

Data yang digunakan pada penelitian ini bersumber dari situs turnbackhoax.id. Situs tersebut telah memvalidasi kebenaran dari berita-berita yang beredar di media sosial. Pengumpulan data dilakukan dengan menggunakan teknik *web srcaping* dengan ketentuan data dari tahun 2015 sampai dengan data tahun 2019.

* + 1. **Metodologi Pengembangan Sistem**

Metodologi pengembangan sistem yang digunakan adalah metode *prototype.* Penggunaan metode *prototype* dikarenakan metode ini lebih menekankan pada komunikasi antara pengembang dan pelanggan sehingga kebutuhan pengguna dapat diterjemahkan dalam bentuk model (prototipe) dan hasil yang diharapkan pun dapat tercapai. Ada beberapa tahapan yang dilakukan pada metode *prototype* yaitu (Pressman, 2010):

1. *Communication*

Pada tahapan ini dilakukan proses komunikasi antara pengembang dan pelanggan mengenai tujuan dibuatnya suatu perangkat lunak. Selain itu, akan dilakukan analisis serta identifikasi kebutuhan apa saja yang akan diperlukan selama proses pembuatan perangkat lunak.

1. *Quick Plan dan Modeling Quick Design*

Pada bagian ini akan membuat perencanaan dan pemodelan secara cepat berdasarkan kebutuhan yang telah diidentifikasi pada tahap sebelumnya. Perencanaan yang akan dilakukan dapat berupa pembagian tugas dan rencana kerja. Sedangkan dari sisi pemodelan akan membuat desain model yang merepresentasikan aspek-aspek yang diinginkan dari pelanggan.

1. *Construction of Prototype*

Pada proses *contruction of prototype,* pengembang akan mulai membuat program (perangkat lunak) berdasarkan rencana dan model yang telah dibuat sebelumnya. Pada tahap ini juga termasuk *testing* dari perangkat lunak yang telah berhasil dibuat.

1. *Deployment Delivery & Feedback*

Setelah perangkat lunak berhasil dibuat dan telah melalui *testing,* proses selanjutnya yaitu tahap *deployment delivery & feedback*. Perangkat lunak tadi akan disampaikan kepada pelanggan untuk dilakukan pengecekan. Jika mendapatkan *feedback* baik, maka perangkat lunak akan diterima oleh pelanggan dan pembuatan perangkat lunak telah selesai. Apabila *feedback* yang diberikan kurang baik atau tidak sesuai dengan pelanggan, maka kembali ke tahap *communication* dengan membahas perbaikan yang perlu dilakukan oleh pengembang.

* 1. **Sistematika Penulisan**

Sistematika penulisan yang digunakan dalam menyusun laporan penelitian ini adalah sebagai berikut :

Bab I Pendahuluan

Pada bagian ini membahas tentang latar belakang masalah, perumusan masalah, batasan masalah, tujuan penelitian, manfaat penelitian, metodologi penelitian, dan sistematika penulisan.

Bab II Tinjauan Pustaka

Tinjauan pustaka memuat tentang dasar teori yang digunakan untuk analisis dan perancangan sistem serta implementasi pada penelitian ini. Selain itu juga sebagai bahan referensi dan pondasi untuk memperkuat argumentasi dalam penelitian ini. Teori-teori yang sesuai dengan penelitian ini antara lain media sosial, *hoax,* klasifikasi, *deep learning,* *convolutional neural network, recurrent neural network,* dan *hierarchical attention network* .

Bab III Analisis dan Perancangan Sistem

Pada bagian ini akan membahas mengenai analisa dan perancangan sistem hingga implementasi sistem dalam mengklasifikasi berita *hoax* di media sosial.

Bab IV Hasil, Pengujian dan Pembahasan

Pada bab ini akan menyajikan hasil penelitian yang berisi hasil implementasi dari perancangan yang telah dibuat pada bab sebelumnya. Selain itu berisi pengujian terhadap hasil penelitian beserta pembahasannya.

Bab V Kesimpulan dan Saran

Pada bagian ini berisi kesimpulan dari hasil penelitian dan saran yang diajukan oleh penulis untuk pengembangan penelitian selanjutnya.

**BAB II**

**TINJAUAN PUSTAKA**

* 1. ***Hoax***

*Hoax* merupakan suatu kepalsuan yang disamarkan seolah-olah itu adalah kebenaran (MacDougall, 1958). *Hoax* dapat diartikan juga sebagai rangkaian informasi yang memang sengaja disesatkan, namun ‘dijual’ sebagai kebenaran (MacDougall, 1958). Beredarnya *hoax* dapat membentuk opini public yang mengarah kepada terjadinya kehebohan di masyarakat (MacDougall, 1958). Ada beberapa jenis informasi yang termasuk *hoax*, antara lain (Vibriza, Rahadi, Marwan, & Ahyad, 2017):

1. *Fake News* (berita palsu)
2. *Clickbait* (tautan jebakan)
3. *Confirmation Bias* (bias informasi)
4. *Misinformation* (informasi yang salah)
5. *Satire* (sindiran)
6. *Post-truth* (pasca kebenaran)
7. *Propaganda* (provokasi)

Dengan mudahnya penyebaran informasi pada saat ini, sangat memudahkan pula berita *hoax* beredar di masyarakat. Sayangnya, masih banyak masyarakat yang percaya dengan berita *hoax* yang beredar. Ada dua faktor yang menyebabkan masyarakat percaya terhadap berita *hoax*. *Pertama,* karena *hoax* yang dilakukan secara berulang-ulang dan dilakukan secara masif dapat dianggap menjadi suatu kebenaran. Selain itu, seseorang akan cenderung percaya terhadap berita *hoax* jika informasinya sesuai dengan opini atau sikap yang dimiliki (Respati, 2017).

Munculnya *hoax* di masyarakat dilatarbelakangi oleh berbagai tujuan, mulai dari main-main, pembentukan opini publik, penipuan hingga hasutan untuk memecah belah masyarakat. Berbagai tujuan tersebut memunculkan dampak negatif dari penyebaran berita *hoax*. Dampak negatif dari berita *hoax* tersebut seperti pengaburan fakta, hilangnya kepercayaan masyarakat hingga dapat mengakibatkan perpecahan antar masyarakat. Hal tersebut akan menjadi semakin buruk jika penyebaran berita *hoax* terus menerus dibiarkan.

Ada berbagai cara yang dapat dilakukan untuk terhindar dari berita *hoax*, seperti membaca seluruh isi berita, menelusuri sumber berita dan membandingkan antar berita. Selain itu, berita *hoax* juga biasanya memiliki pola-pola tertentu. Ada tujuh pola atau ciri-ciri berita *hoax* yang perlu diketahui sebagai antisipasi agar tidak termakan *hoax* yang sedang beredar, antara lain (Nasution, 2017): *Pertama,* terdapat anjuran untuk menyebarkan pesan tersebut ke orang lain. *Kedua*, penggunaan tata bahasa yang kurang sempurna. *Ketiga,* tidak adanya berita lain yang mendukung pesan tersebut. *Keempat,* pesan yang tersebar tidak logis, tidak konsisten dan bertentangan dengan akal sehat. *Kelima,* tidak menyebutkan kenyataan yang dapat dibuktikan dengan fakta-fakta. *Keenam,* pesan berantai atau pesan yang telah di*foward* berkali-kali. *Ketujuh,* pembuat *hoax* akan menghubungkan pesan tersebut dengan sumber resmi yang sebenarnya palsu seperti *hoax* tentang pemenang undian berhadiah dengan menyertakan alamat website yang terlihat seperti asli namun sebenarnya website palsu yang telah dibuat sendiri sebelumnya.

* 1. **Media Sosial**

Menurut *McGraw Hill Dictionary,* media sosial adalah segala bentuk media komunikasi interaktif yang memungkinkan terjadinya interaksi dua arah dan umpan balik. Media sosial dapat didefinisikan sebagai sarana yang digunakan oleh orang-orang untuk berinteraksi satu sama lain dengan cara menciptakan, berbagi serta bertukar informasi dan gagasan dalam sebuah jaringan dan komunikasi virtual. Dengan kata lain, media yang digunakan untuk menjadi sosial (Safko, 2012). Media sosial dapat digunakan untuk berbagi teks, gambar, suara dan video (Kotler & Keller, 2012). Berdasarkan berbagai penjelasan tersebut, dapat disimpulkan bahwa media sosial merupakan media komunikasi yang digunakan untuk melakukan interaksi satu sama lain dengan tujuan untuk berbagi dan bertukar informasi baik berupa teks, gambar, suara maupun video untuk menjadi makhluk sosial.

Media sosial dapat dikenali melalui beberapa karakteristik atau ciri-ciri yang dimiliknya, seperti adanya partisipasi pengguna untuk memberikan umpan balik terhadap konten di media sosial, adanya keterbukaan bagi pengguna untuk memberikan komentar dan sebagainya, adanya interaksi antar pengguna dan semua itu saling terhubung satu sama lain.

Pada saat ini, media sosial memiliki peran besar dalam membangun pola pikir dan perilaku dalam berbagai bidang kehidupan masyarakat. Hal tersebut tidak lepas dari berbagai fungsi yang ditawarkan dalam penggunaan media sosial. Beberapa fungsi dari media sosial, yaitu (Wijayanto, 2012):

* 1. ***Web Scraping***

*Web scraping* adalah salah satu metode untuk melakukan pengambilan data. Cara kerja dari metode ini dengan mengekstraksi data yang berada di suatu website dan menganalisis data tersebut untuk digunakan bagi kepentingan lain (Turland, 2010). Secara garis besar terdapat dua teknik untuk melakukan *web scraping,* yaitu:

1. *Manual*, yaitu menyalin data dari suatu webseite dengan cara *copy paste*

2. *Otomatis,* yaitu dengan menggunakan koding, aplikasi ataupun *extension browser*

* 1. ***Text Mining***

*Text mining* merupakan variasi dari data *mining* yang berusaha menemukan pola yang menarik dari sekumpulan data tekstual yang berjumlah besar (Feldman & Sanger, 2007). *Text mining* adalah satu langkah dari analisis teks yang dilakukan secara otomatis oleh komputer untuk menggali informasi yang berkualitas dari suatu rangkaian teks yang terangkum dalam sebuah dokumen (Han & Kamber, 2006). *Text mining* bertujuan untuk menemukan pola atau kata-kata yang mewakili isi dari dokumen yang selanjutnya akan di analisis.

* 1. ***Text Preprocessing***

*Text preprocessing* adalah proses untuk membersihkan dan mempersiapkan data sebelum dilakukannya proses klasifikasi (Haddi, Liu, & Shi, 2013). Adanya *text preprocessing* bertujuan untuk mendapatkan bentuk data yang siap olah. Ada beberapa tahapan pada *text preprocessing* yaitu :

* + 1. ***Case Folding***

Tahap *case folding* adalah proses merubah semua huruf pada dokumen menjadi huruf kecil (*lowercase*). Huruf yang diterima hanya ‘a’ sampai dengan ‘z’. Karakter selain itu akan dihilangkan dan dianggap *delimiter*.

* + 1. ***Tokenizing***

Tahap *tokenizing* atau tokenisasi adalah proses pemisahan teks dari paragraph atau kalimat menjadi bagian-bagian tertentu yang disebut sebagai token. Pemisah antar token biasanya adalah spasi atau tanda baca. Contoh tokenisasi dari kalimat ”komparasi metode deep learninguntuk klasifikasi berita hoax” akan menghasilkan 8 token, yaitu: “komparasi”, “metode”, “deep”, “learning”, “untuk”, “klasifikasi”, “berita”, dan “hoax”.

* + 1. ***Filtering***

Tahap *filtering* adalah proses pengambilan kata-kata penting (*wordlist*) atau membuang kata kata kurang penting (*stopword*) dari token yang telah dihasilkan pada tahap sebelumnya. Pembuangan *stopword* tidak akan mengubah makna dan isi dari suatu data. Beberapa contoh *stopword* seperti yang, juga, aku, kamu, ini, dan itu.

* + 1. ***Stemming***

Tahap *stemming* adalah proses pemisahan kata yang mengandung awalan, akhiran maupun sisipan sehingga menghasilkan kata dasar. Hal ini bertujuan untuk mengurangi varian kata dengan makna yang hampir sama. Selain itu, proses *stemming* dapat meningkatkan kinerja pengambilan data karena jumlah indeks dengan istilah yang berbeda menjadi berkurang.

* 1. **Klasifikasi Teks**

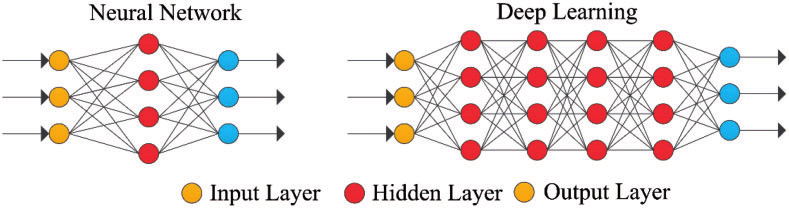
Klasifikasi teks adalah proses untuk menemukan sebuah model atau fungsi yang menggambarkan dan membedakan antar kelas data. Klasifikasi teks didefinisikan sebagai pengkategorian teks secara otomatis ke dalam satu atau lebih kelas yang telah ditentukan berdasarkan isinya (Sebastiani, 2002). Tujuan dari klasifikasi teks yaitu untuk memperkirakan kelas dari suatu objek yang belum diketahui kelasnya (Gaikwad, Chaugule, & Patil, 2014).

* 1. ***Word Embedding***

*Word embedding* adalah proses untuk mengubah suatu kata menjadi sebuah vektor yang terdiri dari kumpulan angka. Jumlah angka yang dihasilkan dari proses *word embedding* disebut dimensi. Penggunaan *word embedding* biasanya untuk algoritma *deep learning,* karenakan algoritma *deep learning* tidak dapat memproses data dalam bentuk *string,* maka dari itu harus dikonversi ke dalam bentuk angka terlebih dahulu.

* 1. ***Deep Learning***

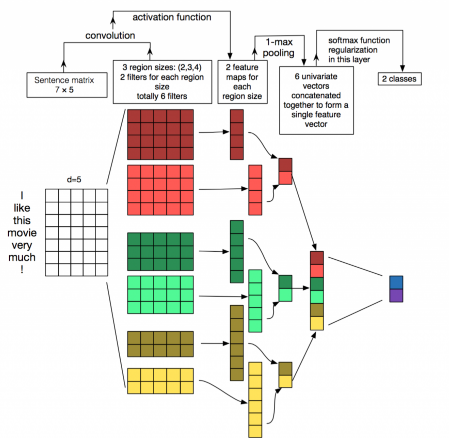
*Deep learning* adalah salah satu bidang dari pembelajaran mesin (*machine learning*) yang berbasis jaringan syaraf tiruan. *Deep learning* merupakan sebuah pendekatan dalam penyelesaian masalah pada sistem pembelajaran komputer yang menggunakan konsep hirarki (Goodfellow, Bengio, & Courville, 2016). Hirarki pada *deep learning* terdiri dari banyaknya struktur dan jumlah jaringan syaraf pada algoritmanya hingga ratusan lapisan yang setiap lapisannya memiliki tanggung jawabnya masing-masing.



**Gambar 2.1** Diagram *Deep Learning* (Xing & Du, 2018)

* 1. ***Convolutional Neural Network* (CNN)**

*Convolutional neural network* (CNN) adalah salah satu algoritma dari *deep learning*. CNN merupakan jaringan syaraf yang dikhususkan untuk mengolah data yang memiliki grid. Sebenarnya, CNN merupakan pengembangan dari *Multilayer Perceptron* (MLP) yang di desain untuk mengolah data dua dimensi. Namun, ada beberapa penelitian yang menerapkan algoritma CNN pada *Natural Language Processing* (NLP) dengan hasil yang cukup memuaskan. Jadi, selain digunakan untuk mengolah data dua dimensi, CNN juga dapat digunakan untuk mengolah data satu dimensi termasuk melakukan klasifikasi teks (Razi, 2017) . Konsep dari CNN 1D tidak jauh berbeda dari *neural network* biasa dimana terdiri dari *neuron* yang memiliki *weight, bias* dan *activation function*. Secara garis besar, arsitektur dari CNN 1D dibagi menjadi beberapa tahapan seperti dibawah ini:



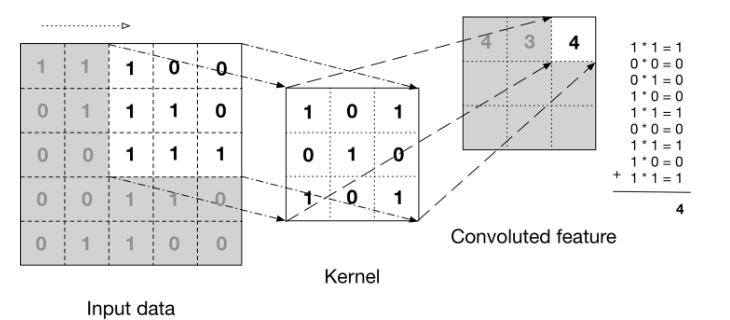
**Gambar 2.2** Arsitektur *Convolutional Neural Network* (Zhang & Wallace, 2015)

* + 1. ***Input Layer***

Layer input adalah layer pertama pada algoritma CNN. Layer ini berfungsi untuk menampung data yang menjadi inputan. Data yang ditampung berupa nilai vektor yang merepresentasikan dari masing-masing kata yang merupakan hasil dari proses *word embedding*. Ukuran layer ini sesuai dengan banyaknya kata dikali dengan panjang vektor atau dimensi vektor. Misalnya terdapat data yang terdiri dari 5 kata dan masing-masing kata direpresentasikan dengan 100 dimensi vektor, maka layer input berukuran .

* + 1. ***Convolution Layer***

*Convolution layer* adalah proses utama dari CNN. Pada tahap ini, *convolutional layer* akan menggunakan filter atau kernel pada setiap kumpulan vektor yang menjadi masukan. Kernel ini berupa array dua dimensi, bisa berukuran , , atau . Pada proses ini akan menghasilkan *feature map* yang menunjukkan fitur atau ciri dari kumpulan vektor inputan.



**Gambar 2.3** *Convolution Layer* pada CNN (Yin, 2018)

Berdasarkan penjelasan tersebut dapat dirumuskan persamaan untuk menghasilkan *feature map* dengan inputan data , kernel dilambangkan dengan dan sebagai *feature map* yang dihasilkan:

(2.1)

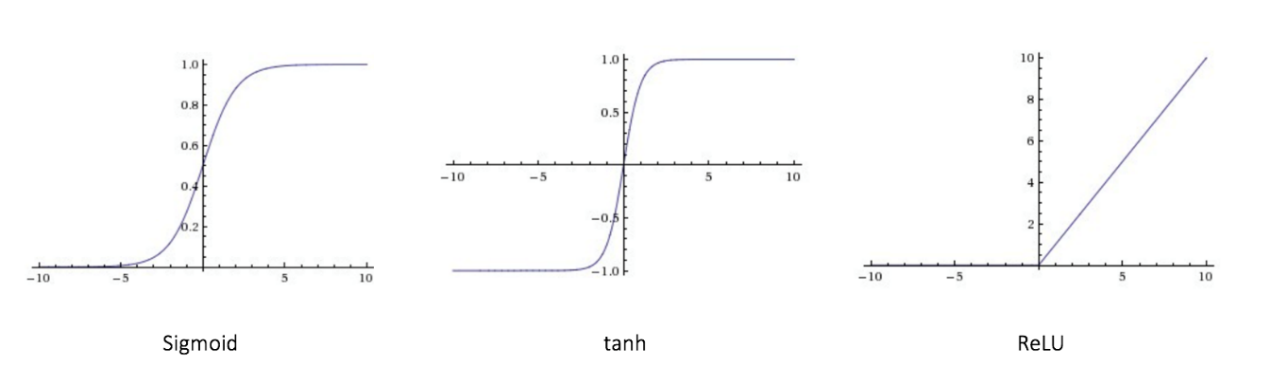
*Convolution layer* memiliki *hyperparameter* yang nilainya mempengaruhi *feature map* yang akan dihasilkan. *Hyperparameter* yang digunakan pada proses konvolusi yaitu:

**Tabel 2.1** *Hyperparameter* pada *convolutional layer* (Zufar & Setiyono, 2016)

|  |  |  |
| --- | --- | --- |
| **No** | ***Hyperparameter*** | **Keterangan** |
| 1 | *Depth* | Kedalaman *layer* atau jumlah *layer* konvolusi |
| 2 | *Stride* | Jumlah pergeseran *filter* pada proses konvolusi |
| 3 | *Zero-padding* | Jumlah penambahan nilai intensitas nol di daerah sekitar input gambar |

* + 1. ***Activation Layer***

Tahap selanjutnya yaitu *activation layer*. Proses pada *activation layer* yaitu dengan memasukkan nilai *feature map* yang telah dihasilkan pada tahap sebelumnya ke dalam fungsi aktivasi. Fungsi aktivasi berfungsi untuk mengubah nilai pada *feature map* pada range tertentu dengan tujuan untuk meneruskan nilai yang menjadi fitur dominan ke layer selanjutnya. Beberapa fungsi aktivasi yang sering digunakan secara umum sebagai berikut:



**Gambar 2.4** Contoh Grafik Fungsi Aktivasi (Tarabay, 2019)

Sigmoid:

(2.2)

Tanh:

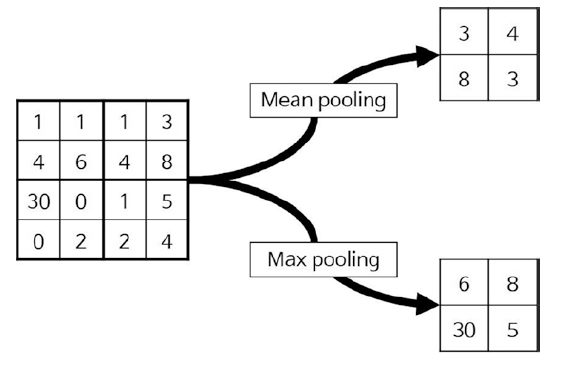
(2.3)

ReLU:

(2.4)

* + 1. ***Pooling Layer***

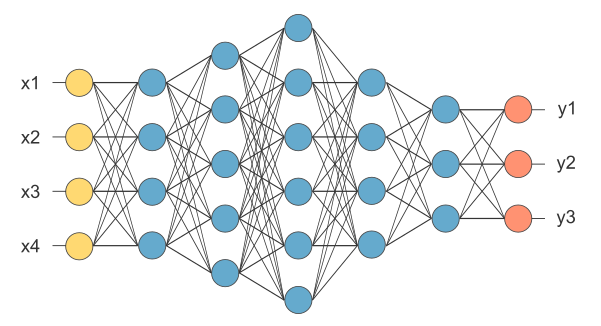
*Pooling layer*  dapat disebut juga dengan *subsampling* atau *downsampling*. Hal tersebut karena *pooling layer* akan mengurangi dimensi dari *feature map* yang telah melalui *activation layer*. Pengurangan dimensi pada tahap ini tidak mempengaruhi informasi penting yang ada didalamnya. Proses *pooling* memiliki beberapa macam, seperti *maxpooling*, *meanpooling, sumpooling* dan sebagainya. Proses pada *pooling* dimulai dengan menentukan ukuran *downsampling,* misalnya . Setelah itu akan dilakukan proses *pooling* pada *feature map,* sebagai contoh *feature map* yang digunakan berukuran .



**Gambar 2.5** *Pooling Layer* (Tandungan, 2019)

* + 1. ***Fully Connected Layer***

*Fully connected layer* akan menggunakan hasil dari *pooling layer* sebagai inputan. Layer ini memiliki struktur yang sama dengan *neural network* pada umumnya yang terdiri dari *input layer*, *hidden layer* dan *output layer*. Masing-masing layer tersebut terdapat kumpulan *neuron* yang saling terhubungdengan *neuron-neuron* pada layer tetangganya. Berikut bentuk dari *full connected layer*:



**Gambar 2.6** *Fully Connected Layer* (Tandungan, 2019)

Layer ini akan menentukan hasil klasifikasi berdasarkan data yang telah dimasukkan. Proses penentuan hasil klasifikasi menggunakan fungsi aktivasi . Persamaan dari fungsi aktivasi sebagai berikut:

(2.5)

dimana :

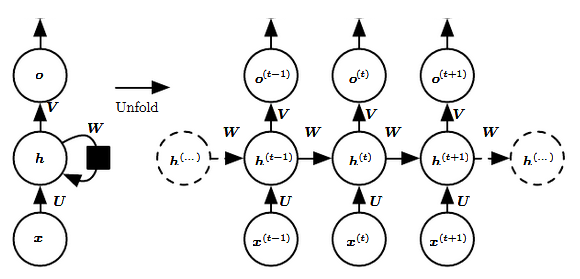
σ = probabilitas input tersebut merupakan class j

z = persamaan linear fungsi pre-aktivasi

j = salah satu class dari K class yang ada

* 1. ***Recurrent Neural Network* (RNN)**

*Recurrent Neural Network* atau RNN adalah algoritma *neural network* yang memiliki kemampuan *feedback* kembali ke *neuron* itu sendiri ataupun ke *neuron* lainnya sehingga aliran informasi dari *inputan* mempunyai arah jamak. Maka dari itu, RNN merupakan *neural network* yang digunakan untuk memproses data *sequence* atau data yang memiliki urutan dengan indeks dimulai dari data sampai data (Nabi, 2019). Penggunaan RNN pada data yang memiliki urutan data karena RNN memiliki memori sebagai pengingat histori masa lalu. RNN melakukan tugas yang sama untuk setiap elemen dari suatu urutan data, dimana *output* yang dihasilkan bergantung pada perhitungan sebelumnya. RNN memiliki minimal satu *feedback loop* di dalam *recurrent*. *Feedback loop* mempengaruhi kemampuan belajar dan kinerja jaringan (Udin et al., 2017).



**Gambar 2.7** Arsitektur *Recurrent Neural Network* (Goodfellow et al., 2016)

Berdasarkan arsitektur RNN diatas, dibagian sebelah kiri menunjukkan notasi dari RNN sedangkan bagian sebelah kanan merupakan jaringan yang belum terbuka namun jika terbuka akan masuk ke dalam jaringan penuh. Sebagai contoh, terdapat kalimat yang terdiri dari 5 kata maka jaringan dibuka menjadi 5 layer *neural network* dimana masing-masing layer mewakili satu kata. Dari arsitektur RNN terdapat beberapa atribut yang perlu diketahui:

1. *Input*  : merupakan data yang dimasukkan kedalam jaringan dengan panjang berupa vektor. Misalanya terdapat suatu kalimat yang memiliki 5 kata maka vektor dari kata pertama dari kalimat tersebut merupakan dan begitu pula seterusnya.
2. *Hidden state* () : merupakan jaringan tersembunyi dengan banyak dan akan bertindak sebagai memori dari jaringan. Nilai dari berdasarkan input terbaru dan hasil dari *hidden state* sebelumnya dengan rumus . Fungsi berdasarkan transformasi *non-linear* (*activation function*)yang digunakan, seperti tanh, sigmoid, ReLU dan lain sebagainya.
3. *Weights* () : Input yang akan dimasukkan ke dalam *hidden state* memiliki diparameterisasi bobot matriks , saat berada di *hidden state* ke *hidden state* selanjutnya diparameterisasi dengan bobot matriks , dan hasil dari *hidden state* ke *output* disebut bobot matriks . Semua bobot matriks () dibagi sepanjang waktu ().
4. *Output* () : menggambarkan hasil dari jaringan.
   * 1. ***Forward Pass***

Setelah membahas beberapa istilah dari RNN, dapat dilakukan *forward pass* untuk melakukan *training* modelnya. Misalnya terdapat beberapa konten berita yang terdiri dari beberapa kalimat dan berisi beberapa kata dengan *output* yang diharapkan berupa kelas dari beritanya. Untuk konsep *forward pass* akan ditinjau dari layer pada saat (berada ditengah). Karena RNN memperhatikan *output* dari data perhitungan sebelumnya pada , maka untuk menghitung *hidden state* dengan melakukan perkalian antara bobot matriks dengan paramter dan dijumlahkan dengan hasil perkalian antara parameter dan . Hasil tersebut lalu diproses menggunakan *activation function*, misalnya menggunakan tanh. Detail perhitungannya sebagai berikut:

(2.6)

Setelah diproses dengan mendapatkan hasilnya, maka nilai diteruskan ke *output* dengan melakukan perkalian dengan parameter . Dari hasil tersebut diproses dengan fungsi . Dapat dirumuskan menjadi sebagai berikut:

(2.7)

* + 1. ***Backward Pass***

*Backward pass* merupakan arsitektur RNN yang memperhatikan deret waktu sebelumnya. Apabila ingin mendapatkan nilai gradien pada waktu maka harus mengkalkulsi nilai gradien sampai dengan kondisi . Istilah ini sering disebut dengan *back-propagation through time* (BPPT) karena parameter dibagi oleh semua *time step* pada jaringan.

* + 1. ***Computing Gradients***

Dengan menggunakan *loss function ,* dapat digunakan untuk menghitung bobot dari matriks dengan istilah bias dan dan hasil pembaruannya *learning rate .* Untuk memperbaharui bobot untuk mengurangi *loss* dengan persamaan sebagai berikut:

(2.8)

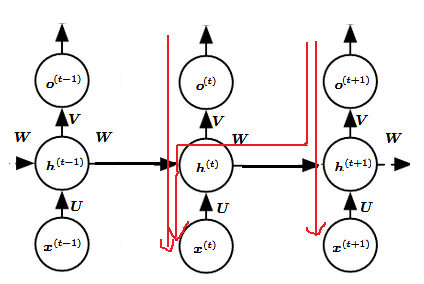
Hal yang sama juga dilakukan pada bobot dan .

Selanjutnya menghitung gradien dengan menggunakan konsep BPPT pada RNN. Berdasarkan grafik komputasi terdapat parameter dan dengan urutan data dan memiliki indeks tertentu terhadap . Dari setiap *node*  perlu menghitung gradien secara rekursif.

Gradien untuk *output* merupakan hasil kalkulasi dari yang telah digunakan *activation function* untuik mendapatkan probabilitas dari vektor . Dengan asumsi bahwa *loss* merupakan kemungkinan negatif dari target sebelumnya.

(2.9)

Setelah mengetahui bagaimana gradien mengalir melalui *hidden state,* sekarang dapat dilihat dengan jelas melihat diagram pada waktu mengalir gradien dari *output* saat ini ke *hidden state* selajutnya.

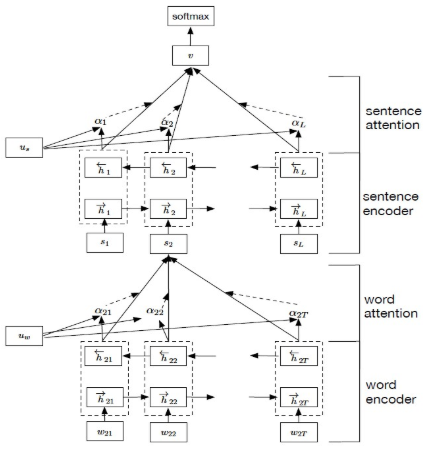


**Gambar 2.8** Alur Gradien pada RNN (Nabi, 2019)

Berdasarkan semua penjelasan mengenai RNN diatas, dapat disimpulkan beberapa tahapan yang dilakukan untuk menggunakan ataupun menerapkan algoritma RNN dalam suatu kasus yang ingin diselesaikan. Tahapan-tahapannya sebagai berikut:

1. Melakukan inisialisasi bobot matriks dengan distribusi acak dan bias dengan nol.
2. Melakukan *forward propagation* untuk menghitung nilai prediksi
3. Hitung nilai *loss*
4. Lakukan proses *back-propagation* untuk menghitung gradien
5. Perbaharui nilai bobot berdasarkan gradien
6. Ulangi langkah 2-5
   1. ***Hierarchical Attention Network* (HAN)**

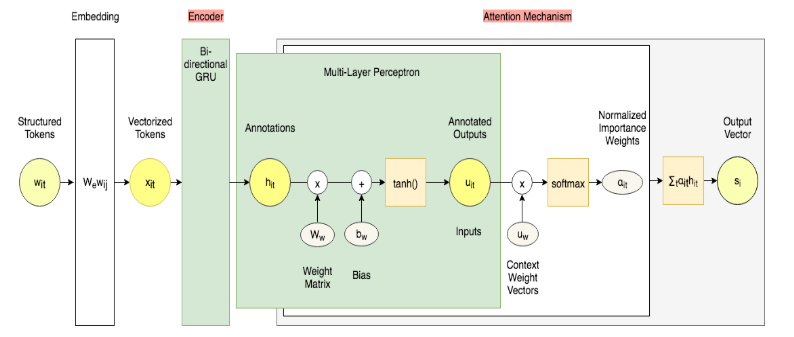
*Hierarchical attention network* (HAN) merupakan algoritma *neural network* yang mempertimbangan struktur hierarki suatu dokumen. Hierarki tersebut merepresentasikan gabungan antara representasi dokumen dengan representasi kalimat. Representasi kalimat dibangun terlebih dahulu dengan *encode* kata dari suatu kalimat dan melalui *attention mechanism* untuk menghasilkan vektor yang merepresentasikan suatu kalimat. Representasi dokumen juga menggunakan cara yang sama, bedanya yang menjadi inputannya adalah vektor kalimat. Berikut arsitektur dari algoritma HAN:



**Gambar 2.9** Arsitektur *Hierarchical Attention Network* (Kränkel & Lee, 2018)

* + 1. ***Word Level***

Tahap ini adalah tahap awal pada algoritma HAN. Prosesnya akan dimulai dari data hasil *preprocessing* yang berupa kumpulan token yang terstruktur yang akan menjadi inputan pada tahap ini. Sedangkan *output* yang akan dihasilkan setelah melalui tahapan ini berupa vektor untuk masing-masing token. Berikut alur dari tahap *word level* yang terdiri dari *word encoder* dan *word attention*:



**Gambar 2.10** Alur pada tahap *word layer* (Kränkel & Lee, 2018)

Token yang terstruktur merupakan representasi dari kata ke dari kalimat . Karena model tidak dapat memproses data berupa teks biasa atau tipe data string, maka diperlukan proses *embedding*. Ada beberapa algoritma yang populer yang dapat digunakan untuk melakukan proses *embedding* yaitu word2vec dan GloVe. Layer *embedding* akan menghasilkan vektor multidimensional untuk masing-masing token . Dengan cara tersebut, token-token akan direpresentasikan secara numerik sebagai . Dari penjelasan tersebut dapat dirumuskan sebagai berikut:

(2.10)

* + 1. ***Word Encoder***

Setelah mendapatkan vektor dari masing-masing token, tahap berikutnya dengan melakukan proses *encoding*. Proses *encoding* akan menggunakan *Gated Recurrent Network* (GRU). GRU akan memprediksi suatu kata bergantung pada kata sebelumnya. GRU memiliki *hidden state* sebagai sel memori untuk transfer informasi. Terdapat dua gerbang yang memutuskan apakah suatu informasi disimpan atau dilupakan sekaligus melakukan pembaharuan informasi pada model. Tujuan dari layer ini akan mengekstrak data yang revelan dari suatu kalimat. Hal ini biasa disebut dengan *annotations*. Pada model ini, *bidirectional* GRU yang diterapkan untuk mendapatkan *annotation* kata dengan meringkas dari dua arah dengan variabel . Persamaan yang akan terbentuk sebagai berikut:

(2.11)

(2.12)

(2.13)

* + 1. ***Word Attention***

*Annotation* akan membangun *attention* yang dimulai dari *hidden layer* lainnya. Tujuannya untuk membiarkan model belajar dengan bobot dan bias yang diinisialisasi secara acak. *Annotation* akan terus diperbaharui dan direpresentasikan oleh . Selanjutnya, proses *actiovation function* dari hasil *annotation* dengan menggunakan fungsi . Fungsi ini akan mengoreksi nilai input antara -1 dan 1 serta memetakan nol hingga mendekati nol.

(2.14)

Anotasi baru akan dikalikan dengan vektor yang telah dilatih . Setelah itu, akan dinormalisasi untuk mendapatkan bobot penting per kata menggunakan fungsi .

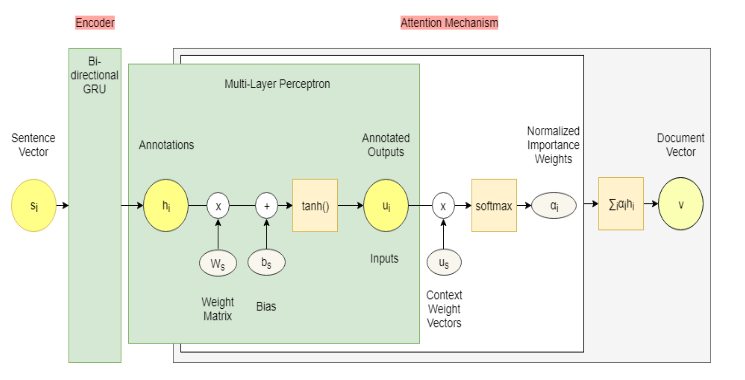
(2.15)

Bobot-bobot penting yang telah didapatkan disatukan dan dijumlah seluruhnya. Hasil dari penjumlah semua bobot disebut vektor kalimat.

(2.16)

* + 1. ***Sentence Level***

Selanjutnya jaringan akan dijalankan pada level kalimat dengan prosedur yang sama seperti level kata, tetapi sekarang fokus pada kalimat . Bedanya, pada level kata terdapat *embedding layer* sedangkan pada level kalimat tidak perlu *embedding layer*. Hal tersebut dikarenakan inputan pada level kalimat sudah berupa vektor kalimat yang merupakan hasil dari level kata. Berikut jalannya proses pada *sentence level*:



**Gambar 2.11** Alur pada tahap *sentence level* (Kränkel & Lee, 2018)

* + 1. ***Sentence Encoder***

Konteks kalimat akan dirangkum menggunakan *bidirectional* GRU dengan alur maju dan mundur. Persamaannya sebagai berikut:

(2.17)

(2.18)

(2.19)

* + 1. ***Sentence Attention***

Bobot dan bias yang didapat akan dilatih kembali dengan inisialisasi secara acak. Hasil akhirnya yaitu berupa vektor dokumen yang bisa digunakan sebagai fitur untuk melakukan klasifikasi dokumen.

(2.20)

(2.21)

(2.22)

* 1. **Penelitian Terdahulu**

Penelitian yang berkaitan dengan *hoax* pernah dilakukan menggunakan metode *naïve bayes* dengan tingkat akurasi 80% untuk mengklasifikasikan 3 kategori yaitu berita palsu (*hoax*), berita asli (*fact*) dan berita yang tidak berindikasi (Tanjung, 2018). Selain itu, ada penelitian lain yang membandingkan antara algoritma *rocchio* dengan *multinomial naïve bayes* dengan akurasi yang didapatkan sebesar 83,501% untuk *rocchio* dan 65,835% untuk *multinomial naïve bayes* (Afriza & Adisantoso, 2018). Algoritma *support vector machine* pun pernah digunakan untuk klasifikasi artikel *hoax* dengan tingkat akurasi yang didapatkan sebesar 95,8333% (Maulina & Sagara, 2018). Ada beberapa metode lain yang pernah digunakan seperti *gradient descent backpropagation* dengan tingkat akurasi 72,04% sedangan *lavenberg-marquad backpropagation* memiliki akurasi 72,19% dalam klasifikasi pengguna media sosial (Lhaksmana, Nhita, & Budhiarto, 2017). Penerapan *deep learning* juga pernah dilakukan dalam klasifikasi sentimen analisis *tweet* dengan menggunakan metode *deep belief network* dengan tingkat akurasi sebesar 93,31% (Zulfa & Winarko, 2017). Algoritma *convolutional neural network* juga pernah digunakan pada penelitian tentang klasifikasi artikel berita berbahasa indonesia dengan nilai akurasi sebesar 96,70% dan presisi, recall serta f-measure mencapai 96,60%.

**Tabel 2.2** Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Penulis** | **Judul** | **Metode** | **Keterangan** |
| 1 | Budi Setiawan Tanjung.  2018 | Pendekatan *Text Mining* sebagai Sistem Pendeteksi Pemberitaan Palsu yang Tersebar dalam Twitter | *Naive Bayes Classifier* | Pola berita palsu belum dapat dikenali secara jelas karena banyak berita yang hampir sama. |
| 2 | Aulia Afriza dan Julio Adisantoso.  2018 | Metode Klasifikasi *Rocchio* untuk Analisis *Hoax* | *Rocchio* dan Multinomial *Naive Bayes* | Pada penelitian ini tidak memiliki fitur untuk menambahkan model pada data latih |
| 3 | Kemas Muslim Lhaksmana, Fhira Nhita dan Ageng Budhiarto.  2017 | Klasifikasi Pengguna Media Sosial Twitter dalam Persebaran *Hoax* Menggunakan Metode *Backpropagation* | *Gradient Descent Backpropagation* dan *Lavenberg Marquad Backpropagation* | Performa sistem baik, namun mendapatkan error yang cukup besar karena adanya kemiripan karakteristik pada data dan membutuhkan waktu pelatihan yang relatif lama |

**Tabel 2.3 Lanjutan** Penelitian Terdahulu

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **No** | **Penulis** | **Judul** | **Metode** | **Keterangan** |
| 4 | Dina Maulina dan Rofie Sagara.  2018 | Klasifikasi Artikel *Hoax* Menggunakan *Support Vector Machine Linear* dengan Pembobotan *Term Frequency-Inverse Document Frequency* | *Support Vector Machine* | Data yang digunakan tergolong sedikit (108 artikel *hoax* dan 132 artikel non *hoax*) |
| 5 | Ira Zulfa dan Edi Winarko.  2017 | Sentimen Analisis Tweet Berbahasa Indonesia dengan *Deep Belief Network* | *Deep Belief Network* | Penggunaan metode *Deep Belief Network* dengan *BagofWord* sebagai fitur ekstraksi terbukti tidak memberikan akurasi lebih baik |
| 6 | Ar Razi.  2017 | Klasifikasi Artikel Berita Berbahasa Indonesia Menggunakan *Convolutional Neural Network* | *Convolutional Neural Network* | Jumlah dataset pada proses *training* masih sedikit dan pada *preprocessing* tidak menggunakan *stemming* serta *stopword* sehingga hasilnya kurang optimal |

**Daftar Pustaka**

Afriza, A., & Adisantoso, J. (2018). Metode Klasifikasi Rocchio untuk Analisis Hoax. *Jurnal Ilmu Komputer Dan Agri-Informatika*, *5*(1), 1–10. https://doi.org/10.29244/jika.5.1.1-10

Assidik, G. K. (2018). Kajian Identifikasi dan Upaya Penangkalan Pemberitaan Palsu (Hoax) Pada Pembelajaran Bahasa Indonesia. *Kongres Bahasa Indonesia*.

Fajri, D. (2019). Menkominfo: Ada 900 Ribu Situs Penyebar Informasi Hoax. Retrieved June 29, 2019, from www.okezone.com website: https://news.okezone.com/read/2019/02/14/337/2018062/menkominfo-ada-900-ribu-situs-penyebar-informasi-hoax

Feldman, R., & Sanger, J. (2007). *The Text Mining Handbook : Advanced Approaches in Analyzing Unstructured Data* (1st ed.). New York: Cambridge University Press.

Gaikwad, S. V., Chaugule, A., & Patil, P. (2014). Text Mining Methods and Techniques. *International Journal of Computer Applications*, *85*(17), 42–45. https://doi.org/10.5120/14937-3507

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.

Haddi, E., Liu, X., & Shi, Y. (2013). The Role of Text Pre-processing in Sentiment Analysis. *Procedia Computer Science*, (17), 26–32.

Han, J., & Kamber, M. (2006). *Data Mining: Concepts and Techniques* (2nd ed.). San Francisco: Morgan Kaufmann Publishers.

Kotler, P., & Keller, K. L. (2012). *Marketing Management* (14th ed.). United State of America: Pearson.

Kränkel, M., & Lee, H.-E. (2018). Text Classification with Hierarchical Attention Networks. Retrieved February 11, 2020, from Seminar Information System website: https://humboldt-wi.github.io/blog/research/information\_systems\_1819/group5\_han/

Lhaksmana, K. M., Nhita, F., & Budhiarto, A. (2017). Klasifikasi Pengguna Media Sosial Twitter Dalam Persebaran Hoax Menggunakan Metode Backpropagation Classification of Users Social Media Twitter in the Hoax Spread. *E-Proceeding of Engineering*, *4*(2), 3082–3090.

Librianty, A. (2017). Survei: Media Sosial Jadi Sumber Utama Penyebaran Hoax.

MacDougall, C. D. (1958). *Hoaxes*. New York: Dover Publication.

Maheshwari, A. (2018). *Report on Text Classifiction using CNN, RNN & HAN*.

Maulina, D., & Sagara, R. (2018). Klasifikasi Artikel Hoax Menggunakan Support Vector Machine Linear Dengan Pembobotan Term Frequency – Inverse Document Frequency. *Jurnal Mantik Penusa*, *2*(1), 35–40.

Nabi, J. (2019). Recurrent Neural Networks (RNNs). Retrieved February 11, 2020, from https://towardsdatascience.com/ website: https://towardsdatascience.com/recurrent-neural-networks-rnns-3f06d7653a85

Nasution, M. A. (2017). HOAX SEBAGAI BENTUK HUDUD MENURUT HUKUM ISLAM. *Yurisprudentia*, *3*(1).

Pressman, R. S. (2010). *Software Engineering : A Practitioner’s Approach* (7th ed.). McGraw-Hill Education.

Razi, A. (2017). *Klasifikasi Artikel Berita Berbahasa Indonesia Menggunakan Convolutional Neural Network*. Universitas Gadjah Mada.

Respati, S. (2017). Mengapa Banyak Orang Mudah Percaya Berita “Hoax”? Retrieved February 9, 2020, from www.kompas.com website: https://nasional.kompas.com/read/2017/01/23/18181951/mengapa.banyak.orang.mudah.percaya.berita

Safko, L. (2012). *The Social Media Bible : Tactics, Tools and Strategies for Business Success* (3rd Editio). New York: John Wiley & Sonic Inc.

Sebastiani, F. (2002). Machine Learning in Automated Text Categorization. *ACM Computing Surveys*, *34*(1), 1–47.

Tandungan, S. (2019). Pengenalan Convolutional Neural Network – Part 1. Retrieved February 13, 2020, from http://sofyantandungan.com/ website: http://sofyantandungan.com/pengenalan-convolutional-neural-network-part-1/

Tanjung, B. S. (2018). Pendekatan Text Mining sebagai Sistem Pendeteksi Pemberitaan Palsu yang Tersebar dalam Twitter. *Universitas Widya Kartika*, 1–6.

Tarabay, M. R. (2019). PyTorch and deep learning. Retrieved February 13, 2020, from https://rafietarabay.blogspot.com/2019/05/?m=0

Turland, M. (2010). *php|architect’s Guide to Web Scraping with PHP*. Los Angeles.

Udin, M., Kaloko, B. S., Hardianto, T., Elektro, J. T., Teknik, F., Unej, U. J., & Kalimantan, J. (2017). Peramalan Kapasitas Baterai Lead Acid pada Mobil Listrik Berbasis Levenberg Marquardt Neural Network. *BERKALA SAINSTEK*, *2*, 112–117.

Vibriza, J., Rahadi, D. R., Marwan, M. R., & Ahyad. (2017). Perilaku pengguna dan informasi. *Jurnal Gunadarma*, *4*(1), 192–208. https://doi.org/https://doi.org/10.22146/jps.v4i2.28586

Wearesocial. (2019). Digital 2019: Indonesia. Retrieved June 29, 2019, from www.datareportal.com website: https://datareportal.com/reports/digital-2019-indonesia/

Wijayanto, F. I. (2012). Social Media: Definisi, Fungsi, Karakteristik. Retrieved February 9, 2020, from www.prezi.com website: https://prezi.com/vddmcub\_-ss\_/social-media-definisi-fungsi-karakteristik/

Xing, W., & Du, D. (2018). Dropout Prediction in MOOCs : Using Deep Learning for Personalized Intervention. *Journal of Educational Computing Research*, (March). https://doi.org/10.1177/0735633118757015

Yin, L. (2018). A Summary of Neural Network Layers. Retrieved February 13, 2020, from https://medium.com/ website: https://medium.com/machine-learning-for-li/different-convolutional-layers-43dc146f4d0e

Zhang, Y., & Wallace, B. (2015). *A Sensitivity Analysis of (and Practitioners’ Guide to) Convolutional Neural Networks for Sentence Classification*.

Zufar, M., & Setiyono, B. (2016). Convolutional Neural Network untuk Pengenalan Wajah Secara Real-Time. *Jurnal Sains Dan Seni ITS*, *5*(2), 72–77.

Zulfa, I., & Winarko, E. (2017). Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, *11*(2), 187–198. https://doi.org/10.22146/ijccs.24716