



**ANALISIS SENTIMEN TWITTER BAHASA
INDONESIA MENGGUNAKAN ALGORITMA
*CONVOLUTIONAL NEURAL NETWORK***

Skripsi

diajukan sebagai salah satu persyaratan untuk memperoleh gelar
**Sarjana Pendidikan Program Studi Pendidikan Teknik Informatika dan
Komputer**

Oleh
Sartini
NIM.5302416006

**PENDIDIKAN TEKNIK INFORMATIKA DAN KOMPUTER
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNIK
UNIVERSITAS NEGERI SEMARANG
2020**

PERSETUJUAN PEMBIMBING

Nama : Sartini
NIM : 5302416006
Program Studi : Pendidikan Teknik Informatika dan Komputer
Judul : Analisis Sentimen Twitter Bahasa Indonesia
Menggunakan Algoritma *Convolutional Neural Network*

Skripsi ini telah disetujui oleh pembimbing untuk diajukan ke sidang panitia ujian
Skripsi Program Studi Pendidikan Teknik Informatika dan Komputer Fakultas
Teknik Universitas Negeri Semarang.

Semarang, 6 Agustus 2020
Pembimbing



Dr. Ir. Subiyanto, S.T., M.T.

NIP. 197411232005011001

PENGESAHAN

Skripsi dengan judul *Analisis Sentimen Twitter Bahasa Indonesia Menggunakan Algoritma Convolutional Neural Network* telah dipertahankan di depan sidang Panitia Ujian Skripsi Fakultas Teknik UNNES pada tanggal 11 Agustus 2020.

Oleh

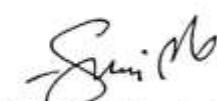
Nama : Sartini

NIM : 5302416006

Program Studi : Pendidikan Teknik Informatika dan Komputer

Panitia:

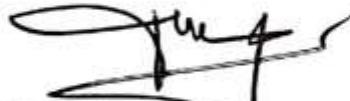
Ketua



Ir. Ulfah Mediaty Arief, M.T., IPM.

NIP. 196605051997022001

Sekretaris



Budi Sunarko, S. T., M. T., Ph. D.

NIP. 197101042006041001

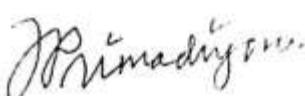
Pengaji 1



Drs. Said Sunardiyo, M.T.

NIP. 196505121991031003

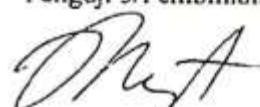
Pengaji 2



Drs. Yohanes Primadiyono, M.T.

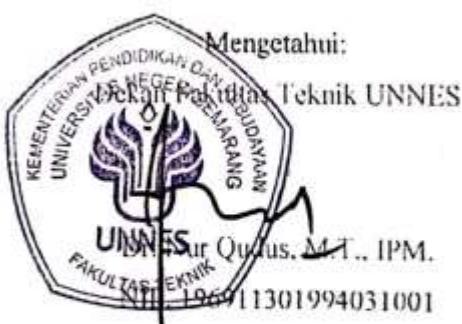
NIP. 196209021987031002

Pengaji 3/Pembimbing



Dr. Ir. Subiyanto, S.T., M.T.

NIP. 197411232005011001



PERNYATAAN KEASLIAN

Dengan ini saya menyatakan bahwa:

1. Skripsi ini adalah asli dan belum pernah diajukan untuk mendapatkan gelar akademik (sarjana, magister, dan/ atau doktor), baik di Universitas Negeri Semarang (UNNES) maupun di perguruan tinggi lain.
2. Karya tulis ini adalah murni gagasan, rumusan, dan penelitian saya sendiri, tanpa bantuan pihak lain, kecuali arahan Pembimbing dan masukan Tim Pengaji.
3. Dalam karya tulis ini tidak terdapat karya atau pendapat yang telah ditulis atau dipublikasikan orang lain, kecuali secara tertulis dengan jelas dicantumkan sebagai acuan dalam naskah dengan disebutkan nama pengarang dan dicantumkan dalam daftar pustaka.
4. Pernyataan ini saya buat dengan sesungguhnya dan apabila di kemudian hari terdapat penyimpangan dan ketidakbenaran dalam pernyataan ini, maka saya bersedia menerima sanksi akademik berupa pencabutan gelar yang telah diperoleh karena karya ini, serta sanksi lainnya sesuai dengan norma yang berlaku di perguruan tinggi ini.

Semarang, 6 Agustus 2020

Yang membuat pernyataan,



Sartini

NIM. 5302416006

MOTTO DAN PERSEMBAHAN

Motto

Allah tidak membebani seseorang melainkan sesuai dengan kesanggupannya. Dia mendapat pahala dari kebijakan yang dikerjakannya dan dia mendapat siksa dari kejahanan yang diperbuatnya (Q.S. Al-Baqarah: 286)

Persembahan

1. Untuk orang tua, nenek, kakak, dan adik tercinta yang senantiasa tulus mendoakan serta memberi dukungan kepada saya
2. Untuk almamater saya prodi Pendidikan Teknik Informatika dan Komputer Jurusan Teknik Elektro, Universitas Negeri Semarang
3. Tim UNNES *Electrical Engineering Students Research Group* (UEESRG) yang memberikan wadah untuk saya belajar penelitian
4. Teman-teman Jurusan Teknik Elektro 2016 yang senantiasa berjuang bersama dalam menempuh studi
5. Komunitas Data Science Indonesia *Chapter* Jawa Tengah yang memberikan ilmu tentang dunia *Data Scientist*
6. Teman-teman yang sudah menguatkan saya dalam mengerjakan skripsi

RINGKASAN

Sartini. 2020. Analisis Sentimen *Twitter* Bahasa Indonesia Menggunakan Algoritma *Convolutional Neural Network*. Dr. Ir. Subiyanto, S.T., M.T. Pendidikan Teknik Informatika dan Komputer.

Analisis sentimen dalam bidang *natural language processing* telah dikembangkan menggunakan metode seperti *machine learning* dan *deep learning*. Namun, penelitian analisis sentimen yang dilakukan di Indonesia masih menggunakan metode *machine learning* yang menyebabkan akurasi yang dihasilkan belum optimal. Dalam skripsi ini, algoritma *Convolutional Neural Network* (CNN) digunakan untuk mendapatkan akurasi yang lebih tinggi untuk diterapkan pada *Indonesian-Sentiment-Analysis-Dataset* yang terdiri dari 10.806 *tweet* berbahasa Indonesia.

Variasi 12 model CNN dikembangkan dengan konfigurasi parameter yang berbeda pada *convolutional layers*, *filter size*, dan *number of filters* yang dikembangkan menggunakan bahasa pemrograman *Python* pada *Google Colaboratory*. Model *Word2vec* berbahasa Indonesia diterapkan sebagai representasi kata dalam bentuk vektor. Desain kerangka kerja sistem analisis sentimen dibagi menjadi tiga bagian yaitu persiapan data, sistem *training* dan *testing* dengan 80% data sebagai data *training* dan 20% sebagai data *testing*, dan *output* sebagai evaluasi atau ukuran kinerja model yang dikembangkan dalam bentuk grafik.

Hasil menunjukkan bahwa algoritma CNN untuk analisis sentimen pada *dataset* umum berbahasa Indonesia memperoleh akurasi yang lebih tinggi dibandingkan algoritma *machine learning* seperti *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Stochastic Gradient Descent* (SGD). Algoritma CNN memperoleh akurasi sebesar 81,4%, SVM memperoleh akurasi sebesar 61,4%, KNN sebesar 52,3%, dan SGD sebesar 62,1%.

Kata kunci: *Analisis Sentimen, CNN, Twitter, Word2vec*.

PRAKATA

Segala puji dan syukur penulis ucapkan kehadiran Allah SWT yang telah melimpahkan rahmat-Nya sehingga penulis dapat menyelesaikan Skripsi yang berjudul Analisis Sentimen *Twitter* Bahasa Indonesia Menggunakan Algoritma *Convolutional Neural Network*. Skripsi ini disusun sebagai salah satu persyaratan meraih gelar Sarjana Pendidikan pada Program Studi S1 Pendidikan Teknik Informatika dan Komputer Universitas Negeri Semarang. Shalawat dan salam disampaikan kepada Nabi Muhammad SAW, mudah-mudahan kita semua mendapat safaat-Nya di yaumil akhir nanti, aamiin.

Penyelesaian karya tulis ini tidak lepas dari bantuan berbagai pihak, oleh karena itu pada kesempatan ini penulis menyampaikan ucapan terima kasih serta penghargaan kepada:

1. Prof. Dr. Fathur Rokhman, M.Hum., Rektor Universitas Negeri Semarang atas kesempatan yang diberikan kepada penulis untuk menempuh studi di Universitas Negeri Semarang.
2. Dr. Nur Qudus, M.T., IPM., Dekan Fakultas Teknik, Ir. Ulfah Mediaty Arief, M.T., IPM., Ketua Jurusan Teknik Elektro, Budi Sunarko, S.T., M.T., Ph.D., Koordinator Program Studi Pendidikan Teknik Informatika dan Komputer atas fasilitas yang disediakan bagi mahasiswa.
3. Dr. Ir. Subiyanto, S.T., M.T., Pembimbing yang penuh perhatian dan atas perkenaan memberi bimbingan dan dapat dihubungi disertai kemudahan menunjukkan sumber-sumber yang relevan dengan penulis karya ini.
4. Drs. Said Sunardiyo, M.T. dan Drs. Yohanes Primadiyono, M.T., Pengui yang telah memberi masukan yang sangat berharga berupa saran, ralat, perbaikan, pernyataan, komentar, tanggapan, menambah bobot, dan kualitas karya tulis ini.
5. Semua dosen Jurusan Teknik Elektro FT UNNES yang telah memberi bekal pengetahuan yang berharga.
6. Berbagai pihak yang telah memberi bantuan untuk karya tulis ini yang tidak bisa disebutkan satu persatu.

Penulis berharap semoga Skripsi ini dapat bermanfaat untuk pelaksanaan penelitian selanjutnya.

Semarang, 6 Agustus 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
PERSETUJUAN PEMBIMBING	ii
PENGESAHAN	iii
PERNYATAAN KEASLIAN	iv
MOTTO DAN PERSEMBAHAN.....	v
RINGKASAN	vi
PRAKATA	vii
DAFTAR ISI.....	viii
DAFTAR TABEL	x
DAFTAR GAMBAR.....	xi
DAFTAR LAMPIRAN	xiv
BAB I PENDAHULUAN.....	1
1.1 Latar Belakang	1
1.2 Identifikasi Masalah.....	4
1.3 Perumusan Masalah.....	5
1.4 Pembatasan Masalah	5
1.5 Tujuan Penelitian	6
1.6 Manfaat Penelitian	6
BAB II KAJIAN PUSTAKA DAN LANDASAN TEORI	7
2.1. Kajian Pustaka	7
2.2. <i>Natural Language Processing</i>	10
2.3 Analisis Sentimen	12
2.4 <i>Text Pre-processing</i>	13
2.5 <i>Feature Extraction</i>	16
2.6 <i>Artificial Neural Network (ANN)</i>	21
2.6.1 <i>Activation Function</i>	22
2.6.2 Arsitektur Neural Network	28
2.7 <i>Convolutional Neural Network</i>	31

2.7.1	<i>Layer pada Convolutional Neural Network</i>	32
2.7.2	Arsitektur <i>Convolutional Neural Network</i>	35
BAB III METODOLOGI PENELITIAN		38
3.1	Waktu dan Tempat Penelitian	38
3.2	Alat dan Bahan Penelitian	39
3.3	Parameter Penelitian.....	40
3.4	Desain Penelitian	42
3.5	Persiapan Data.....	43
3.6	<i>Modelling CNN</i>	48
3.6.1	Tahap Pelatihan	50
3.6.2	Tahap Pengujian	52
3.7	Teknik Pengumpulan Data.....	52
3.8	Teknik Analisis Data	53
BAB IV HASIL DAN PEMBAHASAN		55
4.1	Performa Pelatihan dan Validasi.....	55
4.2	Performa Pengujian Model	64
4.3	Komparasi Performa Model.....	77
4.3.1	Komparasi Performa Antar Variasi Model CNN	77
4.3.2	Komparasi Performa Model CNN dengan <i>Machine Learning</i> .	79
BAB V SIMPULAN DAN SARAN.....		80
5.1	Kesimpulan	80
5.2	Saran.....	80
DAFTAR PUSTAKA		82
LAMPIRAN.....		88

DAFTAR TABEL

Tabel 2.1 Contoh Proses <i>Tokenization</i>	14
Tabel 3.1 Jadwal Penelitian	38
Tabel 3.2 Alat yang Digunakan dalam Penelitian	39
Tabel 3.3 Parameter Pembentukan Model CNN	41
Tabel 3.4 Pengaturan Eksperimen Algoritma CNN	41
Tabel 4.1 Nilai <i>Performa Training Model</i>	64
Tabel 4.2 <i>Classification Report</i> Model CNN 1	66
Tabel 4.3 <i>Classification Report</i> Model CNN 2	67
Tabel 4.4 <i>Classification Report</i> Model CNN 3	68
Tabel 4.5 <i>Classification Report</i> Model CNN 4	69
Tabel 4.6 <i>Classification Report</i> Model CNN 5	70
Tabel 4.7 <i>Classification Report</i> Model CNN 6	71
Tabel 4.8 <i>Classification Report</i> Model CNN 7	72
Tabel 4.9 <i>Classification Report</i> Model CNN 8	73
Tabel 4.10 <i>Classification Report</i> Model CNN 9	74
Tabel 4.11 <i>Classification Report</i> Model CNN 10	75
Tabel 4.12 <i>Classification Report</i> Model CNN 11	76
Tabel 4.13 <i>Classification Report</i> Model CNN 12	77
Tabel 4.14. Nilai <i>Performa Testing Model</i>	77

DAFTAR GAMBAR

Gambar 2.1 Bagian <i>Natural Language Processing</i>	10
Gambar 2.2 Ilustrasi <i>Oversampling</i> dan <i>Undersampling</i>	16
Gambar 2.3 <i>Contextual and Semantic Similarity</i>	18
Gambar 2.4 Arsitektur <i>Skip-Gram</i>	19
Gambar 2.5 Arsitektur CBOW	20
Gambar 2.6 Ilustrasi <i>Neuron</i> dengan Model Matematis	22
Gambar 2.7 <i>Linear Activation Function</i>	23
Gambar 2.8 <i>Sigmoid Function</i>	25
Gambar 2.9 <i>Tanh Function</i>	26
Gambar 2.10 <i>ReLU Function</i>	27
Gambar 2.11 Perceptron <i>Input</i> dan <i>Ouput</i>	28
Gambar 2.12 Arsitektur <i>Neural Network</i>	29
Gambar 2.13 <i>Sliding Window</i>	33
Gambar 2.14 1D <i>Convolution</i>	33
Gambar 2.15 <i>Max Pooling</i> dan <i>Average Pooling</i>	34
Gambar 2.16 LeNet-5 <i>Architecture</i>	35
Gambar 2.17 AlexNet <i>Architecture</i>	36
Gambar 2.18 GoogleNet <i>Architecture</i>	36
Gambar 2.19 Arsitektur CNN untuk Analisis Sentimen	37
Gambar 3.1. Jumlah Data Positif, Negatif, dan Netral	40
Gambar 3.2 Diagram Alir Penelitian	43
Gambar 3.3 Desain Kerangka Kerja Sistem Analisis Sentimen	43
Gambar 3.4 Tampilan Lima Data Teratas Sebelum Mengulang <i>Stopword Removal</i>	44
Gambar 3.5 Lima Data Terbawah Sebelum Mengulang <i>Stopword Removal</i>	44

Gambar 3.6 Tampilan Lima Data Teratas Setelah Mengulang <i>Stopword Removal</i>	45
Gambar 3.7 Lima Data Terbawah Setelah Mengulang <i>Stopword Removal</i> ..	45
Gambar 3.8 <i>Undersampling</i> dan <i>Oversampling</i> Data	46
Gambar 3.9 <i>Worcloud</i> Data Positif	46
Gambar 3.10 <i>Worcloud</i> Data Negatif	47
Gambar 3.11 <i>Worcloud</i> Data Netral	47
Gambar 4.1 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 1	55
Gambar 4.2 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 2	56
Gambar 4.3 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 3	56
Gambar 4.4 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 4	56
Gambar 4.5 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 5	57
Gambar 4.6 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 6	57
Gambar 4.7 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 7	57
Gambar 4.8 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 8	58
Gambar 4.9 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 9	58
Gambar 4.10 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 10	58
Gambar 4.11 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 11	59
Gambar 4.12 Grafik <i>Training</i> (a) <i>acc</i> dan <i>val-acc</i>, (b) <i>loss</i> dan <i>val-loss</i> CNN 12	59

Gambar 4.13 Grafik <i>Training Accuracy</i> Variasi Model CNN 1-12	60
Gambar 4.14 Grafik <i>Training Loss</i> Variasi Model CNN 1-12	60
Gambar 4.15 Grafik <i>Validation Accuracy</i> Variasi Model CNN 1-12	61
Gambar 4.16 Grafik <i>Validation Loss</i> Variasi Model CNN 1-12	61
Gambar 4.17 <i>Confusion Matrix</i> CNN 1	65
Gambar 4.18 <i>Confusion Matrix</i> CNN 2	66
Gambar 4.19 <i>Confusion Matrix</i> CNN 3	67
Gambar 4.20 <i>Confusion Matrix</i> CNN 4	68
Gambar 4.21 <i>Confusion Matrix</i> CNN 5	69
Gambar 4.22 <i>Confusion Matrix</i> CNN 6	70
Gambar 4.23 <i>Confusion Matrix</i> CNN 7	71
Gambar 4.24 <i>Confusion Matrix</i> CNN 8	72
Gambar 4.25 <i>Confusion Matrix</i> CNN 9	73
Gambar 4.26 <i>Confusion Matrix</i> CNN 10	74
Gambar 4.27 <i>Confusion Matrix</i> CNN 11	75
Gambar 4.27 <i>Confusion Matrix</i> CNN 12	76
Gambar 4.28 Performa <i>Testing Model</i>	78
Gambar 4.29 Perbandingan Algoritma CNN dengan <i>Machine Learning</i>	79

DAFTAR LAMPIRAN

Lampiran 1. Kode Program <i>Exploratory Data Analysis</i>	88
Lampiran 2. Kode Program Persiapan Data	89
Lampiran 3. Ekstraksi Fitur Menggunakan <i>Word2vec</i>	91
Lampiran 4. Kode Program <i>Building Model</i>	93
Lampiran 5. Kode Program <i>Training dan Testing Model CNN 1-12</i>	96
Lampiran 6. Kode Program Grafik <i>Training Model CNN 1-12</i>	104
Lampiran 7. Kode Program <i>Confusion Matrix Model CNN 1-12</i>	106
Lampiran 8. <i>Model Summary CNN 1</i>	107
Lampiran 9. <i>Model Summary CNN 6</i>	108
Lampiran 10. Surat Penetapan Dosen Pembimbing Skripsi	109
Lampiran 11. Surat Tugas Seminar Proposal Skripsi	110
Lampiran 12. Surat Tugas Ujian Skripsi	111

BAB I

PENDAHULUAN

1.1 Latar Belakang

Pesatnya perkembangan media sosial membuat pengguna lebih mudah membagikan pendapat pada *platform online* seperti forum web, blog, dan situs media sosial *twitter* maupun *facebook* (Rani and Kumar, 2018; Zhao *et al.*, 2019). Media sosial sebagai media informasi dan komunikasi mulai berkembang pesat sejak internet mudah diakses yang digunakan untuk mengekspresikan pikiran, minat, dan pendapat orang-orang tentang berbagai hal (Fitri, Andreswari and Hasibuan, 2019). Hal ini telah menjadi tren baru untuk mempermudah dalam berinteraksi dan berkomunikasi, sehingga tingkat pertumbuhan pengguna media sosial meningkat dengan cepat selama bertahun-tahun (Saputri, Mahendra and Adriani, 2019). Salah satu media sosial paling popular adalah *twitter* yang memiliki 100 juta pengguna aktif dengan rata-rata 140 juta *tweet* dikirim setiap hari pada tahun 2011 (Vatsavai *et al.*, 2012). Pada tahun 2019 rata-rata pengguna aktif *twitter* per bulan mencapai 330 juta dengan pengguna terbanyak dari Amerika Serikat yaitu 48 juta, sedangkan di Indonesia jumlah pengguna *twitter* aktif selalu mengalami kenaikan dari tahun 2014-2019 mulai dari 12 juta pengguna menjadi 22,8 juta (Statista, 2015).

Konten *twitter* yang disebut *tweet* telah digunakan oleh para peneliti, pemerintah atau industri untuk mendapatkan pengetahuan yang dapat membantu dalam memecahkan masalah sehari-hari, termasuk berbagai perilaku manusia yang

sebenarnya dapat ditangkap dari *tweet* (Saputri, Mahendra and Adriani, 2019). Dengan banyaknya *review*, ulasan, dan komentar yang dapat ditangkap, *twitter* memiliki dampak penting bagi perusahaan-perusahaan untuk melakukan identifikasi sentimen secara otomatis terhadap setiap komentar mengenai perusahaan tersebut. Komentar dapat berupa komentar positif maupun komentar negatif yang secara otomatis dapat dicapai dengan melakukan analisis sentimen atau *sentiment analysis* (Juwiantho *et al.*, 2020).

Analisis sentimen menjadi topik yang populer dalam bidang *Natural Language Processing* (NLP) (Rani and Kumar, 2018) yang efektif untuk menemukan opini publik (Anwar Hridoy *et al.*, 2015). Analisis sentimen atau *opinion mining* merupakan studi komputasi atas sikap, pendapat, dan emosi orang pada suatu entitas. Entitas dapat mewakili individu, acara, maupun suatu topik. Topik-topik ini dapat berupa *review* atau ulasan (Medhat, Hassan and Korashy, 2014). Informasi dari ulasan yang ditemukan dapat digunakan untuk membantu orang, perusahaan, maupun organisasi untuk membuat keputusan yang lebih baik (Alharbi and de Doncker, 2019). Besarnya pengaruh dan manfaat dari analisis sentimen menyebabkan penelitian dan aplikasi berbasis analisis sentimen berkembang pesat, bahkan di Amerika terdapat sekitar 20-30 perusahaan memfokuskan pada layanan analisis sentimen (Indurkhya and Damerau, 2010). Penelitian analisis sentimen di Indonesia juga sangat populer seperti analisis sentimen terhadap operator seluler Indonesia yang berjumlah 1.000 *tweet* yang berisi komentar mengenai operator seluler (Wijaya *et al.*, 2013). Analisis sentimen juga diterapkan untuk kampanye anti LGBT untuk mengetahui kecenderungan

komentar publik tentang LGBT di Indonesia (Fitri, Andreswari and Hasibuan, 2019). Analisis sentimen berbahasa Indonesia yang sudah banyak diteliti menggunakan *dataset* dengan cakupan spesifik terhadap suatu kasus sehingga kurang cocok untuk digunakan pada analisis sentimen dengan jangkauan lebih luas. Terlebih apabila penelitian tersebut tidak memublikasikan data yang digunakan untuk pelatihan modelnya karena berbagai alasan, seperti data-data yang digunakan merupakan data sensitif atau bersumber dari perusahaan komersial (Ferdiana *et al.*, 2019).

Metode analisis sentimen dalam Bahasa Indonesia sudah banyak dilakukan menggunakan metode *classical machine learning* seperti *Naïve Bayes*, *Maximum Entropy* (ME), *Support Vector Machine* (SVM), dan *Decision Tree* (Anastasia and Budi, 2016; Windasari, Uzzi and Satoto, 2017; Lutfi, Permanasari and Fauziati, 2018; Fitri, Andreswari and Hasibuan, 2019). Sedangkan untuk penelitian analisis sentimen dalam Bahasa Inggris sudah menerapkan metode *deep learning* yaitu *Convolutional Neural Network* (CNN) yang mendapatkan hasil jauh lebih baik dibandingkan metode *classical machine learning* *Naïve Bayes* dengan peningkatan *Precision* 7%, *Recall* 8%, *F-1 Score* 9% (Yoo, Song and Jeong, 2018). CNN dapat mengekstraksi fitur dari informasi global dan melakukan operasi konvolusi dari *layer* sebelumnya sehingga data dapat diekstraksi sebagai fitur dan mempertimbangkan hubungan antar fitur tersebut (Rani and Kumar, 2018). Tetapi, masih sedikit penelitian tentang analisis sentimen dengan metode *Convolutional Neural Network* menggunakan bahasa Indonesia (Juwiantho *et al.*, 2020).

1.2 Identifikasi Masalah

Penelitian analisis sentimen di Indonesia masih menggunakan *dataset* spesifik untuk kasus atau permasalahan tertentu. Mengingat masih kurangnya publikasi *dataset*, (Ferdiana *et al.*, 2019) mengembangkan *dataset* umum berbahasa Indonesia yang berjumlah 10.806 *tweet* dan telah diberi label dengan tiga polaritas sentimen yaitu sentimen positif, sentimen negatif, dan sentimen netral. Hasil yang didapatkan bahwa *Indonesia-General-Sentiment-Analysis-Dataset* sudah dibandingkan dengan *dataset SemEval-2018* yang dapat digunakan sebagai acuan dasar untuk penelitian analisis sentimen.

Metode analisis sentimen berbahasa Indonesia juga masih menggunakan *classical machine learning* yang tidak memperhatikan pentingnya urutan kata pada suatu kalimat sehingga akurasi yang dihasilkan belum optimal. (Rani and Kumar, 2018) membandingkan akurasi analisis sentimen menggunakan metode *classical machine learning* seperti *Naïve Bayes*, *K-Nearest Neighbor*, *Maximum Entropy*, *Support Vector Machine* dengan algoritma *Convolutional Neural Network*. Hasil yang didapatkan bahwa algoritma CNN memberikan akurasi sebesar 95% dibandingkan metode *classical machine learning* dengan hasil akurasi tertinggi pada algoritma *Naïve Bayes* sebesar 90%. Akurasi yang dihasilkan meningkat sebesar 5% dari metode *classical machine learning* dibandingkan dengan metode *deep learning*.

Analisis sentimen menggunakan *deep learning* dilakukan oleh (Juwianto *et al.*, 2020) dengan menggunakan model *Word2Vec* untuk Bahasa Indonesia sebagai representasi kata dalam bentuk vektor. Penggunaan *Word2Vec* juga

mempercepat proses pelatihan dan meningkatkan akurasi algoritma *Deep Convolutional Neural Network*. Hasil percobaan yang telah dilakukan menghasilkan akurasi sebesar 76. 40%.

1.3 Perumusan Masalah

Penelitian analisis sentimen *twitter* sudah banyak diteliti di Indonesia yang sebagian besar menggunakan metode *classical machine learning* yang tidak memperhatikan pentingnya urutan kata pada suatu kalimat. Hal ini menyebabkan akurasi yang dihasilkan belum optimal dibandingkan menggunakan metode *deep learning*. Metode *deep learning* dengan algoritma *Convolutional Neural Network* melakukan operasi *convolution* menggunakan filter sebesar ukuran *window* untuk mendapatkan fitur berdasarkan urutan kata sehingga akurasi yang dihasilkan dapat lebih optimal. Namun, analisis sentimen *twitter* berbahasa Indonesia menggunakan algoritma *Convolutional Neural Network* belum banyak diteliti di Indonesia.

1.4 Pembatasan Masalah

Adapun pembatasan masalah dalam penelitian ini adalah:

1. Pada penelitian ini, analisis sentimen tidak digunakan untuk pemodelan analisis yang spesifik terhadap suatu produk tertentu.
2. *Dataset* diperoleh dari riset *Group Cloud Experience* yang disebut *Indonesian-General-Sentiment-Analysis-Dataset*.

3. Data yang digunakan adalah data *tweet* dalam Bahasa Indonesia yang hanya berupa teks dan sudah dilakukan tahap *preprocessing*.
4. Ekstraksi fitur yang digunakan menggunakan metode *Word2Vec* Bahasa Indonesia.
5. Uji validasi dilakukan dengan membandingkan metode *classical machine learning* seperti algoritma *Support Vector Machine* (SVM), *K-Nearest Neighbor* (KNN), dan *Stochastic Gradient Descent* (SGD). Algoritma yang digunakan untuk komparasi tidak dibahas secara detail.

1.5 Tujuan Penelitian

Tujuan dari penelitian ini adalah mendapatkan dan mengetahui tingkat akurasi yang setinggi-tingginya untuk kasus analisis sentimen *twitter* berbahasa Indonesia menggunakan algoritma *Convolutional Neural Network*.

1.6 Manfaat Penelitian

Manfaat yang diharapkan dari penelitian ini adalah menambah wawasan ilmu pengetahuan mengenai analisis sentimen menggunakan metode *deep learning*. Metode *deep learning* yang digunakan adalah algoritma *Convolutional Neural Network* yang diterapkan pada *dataset* umum berbahasa Indonesia dari sosial media *twitter*.

BAB II

KAJIAN PUSTAKA DAN LANDASAN TEORI

2.1. Kajian Pustaka

Terdapat beberapa penelitian terdahulu yang relevan untuk dijadikan referensi dalam penelitian ini. Adapun penelitian tersebut dari (Hassan and Mahmood, 2017) berjudul ‘*Deep Learning Approach for Sentiment Analysis of Short Texts*’. Penelitian tersebut mengusulkan arsitektur *neural network ConvLstm*, yang menggabungakan *Convolutional Neural Network* (CNN) dan *Long Short-Term Memory* (LSTM) untuk analisis sentimen pada teks pendek. Data yang digunakan berasal dari *Stanford Sentiment Treebank* dan *IMDB Dataset*. Hasil akurasi yang diperoleh 88,3% untuk *binary classification* dan 47,5% untuk klasifikasi lima kelas sentimen.

Penelitian selanjutnya dilakukan oleh (Rani and Kumar, 2018) berjudul ‘*Deep Learning Based Sentiment Analysis Using Convolution Neural Network*’. Penelitian analisis sentimen berbahasa India dengan data yang diperoleh dari surat kabar daring dan website. Metode *classical machine learning* seperti *Naïve Bayes*, *K-Nearest Neighbor*, *Maximum Entropy*, dan *Support Vector Machine* dikomparasikan dengan metode *deep learning* yaitu *Covolutional Neural Network* (CNN). Algoritma CNN diberikan pengaturan konfigurasi berbeda seperti banyaknya *convolutional layer*, *hidden layer*, dan *filter size* untuk 12 kali percobaan. Sedangkan dimensi *output*, *regularizer*, *dropout*, dan jumlah *epoch*

tidak diperhitungkan karena tidak menunjukkan peningkatan akurasi yang signifikan. Hasil yang diperoleh algoritma *Covolutional Neural Network* mencapai akurasi 95,4% dibandingkan metode *classical machine learning* dengan hasil akurasi tertinggi pada algoritma *Naïve Bayes* sebesar 90%.

Penelitian analisis sentimen dengan mamasukkan informasi perilaku pengguna dari (Alharbi and de Doncker, 2019) berjudul ‘*Twitter Sentiment Analysis with a Deep Neural Network: An Enhanced Approach using User Behavioral Information*’. Penelitian tersebut menerapkan algoritma *Convolutional Neural Network* dari *dataset SemEval-2016 1* dan *SemEval-2016 2*. Metode *classical machine learning* seperti *Support Vector Machine*, *Naïve Bayes*, *K-Nearest Neighbor*, *J48* digunakan untuk komparasi dari algoritma CNN. Hasil yang diperoleh algoritma CNN menunjukkan kinerja yang konsisten baik di kedua *dataset* sehingga akurasi yang dihasilkan lebih konsisten.

Analisis sentimen berbasis *Word2Vec* dilakukan oleh (Juwianto *et al.*, 2020) dalam penelitian yang berjudul ‘*Sentiment Analysis Twitter Bahasa Indonesia Berbasis Word2vec Menggunakan Deep Convolutional Neural Network*’. Penelitian tersebut menggunakan model *Word2Vec* untuk Bahasa Indonesia sebagai representasi kata dalam bentuk vektor. Penggunaan *Word2Vec* juga mempercepat proses pelatihan dan meningkatkan akurasi algoritma *Deep Convolutional Neural Network*. Hasil percobaan yang telah dilakukan menghasilkan akurasi sebesar 76,40%.

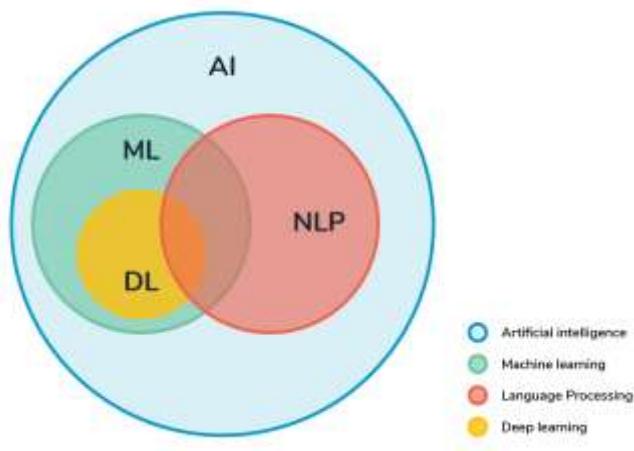
(Ratnawati and Winarko, 2018) dalam penelitian untuk tesis yang berjudul ‘*Sentiment Analysis of Movie Opinion in Twitter Using Dynamic Convolutional Neural Network Algorithm*’ melakukan pengklasifikasian terhadap sentimen positif, negatif, dan netral dari opini film berbahasa Indonesia. Metode yang digunakan yaitu *Dynamic Convolutional Neural Network* yang dibandingkan dengan metode *Naïve Bayes*. Hasil pengujian pada sistem yang dibangun memperlihatkan bahwa algoritma *Dynamic Convolutional Neural Network* memberikan hasil akurasi lebih baik dari pada metode *Naïve Bayes*, dengan nilai akurasi sebesar 85,56% sedangkan nilai akurasi yang dihasilkan *Naïve Bayes* sebesar 82,92%.

Penelitian selanjutnya dilakukan oleh (Zulfa and Winarko, 2017) berjudul “*Sentimen Analisis Tweet Berbahasa Indonesia dengan Deep Belief Network*”, melakukan pengklasifikasian terhadap kelas sentimen netral, negatif, dan positif pada data *twitter* dengan tujuan untuk mengetahui tingkat akurasi menggunakan algoritma *Deep Belief Network*. Percobaan yang dilakukan menghasilkan akurasi sebesar 93,31% dengan algortima komparasi seperti *Naïve Bayes* dengan akurasi 79,10%, dan *Support Vector Machine* dengan akurasi 92,18%.

(Nasichuddin, Adji and Widyawan, 2018) dalam penelitian yang berjudul “*Performance Improvement Using CNN for Sentiment Analysis*” melakukan peningkatan akurasi dan waktu proses pelatihan menggunakan algortima CNN. Hal ini dilakukan dengan *tuning* model pada *filter size*, *frameworks*, dan *pre-training word vector*. Hasil menunjukkan bahwa penggunaan ukuran filter yang lebih kecil dan *Word2vec* model memberikan akurasi yang lebih besar.

2.2. *Natural Language Processing*

Natural Language Processing (NLP) merupakan bidang *computer science* dan *engineering* yang telah berkembang dari studi bahasa dan komputasi linguistik dalam bidang *artificial intelligence* atau kecerdasan buatan (Pustejovsky and Stubbs, 2013). Tujuan NLP adalah merancang dan membangun aplikasi yang memfasilitasi interaksi manusia dengan mesin maupun perangkat lain melalui penggunaan bahasa alami. NLP berfokus pada pengolahan bahasa natural yaitu bahasa yang umum digunakan oleh manusia dalam berkomunikasi satu sama lain. Bahasa yang diterima oleh komputer butuh diproses dan dipahami terlebih dahulu supaya maksud dari manusia atau *user* bisa dipahami dengan baik oleh komputer. Bagian NLP dalam lingkup AI seperti yang disajikan pada gambar 2.1.



Gambar 2.1 Bagian *Natural Language Processing*

Beberapa contoh permasalahan yang dapat diselesaikan menggunakan teknologi NLP antara lain:

1. *Question Answering Systems* (QAS)

Merupakan sistem penjawab pertanyaan yang memanfaatkan pengetahuan luas untuk dapat memberikan jawaban. Program ini dapat bekerja secara otomatis tanpa harus dibelakangi operator.

2. *Text Summarization*

Suatu sistem peringkasan teks yang dapat digunakan untuk merangkum esai panjang maupun peringkasan teks dari buku yang tetap mempertahankan poin penting dari dokumen asli.

3. *Machine Translation*

Mesin penerjemah ini dapat digunakan secara otomatis mengonversi sebuah teks dari sebuah bahasa ke bahasa yang lain. Hal ini memungkinkan sebuah sistem tunggal untuk dapat menerjemahkan ke dalam beberapa bahasa.

4. *Sentiment Analysis*

Suatu sistem yang mengidentifikasi dan mengekstrak informasi subjektif dalam sumber. Informasi bisa didapatkan dari suatu ulasan produk atau pesan media sosial yang bertugas untuk mencari tahu apakah sentimen memiliki polaritas positif, netral atau negatif. Hal ini berguna untuk membantu para pembisnis dalam memahami sentimen sosial dari merek, produk atau layanan saat memantau percakapan *online* di media sosial.

5. *Speech Recognition*

Suatu sistem yang memungkinkan komputer menerima masukan berupa kata yang diucapkan. Teknologi ini memungkinkan suatu perangkat untuk mengenali dan memahami kata-kata yang diucapkan dengan digitalisasi

kata dan mencocokkan sinyal digital tersebut dengan suatu pola tertentu yang tersimpan dalam suatu perangkat.

2.3 Analisis Sentimen

Analisis sentimen adalah suatu proses ekstraksi opini dalam bentuk teks (Al-Smadi *et al.*, 2018). Informasi berbentuk teks banyak terdapat di internet dalam blog, media sosial, maupun situs berisi *review*. Dengan bantuan analisis sentimen, informasi yang tidak terstruktur dapat diubah menjadi data yang lebih terstruktur. Secara umum, analisis sentimen dibagi menjadi tiga level bagian utama (Ray and Chakrabarti, 2019), yaitu:

1. Level Dokumen

Analisis sentimen pada level dokumen diasumsikan bahwa setiap dokumen mengekspresikan pendapat tentang satu kesatuan. Dokumen dianalisis apakah memiliki orientasi positif, negatif, maupun netral.

2. Level Kalimat

Analisis sentimen pada level kalimat dapat memecah dokumen menjadi beberapa kalimat. Setiap kalimat diperlakukan sebagai entitas tunggal yang dianalisis pada suatu waktu. Level ini membedakan kalimat objektif yang mengungkapkan informasi faktual dan kalimat subjektif yang mengekspresikan pendapat.

3. Level Aspek

Analisis sentimen pada level aspek dapat mengekstraksi aspek-aspek produk dan kemudian menganalisis umpan balik atau menilai pendapat dari aspek yang diekstraksi.

2.4 *Text Pre-processing*

Bagian yang sangat penting pada analisis sentimen adalah *text pre-processing*. *Text pre-processing* merupakan langkah persiapan data yang diperlukan untuk klasifikasi (Handayani and Pribadi, 2015; Krouska, Troussas and Virvou, 2016). Hal ini bertujuan untuk menanggulangi salah mengambil ciri atau atribut yang dapat menurunkan performa analisis sentimen secara signifikan. Tahap ini melakukan seleksi data yang akan diproses pada setiap dokumen. *Text pre-processing* pada analisis sentimen adalah bagaimana tahap *pre-processing* yang dilakukan dengan tidak mengurangi makna yang terkandung dalam data. Kata yang memiliki makna penting tidak dihilang, sehingga data *input* tidak kehilangan makna sesungguhnya. Selain itu, proses menyikapi bahasa gaul dalam bahasa Indonesia harus diperhatikan karena setiap kata pada proses analisis sentimen akan menjadi fitur dalam *dataset*. Umumnya langkah *pre-processing* terdiri dari *case folding*, *tokenization*, *stopword removal*, *stemming* (Agastya, 2018). Penjelasan langkah-langkah tersebut adalah:

1. *Case Folding*

Tidak semua dokumen teks konsisten dalam penggunaan huruf kapital. Oleh karena itu, peran *case folding* dibutuhkan dalam mengonversi keseluruhan teks dalam dokumen menjadi suatu bentuk standar atau *lower case*. *Case*

folding berperan untuk mengubah semua huruf dalam dokumen menjadi huruf kecil. Hanya huruf ‘a’ sampai dengan ‘z’ yang diterima. Karakter selain huruf dihilangkan dan dianggap delimiter seperti tanda koma, titik koma atau titik dua. Proses *case folding* tidak memerlukan *library* tertentu, karena dapat menggunakan fungsi *lower()* yang merupakan bawaah dari bahasa pemrograman *Python*.

2. *Tokenization*

Suatu dokumen teks terdiri dari sekumpulan kalimat dimana proses *tokenization* memecah dokumen tersebut menjadi bagian-bagian kata yang disebut token. Setelah melalui proses *tokenization* dapat diketahui jumlah kemunculan setiap token. Contoh proses *tokenization* seperti yang disajikan pada tabel 2.1.

Tabel 2.1 Contoh Proses *Tokenization*

Teks Input	Teks Ouput
Sistem penjawab pertanyaan memanfaatkan pengetahuan luas untuk dapat memberikan jawaban	sistem penjawab pertanyaan memanfaatkan pengetahuan luas untuk dapat memberikan jawaban

3. *Stopword Removal*

Tahap *stopword removal* bertujuan untuk mengurangi jumlah kata dalam dokumen yang berpengaruh dalam kecepatan maupun performa pada *Natural Language Processing*. Tahap ini mengambil kata-kata yang dianggap penting dan membuang kata-kata yang dianggap tidak penting dari hasil *tokenization*. *Stopword* merupakan kata yang diabaikan dalam pemrosesan yang disimpan dalam *stoplists*. Karakteristik utama dalam pemilihan *stopword* adalah kata yang mempunyai frekuensi kemunculan tinggi seperti kata penghubung “dan”, “atau”, “tapi”, “akan”. Tidak ada aturan pasti dalam menentukan *stopword* yang digunakan, penentuan *stopword* dapat disesuaikan dengan kasus yang diselesaikan.

4. *Stemming*

Tahap *stemming* bertujuan untuk mentransformasikan kata menjadi kata dasar atau *root word* dengan menghilangkan semua imbuhan kata meliputi awalan kata (*prefixes*), sisipin kata (*infixes*), akhiran kata (*suffixes*), dan menghilangkan awalan serta akhiran kata (*confixes*) pada kata turunan. Salah satu *library* yang bisa digunakan dalam melakukan proses *stemming* Bahasa Indonesia adalah menggunakan *Library Python Sastrawi*.

Selain melakukan pembersihan data maka dilakukan *balancing dataset*. Salah satu masalah umum yang ditemukan dalam kumpulan data untuk klasifikasi adalah persebaran data yang tidak seimbang. Persebaran data yang tidak seimbang dapat menyebabkan kurang tepatnya model yang dibuat pada saat *training data* serta algoritma klasifikasi memiliki kinerja yang buruk (Sonak and Patankar, 2015).

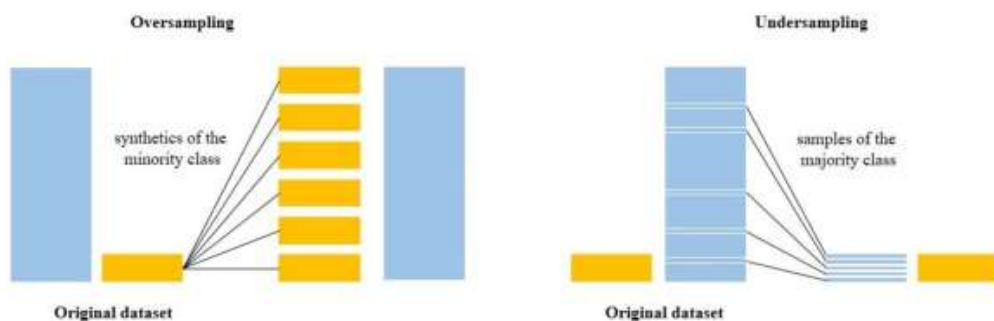
Untuk menangani masalah ketidakseimbangan data dapat dilakukan dengan teknik *resampling* seperti *oversampling* dan *undersampling* (Ganganwar, 2012), seperti yang disajikan pada gambar 2.2.

1. *Oversampling*

Oversampling adalah teknik pengambilan sampel yang menyeimbangkan kumpulan data dengan mereplikasi kelas minoritas. Keuntungan dari metode ini adalah tidak ada kehilangan data sedangkan kerugian dari teknik ini dapat menyebabkan pemasangan yang berlebihan dan dapat menyebabkan *overhead* komputasi tambahan.

2. *Undersampling*

Metode *undersampling* dilakukan menggunakan *subset* dari kelas mayoritas untuk melatih *classifier* dengan menghapus kelas mayoritas.



Gambar 2.2 Ilustrasi *Oversampling* dan *Undersampling*

Sumber: (Xia *et al.*, 2019)

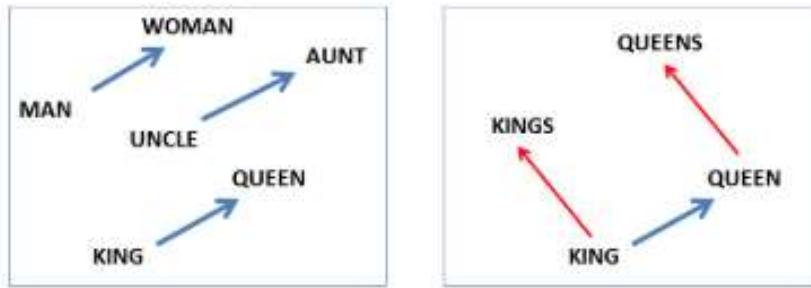
2.5 *Feature Extraction*

Suatu objek dalam melakukan pembelajaran memerlukan ciri-ciri dari objek tersebut atau dikenal sebagai *feature*. Apabila objek yang diterima berupa teks,

maka diperlukan proses ekstraksi menjadi vektor numerik yang merepresentasikan setiap kata karena komputer tidak dapat mengolah data selain data berbentuk angka. Oleh karena itu dibutuhkan langkah untuk mengekstrak kata menjadi numerik dalam proses analisis sentimen (Guia, Silva and Bernardino, 2019).

Metode *feature extraction* dilakukan setelah tahap *pre-processing* sebagai tahap awal untuk merepresentasikan data secara menyeluruh. Metode *feature extraction* yang efektif menuntun pada peningkatan performa model yang lebih baik. Bebagai jenis *feature* diambil dari data, kemudian dilakukan pembelajaran secara khusus pada *feature* tersebut. *Word vector representation* merupakan hasil pembelajaran dari algoritma *deep learning*. Setiap kata memiliki vektor yang mewakili makna dari kata tersebut dengan dimensi vektor yang bervariasi. *Word vector representation* juga dikenal sebagai *word embedding*.

Salah satu metode dari *word embedding* adalah *Word2Vec*. *Word2vec* merupakan *word vector representation* yang dibuat oleh Google (Google, 2013). Kelebihan model *Word2Vec* adalah mampu merepresentasikan fitur sebagai vektor yang padat dibandingkan dengan model konvensional. Dalam mempresentasikan suatu kata, *Word2Vec* mengimplementasi *neural network* untuk menghitung *contextual and semantic similarity* (kesamaan kontekstual dan makna) dari setiap inputan kata yang berbentuk *one-hot encoded vectors*. Hasil dari *contextual and semantic similarity* dapat merepresentasikan relasi suatu kata dengan kata lainnya, seperti relasi antara ‘*Male - Female*’, relasi pada ‘*Verb tense*’, dan bahkan relasi pada ‘*Country - Capital*’, seperti yang diilustrasikan pada gambar 2.3 (Mikolov, Yih and Zweig, 2013).



Gambar 2.3 *Contextual and Semantic Similarity*

Sumber : (Mikolov, Yih and Zweig, 2013)

Selain itu, model *word2vec* dapat mengatasi permasalahan sinonim dan homonim yang ada sering ditemukan dalam bidang *Natural Language Processing* (Nawangsari, Kusumaningrum and Wibowo, 2019). *Word2Vec* memiliki dua model arsitektur yaitu *Skip-Gram* dan *Continous Bag of Words* (CBOW) (Mikolov *et al.*, 2013) seperti yang disajikan pada gambar 2.4 dan 2.5. Kedua metode ini menggunakan konsep jaringan saraf tiruan yang memetakan kata ke variabel target yang merupakan sebuah kata. Tujuan dalam arsitektur skip-gram adalah untuk memprediksi kata yang ada di sekitar *current word*. Sedangkan arsitektur CBOW digunakan untuk memprediksi kata yang ada pada sekitar kata tersebut. Berikut adalah ilustrasi arsitektur skip-gram dan CBOW dengan *window size* adalah 2 dan *current word* atau input “universitas”.

“*Jurusan teknik elektro universitas negeri semarang*”

jurusank	: [1,0,0,0,0,0]
teknik	: [0,1,0,0,0,0]
elektro	: [0,0,1,0,0,0]
universitas	: [0,0,0,1,0,0]

negeri : [0,0,0,0,1,0]

Semarang : [0,0,0,0,0,1]

Data *input* berbentuk *one-hot encoded vector*, berikut sebagai proses dari *forward-backward* proses *training* arsitektur *skip-gram*:

- a. Jurusan Teknik Elektro Universitas Negeri Semarang

Input, target = (*jurusan*, *teknik*); (*jurusan*, *elektro*)

- b. Jurusan Teknik Elektro Universitas Negeri Semarang

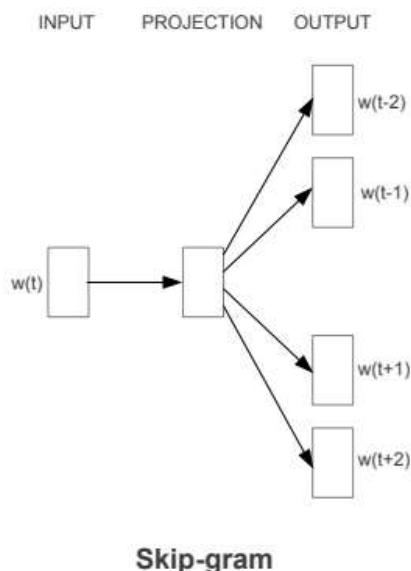
Input, target = (*teknik*, *jurusan*); (*teknik*, *elektro*); (*teknik*, *universitas*)

- c. Jurusan Teknik Elektro Universitas Negeri Semarang

Input, target = (*elektro*, *jurusan*); (*elektro*, *teknik*); (*elektro*, *universitas*); (*elektro*, *negeri*)

- d. Jurusan Teknik Elektro Universitas Negeri Semarang

Input, target = (*universitas*, *teknik*); (*universitas*, *elektro*); (*universitas*, *negeri*); (*universitas*, *semarang*)

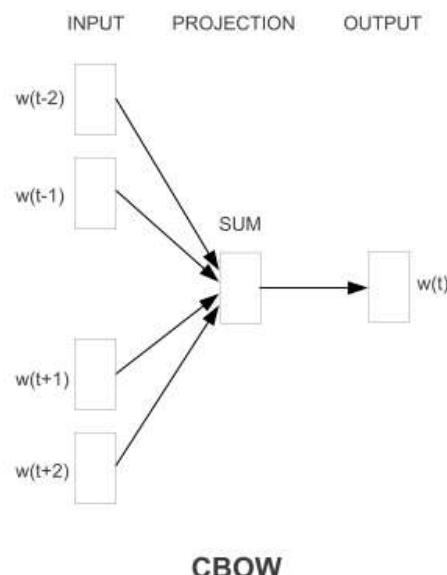


Gambar 2.4 Arsitektur *Skip-Gram*

Sumber: (Mikolov *et al.*, 2013)

Berikut adalah illustrasi dari arsitektur CBOW yang merupakan kebalikan dari arsitektur skip-gram, dengan memprediksi kata yang kosong:

- a. **Jurusan Teknik Elektro Universitas Negeri Semarang**
Input, target = (teknik, jurusan); (elektro, jurusan)
- b. **Jurusan Teknik Elektro Universitas Negeri Semarang**
Input, target = (jurusan, teknik); (elektro, teknik); (universitas, teknik)
- a. **Jurusan Teknik Elektro Universitas Negeri Semarang**
Input, target = (jurusan, elektro); (teknik, elektro); (universitas, elektro);
(negeri, elektro)
- b. **Jurusan Teknik Elektro Universitas Negeri Semarang**
Input, target = (teknik, universitas); (elektro, universitas); (negeri, universitas);
(semarang, universitas)



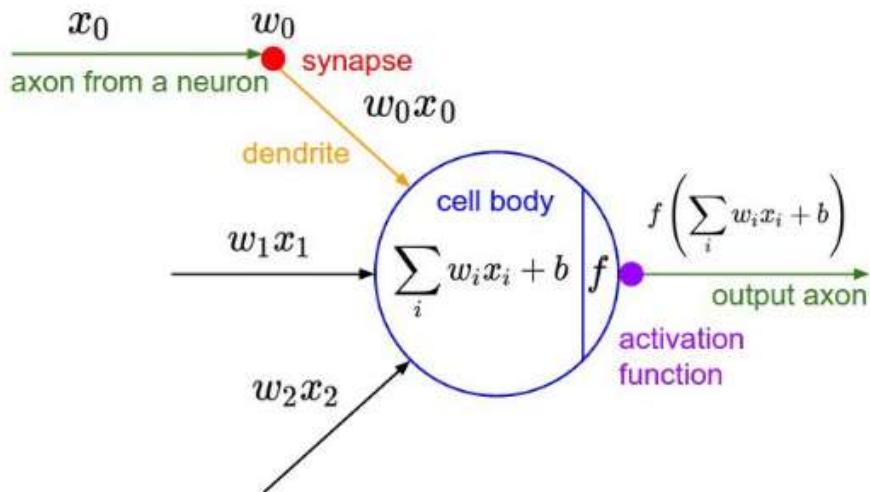
Gambar 2.5 Arsitektur CBOW

Sumber: (Mikolov *et al.*, 2013)

2.6 Artificial Neural Network (ANN)

Neural network atau jaringan saraf tiruan merupakan suatu model yang terinspirasi dari cara kerja *neuron* pada otak manusia. Setiap *neuron* yang ada pada otak manusia saling berhubungan menghasilkan informasi dari setiap *neuron* tersebut. *Neuron* tersusun dari tiga bagian penting yaitu badan sel sebagai jantung *neuron* yang mengarahkan semua aktivitas, *dendrit* untuk mengirimkan *impuls* yang diterima ke badan sel saraf, dan *akson* yang mengantarkan *impuls* dari badan sel ke jaringan lain seperti illustrasi pada gambar 2.6. Proses yang terjadi pada otak manusia adalah sebuah *neuron* menerima *impuls* dari *neuron* lain melalui *dendrit* kemudian mengirimkan sinyal yang dihasilkan oleh badan sel melalui *akson*. *Akson* dari sel saraf ini bercabang-cabang dan berhubungan dengan *dendrit* dari sel saraf lain dengan mengirimkan *impuls* melalui *sinapsis*.

Berdasarkan struktur *neuron* dan proses kerja pada otak manusia, maka konsep dasar pembangunan *Artificial Neural Network* (ANN) terbentuk. ANN terdiri dari *neuron* yang berisi *adder* dan fungsi aktivasi yang berisi sejumlah bobot atau sinapsis apabila dalam otak manusia dan berisi sejumlah vektor masukan atau *dendrit* apabila di dalam otak manusia. Setiap *neuron* menerima input dan melakukan operasi *dot* dengan sebuah *weight*, menjumlahkannya (*weighted sum*) dan menambahkan bias. Hasil dari operasi ini menjadi parameter dari *activation function* yang dijadikan sebagai output dari *neuron* tersebut.



Gambar 2.6 Ilustrasi *Neuron* dengan Model Matematis

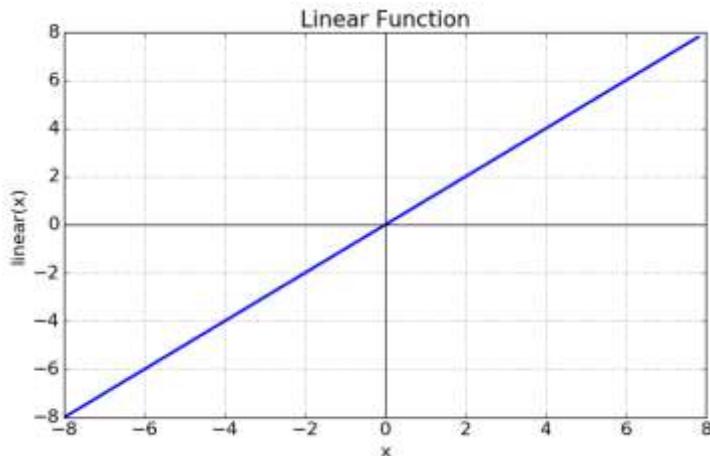
Sumber: (Alfarizi, H.M., 2020)

2.6.1 Activation Function

Activation function adalah persamaan matematika yang menentukan *output* dari jaringan saraf yang digunakan untuk menentukan apakah suatu *neuron* harus aktif atau tidak berdasarkan *weighted sum* dari *input*. *Activation function* juga membantu normalisasi *output* setiap *neuron* dengan kisaran antara 1 dan 0 atau antara -1 dan 1. Secara umum terdapat 2 jenis *activation function* yaitu *linear* dan *non linear* yang diuraikan sebagai berikut:

1. Linear Activation Function

Standar *activation function* dari sebuah *neuron* adalah *linear*. Jika sebuah *neuron* menggunakan *linear function*, maka *output* dari *neuron* tersebut adalah *weighted sum* dari *input* dengan bias. *Linear activation function* lebih baik dari *step function* karena memungkinkan adanya beberapa *output* yang artinya *output* tidak hanya ya dan tidak. *Linear activation function* seperti yang disajikan pada gambar 2.7.



Gambar 2.7 *Linear Activation Function*

Linear activation function memiliki dua masalah utama yaitu tidak memungkinkan adanya penggunaan *backpropagation (gradient descent)* untuk pelatihan modelnya karena turunan dari *linear activation function* sifatnya konstan sehingga tidak memiliki hubungan dengan *input*. Kedua, pada *linear activation function* tidak memperhitungkan berapa banyak *layer* dalam *neural network*, *layer terakhir* menjadi *linear function* dari *layer* pertama. Hal ini karena karena kombinasi *linear* dari *linear function* masih berupa fungsi *linear* sehingga hanya mengubah satu *layer neural network*.

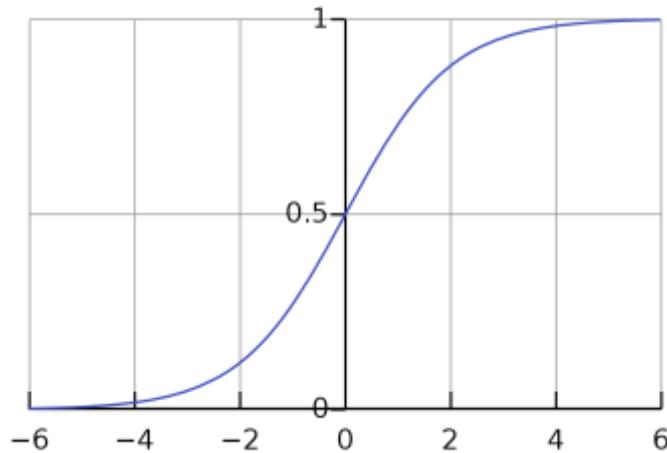
2. *Non Linear Activation Function*

Nural Network yang lebih modern menggunakan *non linear activation function*. *Non linear activation function* memungkinkan suatu model untuk membuat *mapping* yang lebih kompleks antara jaringan *input* dan *output* seperti pada gambar, video, audio, dan set data *non linear* atau memiliki dimensi tinggi. *Non linear activation function* mengatasi masalah dari *linear function* seperti memungkinkan penggunaan *backpropagation* karena *non*

linear activation function memiliki fungsi turunan yang berhubungan dengan *input*. Fungsi ini juga memungkinkan penggunaan *multiple layer* untuk membuat jaringan saraf yang lebih dalam. Oleh karena itu diperlukan *hidden layer* untuk mempelajari kumpulan data yang kompleks dengan tingkat akurasi tinggi. *Non linear activation function* yang umum digunakan diantaranya:

a. *Sigmoid*

Sigmoid menerima angka tunggal dan mengubah nilai x menjadi sebuah nilai yang memiliki rentang 0 sampai 1. Kelebihan yang dimiliki *sigmoid* adalah nilai *output* terikat antara 0 dan 1 sehingga menormalkan *output* masing-masing *neuron*. Tetapi *sigmoid* juga memiliki kelemahan yang dapat mematikan *gradient*, ketika *activation* dari *neuron* mengeluarkan nilai yang berada pada 0 atau 1, dimana *gradient* di wilayah ini hampir nol. Karena itu, jika *gradient* memiliki nilai yang sangat kecil, *sigmoid* akan mematikan *gradient*. Selain itu, *output* dari *sigmoid* tidak *zero-centered*. Hal ini berimplikasi pada kedinamisan saat melakukan *gradient descent*, karena apabila data yang datang ke *neuron* selalu positif maka *gradient* pada bobot selama proses *backpropagation* menjadi semua positif atau semua negatif. Hal ini dapat mengganggu proses *training*. *Sigmoid activation function* seperti yang diilustrasikan pada gambar 2.8.



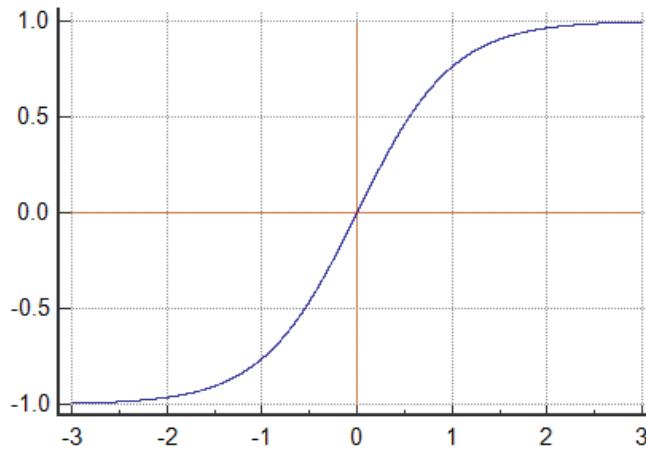
Gambar 2.8 *Sigmoid Function*

Sigmoid memiliki bentuk formula sebagai berikut:

$$f(x) = \frac{1}{1+e^{-x}} \quad (2.1)$$

b. *Tanh*

Tanh function mengubah nilai input x menjadi nilai yang memiliki rentang mulai dari -1 sampai 1. *Tanh* memiliki kekurangan seperti *sigmoid* yaitu dapat bisa mematikan *gradient* tetapi, kelebihannya adalah output yang dimiliki *tanh* merupakan *zero-centered*. *Tanh activation function* seperti yang diilustrasikan pada gambar 2.9.



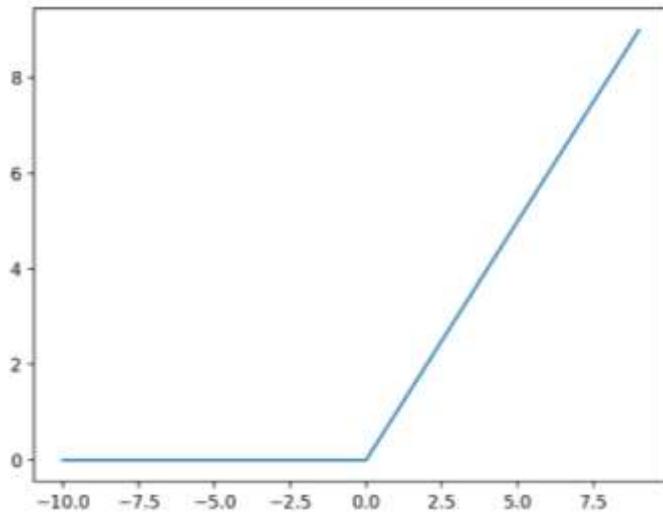
Gambar 2.9 *Tanh Function*

Tanh merupakan pengembangan dari *sigmoid* yang memiliki bentuk persamaan 2.2.

$$f(x) = \tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.2)$$

c. ReLU

ReLU atau *Rectified Linear Unit* pada dasarnya membuat pembatas pada bilangan nol, artinya apabila $x \leq 0$ maka $x = 0$ dan apabila $x > 0$ maka $x = x$. ReLU dapat mempercepat proses konvergensi yang dilakukan dengan *stochastic gradient descent* apabila dibandingkan dengan *sigmoid* maupun *tanh*. Tetapi, ReLU lemah pada saat proses *training* dan dapat membuat unit tersebut mati apabila *learning rate* yang diinisialisasi terlalu tinggi. ReLU *activation function* seperti yang diilustrasikan pada gambar 2.10.



Gambar 2.10 *ReLU Function*

ReLU memiliki bentuk formula sebagai berikut:

$$f(x) = \max(0, x) \quad (2.3)$$

d. *Softmax*

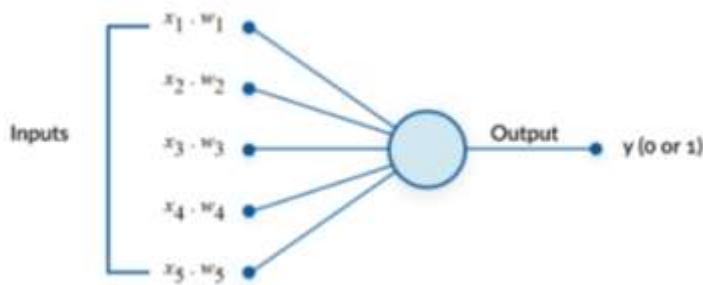
Softmax activation function menghitung probabilitas untuk masing-masing kelas atas semua kelas yang ada. Rentang probabilitas *output* dari fungsi aktivasi ini memiliki nilai 0 hingga 1 dan jumlah probabilitas adalah satu. Apabila *softmax* digunakan untuk model multiklasifikasi, *softmax* mengembalikan peluang dari setiap kelas dan kelas target memungkinkan memiliki probabilitas yang tinggi.

Softmax memiliki bentuk formula sebagai berikut, dimana $i = 0, 1, 2, \dots, k$:

$$f(x_i) = \frac{\exp(x_i)}{\sum_{j=0}^k \exp(x_j)} \quad (2.4)$$

2.6.2 Arsitektur Neural Network

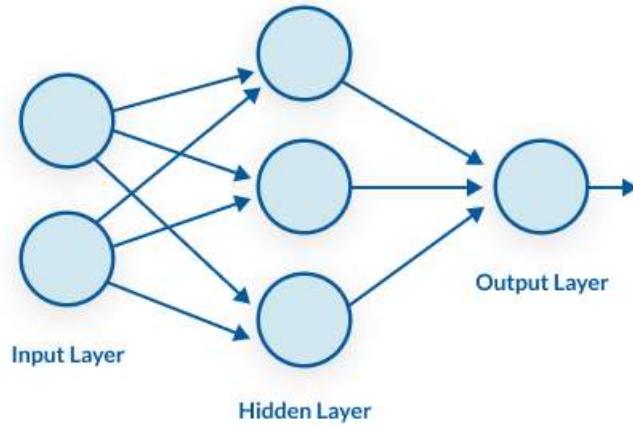
Arsitektur atau struktur *neural network* adalah gambaran susunan komponen layer dan *neuron* pada *input*, *hidden* dan *output* yang terhubung dengan *weight* atau bobot, *activation function* dan *learning function*. *Perceptron* dan *Multilayer Perceptron* adalah dasar dari jaringan saraf tiruan. Sebuah *perceptron* adalah algoritma klasifikasi biner yang dimodelkan setelah berfungsinya otak manusia, hal ini dimaksudkan untuk meniru *neuron*. Meskipun *perceptron* memiliki struktur sederhana tetapi memiliki kemampuan untuk belajar dan menyelesaikan masalah yang sangat kompleks. *Neural Network* yang paling populer adalah jaringan *multi perceptron feed-forward* yang dilatih melalui algoritma *backpropagation* (Ranjit et al., 2018).



Gambar 2.11 *Perceptron Input* dan *Ouput*

Sumber: missinglink.ai

Multilayer Perceptron (MLP) adalah sekelompok *perceptron* yang disusun dalam beberapa lapisan yang dapat menyelesaikan permasalahan kompleks. Setiap *perceptron* di lapisan pertama atau *input layer* mengirimkan sinyal ke semua *perceptron* di lapisan kedua atau *hidden layer*, dan berlangsung seterusnya. MLP berisi *input layer*, minimal satu *hidden layer*, dan *output layer*.



Gambar 2.12 Arsitektur *Neural Network*

Sumber: *missinglink.ai*

Pada Gambar 2.12, tahap *training* untuk setiap *weight* dan bias setiap *neuron* diupdate secara terus menerus hingga menghasilkan *output* sesuai dengan harapan. Pada setiap iterasi dilakukan proses *evaluation* yang digunakan untuk menentukan kapan harus menghentikan proses *training* (*stopping point*). Proses training terdiri dari 2 tahap yaitu *Forward Propagation* dan *Backward Propagation*:

a. *Forward Propagation*

Forward pass atau disebut *forward propagation* merupakan proses membawa data pada *input* melewati seiap *neuron* pada *hidden layer* sampai *output layer* yang akan dihitung *error*-nya. Persamaan *forward pass* pada Gambar 2.12 dapat dilihat pada formula 2.5. Dimana i adalah *node* pada *input layer* (2 *node input*), j adalah *node* pada *hidden layer*, dan h adalah *output* dari *node* di *hidden layer*.

$$\text{dot } j = \sum_i^2 w_{ji} x_i + b_j \quad (2.5)$$

Pada saat *forward pass*, *input* di-*propagate* menuju *output layer* dan hasil prediksi *output* akan dibandingkan dengan target menggunakan fungsi yang disebut dengan *Loss Function*. *Loss function* digunakan untuk mengukur performa dari *neural network* dalam melakukan prediksi terhadap target. Untuk menghitung loss seperti persamaan (2.6).

$$\text{Loss} = (\text{Target} - \text{Prediction})^2 \quad (2.6)$$

b. *Backward Propagation*

Nilai *error* yang diperoleh dari tahap *forward propagation* digunakan untuk *update* setiap *weight* dan bias dengan *learning rate* tertentu. Proses tersebut dilakukan secara berulang-ulang sampai didapatkan nilai *weight* dan bias yang memberikan nilai *error* sekecil mungkin pada *output layer*. Tahapan dari *backward pass* atau *backpropagation* adalah dengan menghitung *gradient* dari *loss function* terhadap semua parameter yang ada dengan mencari turunan parsial dan tahap kedua dengan melakukan *update* semua parameter *weight* dan bias menggunakan *Stochastic Gradient Descent* (SGD) dengan menambahkan atau mengurangi nilai *weight* lama dengan sebagian *learning rate* dari nilai *gradient* yang telah didapatkan.

Turunun ReLU *activation function* adalah seperti persamaan 2.7 dan 2.8.

$$y = \max(0, x) \quad (2.7)$$

$$\frac{\partial y}{\partial x} = \begin{cases} 1 & x > 0 \\ 0 & x \leq 0 \end{cases} \quad (2.8)$$

Turunun Sigmoid *activation function* adalah seperti persamaan 2.9 dan 2.10.

$$y = \frac{1}{1+e^{-x}} \quad (2.9)$$

$$\frac{\partial y}{\partial x} = \frac{1}{1+e^{-x}} x \left(1 - \frac{1}{1+e^{-x}}\right) \quad (2.10)$$

Turunun *Linear activation function* adalah seperti persamaan 2.11 dan 2.12.

$$y = x \quad (2.11)$$

$$\frac{\partial y}{\partial x} = 1 \quad (2.12)$$

Setelah jaringan saraf didefinisikan dengan bobot awal dan tahap *forward pass* digunakan untuk prediksi awal, tahap *backpropagation* digunakan untuk menyesuaikan kembali tiap *weight* dan bias berdasarkan *error* yang didapatkan pada tahap *forward pass*. Algoritma *backpropagation* digunakan pada proses pelatihan yang memiliki kinerja baik bahkan dengan data yang kompleks (Setti and Wanto, 2018).

2.7 Convolutional Neural Network

Convolutional Neural Network (CNN) adalah jaringan saraf *multilayer* jenis *feed forward network* yang terdiri dari dua atau lebih *deep layer* dan kemudian dihubungkan dengan *fully connected layer* seperti *multilayer neural network*. Secara garis besar CNN tidak jauh berbeda dengan *neural network* biasanya, terdiri dari *neuron* yang memiliki *weight*, bias, dan *activation function*. Kemampuan utama dari CNN adalah arsitektur yang dapat mengenali informasi prediktif dari suatu objek seperti teks, potongan suara, gambar. Dalam CNN tidak perlu beberapa langkah yang dilakukan pada ANN karena pada CNN menggunakan operasi konvolusi melalui *input layer* untuk menghitung *output*. Pada masing-masing *layer*

memiliki *filter* yang berbeda lalu mengkombinasikan hasil operasi konvolusi tersebut.

CNN merupakan salah satu penerapan *deep learning* model yang berbeda dengan metode *machine learning*. Dilihat dari waktu eksekusi yang dilakukan, model-model *deep learning* membutuhkan waktu yang lebih lama dibandingkan *machine learning* karena penghitungan bobot yang besar dan penambahan parameter. Selain itu, dari segi *hardware* membutuhkan spesifikasi yang lebih tinggi untuk menghitung operasi matriks dalam jumlah besar. *Machine learning* cocok untuk data yang kecil atau data yang jumlahnya tidak terlalu banyak, sedangkan *deep learning* model bagus diterapkan untuk data yang lebih besar. Kontribusi utama dari algoritma CNN adalah pada *convolutional* dan *pooling layer*.

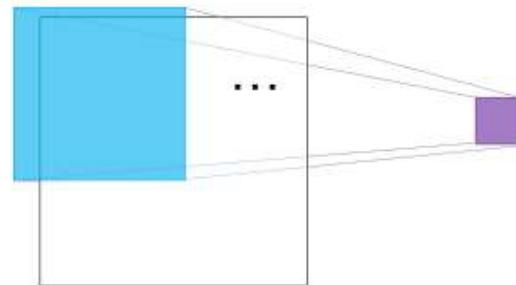
2.7.1 Layer pada Convolutional Neural Network

CNN dibagi menjadi 2 bagian besar yaitu *feature extraction layer* dan *fully-connected layer*. *Feature extraction layer* terdiri dari dua bagian yaitu *convolutional layer* dan *pooling layer*.

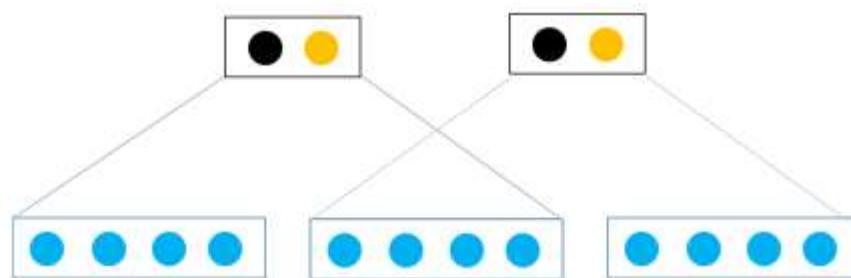
a. Convolutional Layer

Convolutional layer menggunakan prinsip *sliding window* dan *weight sharing* yang bertujuan untuk mengurangi kompleksitas perhitungan. Operasi pada *window* digunakan untuk mengetahui aspek lokal yang paling informatif atau dikenal dengan *filter* yang dapat dikenali oleh *window*. Seperti yang disajikan pada gambar 2.13, warna biru merepresentasikan

window dan warna ungu merepresentasikan *filter*. *Window* digeser sebanyak T yang dapat menghasilkan vektor dengan ukuran tertentu. *Window* ditransformasikan menjadi suatu nilai numerik. Dalam CNN 1 dimensi atau CNN 1-D disajikan seperti gambar 2.14, lingkaran warna biru merepresentasikan *feature vector* dari suatu *input*. Setiap dua *input* ditransformasikan menjadi dua dimensi atau *channel* yang dapat menghasilkan vektor dengan 4 dimensi ($2 \text{ window} \times 2$). Suatu *input* x dapat menggunakan *stride* sebesar s untuk menentukan seberapa banyak data dapat digeser untuk *window* baru.



Gambar 2.13 *Sliding Window*



Gambar 2.14 1D *Convolution*

b. *Pooling Layer*

Pooling layer berada setelah *convolutional layer* yang digunakan untuk merangkum informasi yang dihasilkan dari *convolutional layer*. Vektor-vektor yang dihasilkan dikombinasikan (*pooled*) menjadi vektor baru. *Pooling* yang biasa digunakan adalah *max pooling* dan *average pooling* yang diilustrasikan seperti gambar 2.15. *Max pooling* mengambil nilai terbesar dari lingkaran hitam dan mengambil nilai terbesar dari lingkaran *orange*. *Average pooling* mengambil nilai rata-rata dari lingkaran hitam dan rata-rata lingkaran *orange* untuk menghasilkan nilai baru. Tujuan dari penggunaan *pooling layer* adalah untuk mengurangi dimensi dari *feature map* (*downsampling*), sehingga mempercepat komputasi karena parameter yang harus diupdate semakin sedikit dan mengatasi *overfitting*.



Gambar 2.15 *Max Pooling* dan *Average Pooling*

c. *Fully Connected Layer*

Setelah melalui *convolutional layer* dan *pooling layer*, *feature map* yang dihasilkan masih berbentuk multidimensional *array*, sehingga perlu dilakukan *flatten* atau *reshape feature map* menjadi sebuah vektor agar bisa kita gunakan sebagai input dari *fully connected layer*. *Fully connected layer*

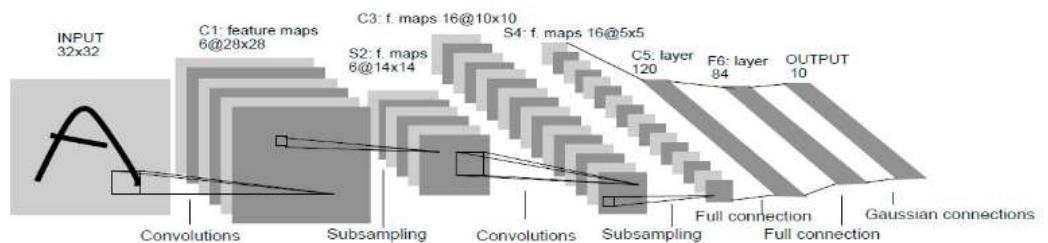
sama dengan *multi layer perceptron* yang memiliki *hidden layer*, *activation function*, *output layer*, dan *loss function*.

2.7.2 Arsitektur Convolutional Neural Network

Berbagai arsitektur telah dikembangkan dan diimplementasikan pada CNN, beberapa diantaranya seperti *LeNet-5 Architecture*, *AlexNet Architecture*, *GoogleNet Architecture* (Indolia *et al.*, 2018).

a. LeNet5 Architecture

Model LeNet-5 dikembangkan untuk mengidentifikasi angka tulisan tangan untuk pengenalan kode pos pada layanan pos besar yang memperkenalkan model CNN seperti saat ini. *LeNet5 Architecture* seperti yang ditunjukkan pada gambar xx memiliki delapan *layer* yang terdiri dari lima *convolutional layer* dan tiga *fully connected layer*.

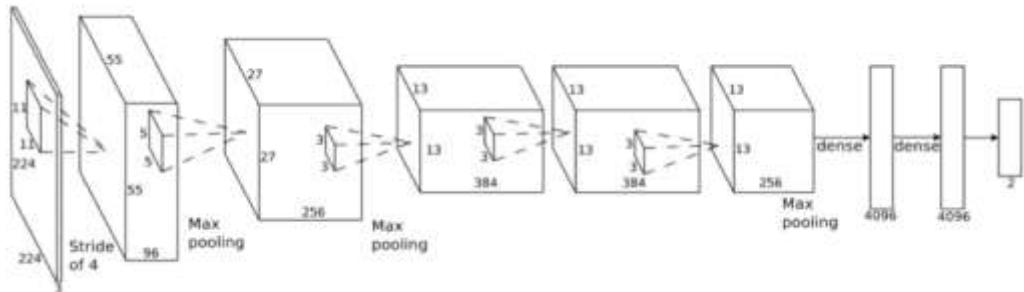


Gambar 2.16 LeNet-5 Architecture

Sumber: (Lecun *et al.*, 1998)

b. AlexNet Architecture

AlexNet Architecture mirip dengan *LeNet-5 Architecture*, meskipun model ini jauh lebih besar dan menampilkan *convolutional* yang bertumpuk.

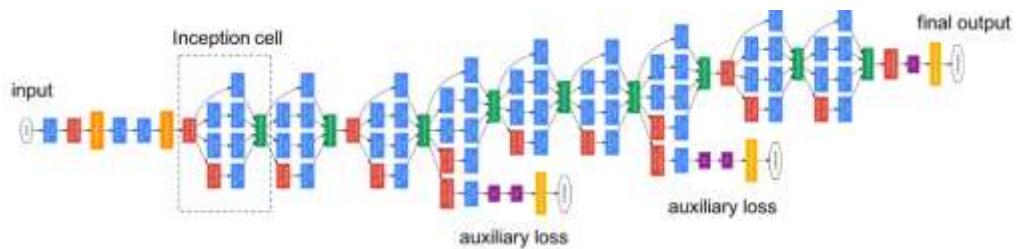


Gambar 2.17 AlexNet Architecture

Sumber: (Krizhevsky, Sutskever and Hinton, 2007)

c. GoogleNet Architecture

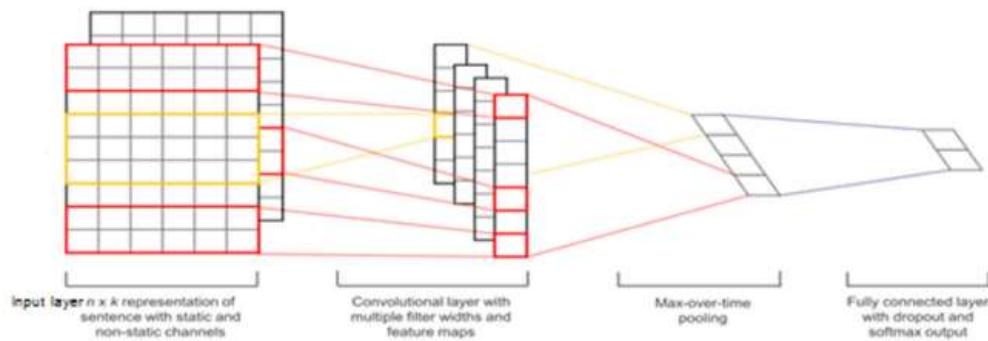
GoogLeNet architecture atau dikenal dengan *inception module* ini merupakan kontribusi utama dari pengembangan modul *inception*. Arsitektur ini berjalan secara paralel dengan ukuran reseptif yang berbeda dengan *learning rate* 6,67%.



Gambar 2.18 GoogleNet Architecture

Dalam perspektif analisis sentimen, CNN bekerja pada proses di mana setiap kata diberi bobot pada *hidden layer*. Selanjutnya setiap kata diperiksa untuk kecocokan kata tersebut secara berulang (Swathi Lakshmi, Sini Raj and Raj Vikram, 2017). Salah satu keuntungan yang dimiliki CNN dibandingkan dengan algoritma klasifikasi lain adalah bahwa CNN membutuhkan parameter dan

pengawasan lebih sedikit. Lapisan pertama yang ada pada CNN adalah *layer input* yang mewakili kalimat lebih dari dimensi $n \times k$, kedua adalah *convolutional layer*, selanjutnya *global max pooling layer*, dan terakhir *fully connected layer* yang menghasilkan *output* seperti yang ditunjukkan pada Gambar 2.19.



Gambar 2.19 Arsitektur CNN untuk Analisis Sentimen

Sumber: (Swathi Lakshmi, Sini Raj and Raj Vikram, 2017)

BAB V

SIMPULAN DAN SARAN

5.1 Kesimpulan

Penelitian mengenai analisis sentimen *twitter* menggunakan *dataset* umum berbahasa Indonesia dengan algoritma *Convolutional Neural Network* (CNN) telah berhasil dilakukan. Eksperimen dilakukan dengan 12 variasi model CNN menggunakan parameter yang berbeda pada *convolutional layer*, *filter size*, dan jumlah filter. Hasil penelitian menunjukkan bahwa model CNN dengan performa terbaik pada CNN 12 dengan tingkat *accuracy* 81,4%, sedangkan model dengan performa paling buruk pada CNN 1 dengan tingkat *accuracy* 70,5%. Algoritma CNN dibandingkan dengan algoritma SVM yang menghasilkan *accuracy* 61,4%, KNN menghasilkan *accuracy* 52,3%, dan GSD menghasilkan *accuracy* 62,1%. Algoritma CNN menghasilkan performa *accuracy* yang lebih baik dibandingkan *machine learning*.

5.2 Saran

Beberapa saran dapat dipertimbangkan dalam mengembangkan penelitian lebih lanjut adalah sebagai berikut.

1. Penambahan jumlah *dataset* yang digunakan untuk proses *training* dan *testing* untuk mendapatkan fitur lebih banyak. Hal ini berpengaruh pada tingkat *accuracy*, *precision*, *recall*, dan *f1-score*.

2. Pembuatan model CNN dengan variasi parameter berbeda lainnya untuk mengetahui perbedaan performa secara signifikan.
3. Penggunaan *word vector representation* dengan dimensi berbeda untuk mengetahui performa model yang dihasilkan.
4. Pengembangan analisis sentimen selanjutnya dapat menggunakan algoritma *deep learning* lain seperti Recurrent Neural Networks (RNN) maupun *Long Short-Term Memory* (LSTM), sebagai komparasi dari algortima *Convolutional Neural Network* (CNN).

DAFTAR PUSTAKA

- Agastya, I. M. A. (2018) ‘Pengaruh Stemmer Bahasa Indonesia Terhadap Peforma Analisis Sentimen Terjemahan Ulasan Film’, *Jurnal Tekno Kompak*, 12(1), p. 18. doi: 10.33365/jtk.v12i1.70.
- Al-Smadi, M. et al. (2018) ‘Deep Recurrent Neural Network vs. Support Vector Machine for Aspect-Based Sentiment Analysis of Arabic Hotels’ Reviews’, *Journal of Computational Science*. Elsevier B.V., 27, pp. 386–393. doi: 10.1016/j.jocs.2017.11.006.
- Alharbi, A. S. M. and de Doncker, E. (2019) ‘Twitter Sentiment Analysis with a Deep Neural Network: An Enhanced Approach using User Behavioral Information’, *Cognitive Systems Research*. Elsevier B.V., 54, pp. 50–61. doi: 10.1016/j.cogsys.2018.10.001.
- Anastasia, S. and Budi, I. (2016) ‘Twitter Sentiment Analysis of Online Transportation Service Providers’, in *International Conference on Advanced Computer Science and Information Systems, ICACSIS 2016*, pp. 359–365. doi: 10.1109/ICACSIS.2016.7872807.
- Anwar Hridoy, S. A. et al. (2015) ‘Localized twitter opinion mining using sentiment analysis’, *Decision Analytics*. Springer Berlin Heidelberg, 2(1). doi: 10.1186/s40165-015-0016-4.
- Ferdiana, R. et al. (2019) ‘Dataset Indonesia untuk Analisis Sentimen’, *Jurnal Nasional Teknik Elektro dan Teknologi Informasi (JNTETI)*, 8(4), pp. 334–339. doi: 10.22146/jnteti.v8i4.533.
- Fitri, V. A., Andreswari, R. and Hasibuan, M. A. (2019) ‘Sentiment Analysis of Social Media Twitter with Case of Anti-LGBT Campaign in Indonesia using Naïve Bayes, Decision Tree, and Random Forest Algorithm’, in *Procedia Computer Science*. Elsevier B.V., pp. 765–772. doi: 10.1016/j.procs.2019.11.181.

- Ganganwar, V. (2012) ‘An Overview of Classification Algorithms for Imbalanced Datasets’, *International Journal of Emerging Technology and Advanced Engineering*, 2(4), pp. 42–47. Available at: http://www.ijetae.com/files/Volume2Issue4/IJETAE_0412_07.pdf.
- Google (2013) *Word2vec*. Available at: <https://code.google.com/archive/p/word2vec/> (Accessed: 17 September 2020).
- Guia, M., Silva, R. R. and Bernardino, J. (2019) ‘Comparison of Naïve Bayes, Support Vector Machine, Decision Trees and Random Forest on Sentiment Analysis’, in *IC3K 2019 - Proceedings of the 11th International Joint Conference on Knowledge Discovery, Knowledge Engineering and Knowledge Management*, pp. 525–531. doi: 10.5220/0008364105250531.
- Handayani, F. and Pribadi, S. (2015) ‘Implementasi Algoritma Naive Bayes Classifier dalam Pengklasifikasian Teks Otomatis Pengaduan dan Pelaporan Masyarakat melalui Layanan Call Center 110’, *Jurnal Teknik Elektro*, 7(1), pp. 19–24. doi: 10.15294/jte.v7i1.8585.
- Hassan, A. and Mahmood, A. (2017) ‘Deep Learning Approach for Sentiment Analysis of Short Texts’, in *2017 3rd International Conference on Control, Automation and Robotics Deep*, pp. 705–710. doi: 10.1109/ICCAR.2017.7942788.
- Indolia, S. et al. (2018) ‘Conceptual Understanding of Convolutional Neural Network- A Deep Learning Approach’, in *Procedia Computer Science*. Elsevier B.V., pp. 679–688. doi: 10.1016/j.procs.2018.05.069.
- Indurkhya, N. and Damerau, F. J. (2010) *Handbook of Natural Language Processing, Second Edition*, *Handbook of Natural Language Processing, Second Edition*. doi: 10.1201/9781420085938.
- Juwiantoro, H. et al. (2020) ‘Sentiment Analysis Twitter Bahasa Indonesia Berbasis Word2vec Menggunakan Deep Convolutional Neural Network’, *Jurnal*

- Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 7(1), pp. 181–188. doi: 10.25126/jtiik.202071758.
- Krizhevsky, A., Sutskever, I. and Hinton, G. E. (2007) ‘ImageNet Classification with Deep Convolutional Neural Networks’, pp. 1–1432. doi: 10.1201/9781420010749.
- Krouská, A., Troussas, C. and Virvou, M. (2016) ‘The effect of preprocessing techniques on Twitter sentiment analysis’, in *IISA 2016 - 7th International Conference on Information, Intelligence, Systems and Applications*. doi: 10.1109/IISA.2016.7785373.
- Lecun, Y. et al. (1998) ‘Gradient-Based Learning Applied to Document Recognition’, in *Biochemical and Biophysical Research Communications*, pp. 2278–2324. doi: 10.1016/j.bbrc.2005.03.111.
- Lutfi, A. A., Permanasari, A. E. and Fauziati, S. (2018) ‘Sentiment Analysis in the Sales Review of Indonesian Marketplace by Utilizing Support Vector Machine’, *Journal of Information Systems Engineering and Business Intelligence*, 4(1), pp. 57–64. doi: 10.20473/jisebi.4.2.169.
- Medhat, W., Hassan, A. and Korashy, H. (2014) ‘Sentiment analysis algorithms and applications: A survey’, *Ain Shams Engineering Journal*. Faculty of Engineering, Ain Shams University, 5(4), pp. 1093–1113. doi: 10.1016/j.asej.2014.04.011.
- Mikolov, T. et al. (2013) ‘Efficient Estimation of Word Representations in Vector Space’, in *1st International Conference on Learning Representations, ICLR 2013 - Workshop Track Proceedings*, pp. 1–12.
- Mikolov, T., Yih, W. and Zweig, G. (2013) ‘Linguistic Regularities in Continuous Space Word Representations’, in *Proceedings of NAACL-HLT 2013*, pp. 746–751. doi: 10.3109/10826089109058901.
- Nasichuddin, M. A., Adji, T. B. and Widayawan, W. (2018) ‘Performance Improvement Using CNN for Sentiment Analysis’, *IJITEE (International*

- Journal of Information Technology and Electrical Engineering), 2(1), pp. 9–14.* doi: 10.22146/ijitee.36642.
- Nawangsari, R. P., Kusumaningrum, R. and Wibowo, A. (2019) ‘Word2Vec for Indonesian Sentiment Analysis towards Hotel Reviews: An Evaluation Study’, in *4th International Conference on Computer Science and Computational Intelligence 2019 (ICCSCI), 12–13 September 2019*. Elsevier B.V., pp. 360–366. doi: 10.1016/j.procs.2019.08.178.
- Pustejovsky, J. and Stubbs, A. (2013) *Natural Language Annotation for Machine Learning*. Edited by J. Steele and M. Blanchette. United States of America: O’Reilly Media.
- Rani, S. and Kumar, P. (2018) ‘Deep Learning Based Sentiment Analysis Using Convolution Neural Network’, *Arabian Journal for Science and Engineering*. Springer Berlin Heidelberg, 44(4), pp. 3305–3314. doi: 10.1007/s13369-018-3500-z.
- Ranjit, S. et al. (2018) ‘Foreign Rate Exchange Prediction Using Neural Network and Sentiment Analysis’, in *Proceedings - IEEE 2018 International Conference on Advances in Computing, Communication Control and Networking, ICACCCN 2018*. IEEE, pp. 1173–1177. doi: 10.1109/ICACCCN.2018.8748819.
- Ratnawati, F. and Winarko, E. (2018) ‘Sentiment Analysis of Movie Opinion in Twitter Using Dynamic Convolutional Neural Network Algorithm’, *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 12(1), p. 1. doi: 10.22146/ijccs.19237.
- Ray, P. and Chakrabarti, A. (2019) ‘A Mixed approach of Deep Learning method and Rule-Based method to improve Aspect Level Sentiment Analysis’, *Applied Computing and Informatics*. The Authors, (xxxx). doi: 10.1016/j.aci.2019.02.002.
- Saputri, M. S., Mahendra, R. and Adriani, M. (2019) ‘Emotion Classification on

- Indonesian Twitter Dataset', *Proceedings of the 2018 International Conference on Asian Language Processing, IALP 2018*. IEEE, pp. 90–95. doi: 10.1109/IALP.2018.8629262.
- Setti, S. and Wanto, A. (2018) 'Analysis of Backpropagation Algorithm in Predicting the Most Number of Internet Users in the World', *Jurnal Online Informatika*, 3(2), pp. 110–115. doi: 10.15575/join.join.v3i2.205.
- Sonak, A. and Patankar, R. A. (2015) 'A Survey on Methods to Handle Imbalance Dataset', *International Journal of Computer Science and Mobile Computing*, 4(11), pp. 338–343. Available at: <http://ijcsmc.com/docs/papers/November2015/V4I11201573.pdf>.
- Statista (2015) *Number of Twitter users in Indonesia from 2014 to 2019*, Statista Research Department. Available at: <https://www.statista.com/statistics/490548/twitter-users-indonesia/#statisticContainer>.
- Swathi Lakshmi, B., Sini Raj, P. and Raj Vikram, R. (2017) 'Sentiment Analysis Using Deep Learning Technique CNN with KMeans', *International Journal of Pure and Applied Mathematics*, 114(11 Special Issue), pp. 47–57.
- Vatsavai, R. R. et al. (2012) 'Spatiotemporal data mining in the era of big spatial data: Algorithms and applications', in *Proceedings of the 1st ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, BigSpatial 2012*, pp. 1–10. doi: 10.1145/2447481.2447482.
- Wijaya, H. et al. (2013) 'Twitter Sentiment Analysis and Insight for Indonesian Mobile Operators', in *Information Systems International Conference (ISICO)*, p. 367.
- Windasari, I. P., Uzzi, F. N. and Satoto, K. I. (2017) 'Sentiment Analysis on Twitter Posts: An analysis of Positive or Negative Opinion on GoJek', in *Proceedings - 2017 4th International Conference on Information*

- Technology, Computer, and Electrical Engineering, ICITACEE 2017.* doi: 10.1109/ICITACEE.2017.8257715.
- Xia, W. *et al.* (2019) ‘High-Resolution Remote Sensing Imagery Classification of Imbalanced Data Using Multistage Sampling Method and Deep Neural Networks’, *Remote Sensing*, 11(21), pp. 1–25. doi: 10.3390/rs11212523.
- Yoo, S. Y., Song, J. I. and Jeong, O. R. (2018) ‘Social media contents based sentiment analysis and prediction system’, *Expert Systems with Applications*. Elsevier Ltd, 105, pp. 102–111. doi: 10.1016/j.eswa.2018.03.055.
- Zhao, Z. *et al.* (2019) ‘An image-text consistency driven multimodal sentiment analysis approach for social media’, *Information Processing and Management*. Elsevier, 56(6), p. 102097. doi: 10.1016/j.ipm.2019.102097.
- Zulfa, I. and Winarko, E. (2017) ‘Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network’, *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 11(2), pp. 187–198. doi: 10.22146/ijccs.24716.