

**PENERAPAN DEEP LEARNING UNTUK KLASIFIKASI BERITA HOAX
MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**
(Studi Kasus: Penyebaran Berita Hoax di Media Sosial)

TUGAS AKHIR

Tugas Akhir ini sebagai salah satu syarat untuk memenuhi gelar sarjana Informatika Universitas Pembangunan Nasional “Veteran” Yogyakarta



Disusun oleh:

Yahdi Indrawan

123160039

**PROGRAM STUDI INFORMATIKA
JURUSAN INFORMATIKA
FAKULTAS TEKNIK INDUSTRI
UNIVERSITAS PEMBANGUNAN NASIONAL “VETERAN”
YOGYAKARTA
2020**

HALAMAN PENGESAHAN PEMBIMBING

PENERAPAN *DEEP LEARNING* UNTUK KLASIFIKASI BERITA HOAX MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*



Mengetahui,
Ketua Program Studi
Dr. Heriyanto, A.Md., S.Kom., M.Cs.
NIK. 2 7706 11 0301 1

HALAMAN PENGESAHAN PENGUJI

PENERAPAN *DEEP LEARNING* UNTUK KLASIFIKASI BERITA HOAX MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*



Penguji III

Nur Heri Cahyana, S.T., M.Kom.
NIK. 1960 09 22 1984 03 1001

Penguji IV

Budi Santosa, S.Si., M.T.
NIK. 2 7009 02 0234 1

SURAT PERNYATAAN
KARYA ASLI TUGAS AKHIR

Sebagai mahasiswa Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta, yang bertanda tangan dibawah ini, saya :

Nama : Yahdi Indrawan

No Mhs : 123160039

Menyatakan bahwa karya ilmiah saya yang berjudul :

**PENERAPAN DEEP LEARNING UNTUK KLASIFIKASI BERITA HOAX
MENGGUNAKAN CONVOLUTIONAL NEURAL NETWORK**

merupakan karya asli saya dan belum pernah dipublikasikan dimanapun. Apabila di kemudian hari, karya saya disinyalir bukan merupakan karya asli saya, maka saya bersedia menerima konsekuensi apa pun yang diberikan Program Studi Informatika Fakultas Teknik Industri Universitas Pembangunan Nasional “Veteran” Yogyakarta kepada saya.

Demikian surat pernyataan ini saya buat dengan sebenarnya.

Dibuat di : Yogyakarta

Pada tanggal : 11 November 2020

Yang menyatakan



(Yahdi Indrawan)

PERNYATAAN BEBAS PLAGIAT

Saya yang bertanda tangan dibawah ini :

Nama : Yahdi Indrawan

No. Mhs : 123160039

Fakultas/Prodi : Teknik Industri/ Informatika

Dengan ini saya menyatakan bahwa judul Tugas Akhir

PENERAPAN *DEEP LEARNING* UNTUK KLASIFIKASI BERITA HOAX MENGGUNAKAN *CONVOLUTIONAL NEURAL NETWORK*

adalah hasil kerja saya sendiri dan benar bebas dari plagiat kecuali cuplikan serta ringkasan yang terdapat di dalamnya telah saya jelaskan sumbernya (sitasi) dengan jelas. Apabila pernyataan ini terbukti tidak benar maka saya bersedia menerima sanksi sesuai peraturan Mendiknas RI No 17 Tahun 2010 dan Peraturan Perundang-undangan yang berlaku.

Demikian surat pernyataan ini saya buat dengan penuh tanggung jawab.

Yogyakarta, 11 November 2020

Yang membuat pernyataan



Yahdi Indrawan
NIM 123160039

HALAMAN PERSEMBAHAN

“Puji syukur kehadirat Allah SWT yang telah melimpahkan karunia-Nya sehingga saya dapat menyelesaikan tugas akhir ini. Tugas akhir ini saya persembahkan kepada diri saya sendiri yang telah berjuang sekuat tenaga hingga berada di titik ini, dan kedua orang tua yang senantiasa mendukung, mendoakan dan memberi semangat kepada saya untuk menyelesaikan tugas akhir ini.”

***“Hari ini harus lebih baik dari hari kemarin,
dan hari esok harus lebih baik dari hari ini”***

ABSTRAK

Penyebaran berita *hoax* saat ini sedang marak terjadi, bahkan sudah masuk fase yang cukup mengkhawatirkan. Hal ini berdampak buruk bagi masyarakat seperti pengaburan fakta, hilangnya kepercayaan masyarakat dan dapat mengakibatkan perpecahan antar masyarakat. Banyaknya berita *hoax* yang tersebar membuat sulitnya masyarakat dalam memilah dan membedakan mana berita *hoax* atau bukan. Ada beberapa masyarakat yang kritis terhadap berita yang tersebar, lalu melakukan pengecekan apakah berita tersebut *hoax* atau bukan. Namun, tidak sedikit pula masyarakat yang langsung percaya terhadap berita yang disampaikan. Maka dari itu, diperlukan proses pengklasifikasian suatu konten berita apakah berita tersebut termasuk berita *hoax* atau fakta.

Penelitian ini menggunakan algoritma *Convolutional Neural Network* (CNN) dengan pembobotan *word2vec*. CNN adalah salah satu algoritma dari metode *deep learning*. Algoritma ini merupakan pengembangan dari *Multi Layer Perceptron* yang didesain untuk mengolah data dua dimensi. Namun, pada penelitian ini menggunakan algoritma CNN untuk klasifikasi berita *hoax* dengan bentuk data teks atau satu dimensi. Proses yang dilakukan yaitu melakukan pengumpulan data menggunakan metode *web scraping* dari situs turnbackhoax.id dan dilanjutkan dengan proses labeling data menjadi 2 kelas (*hoax* dan fakta). Lalu, data dilakukan proses *text processing*, pembobotan kata (*word embedding*) menggunakan metode *word2vec* serta melakukan proses *training* dan *testing* untuk membuat model klasifikasi CNN. Model yang dihasilkan akan dilakukan pengujian menggunakan tabel *confusion matrix* dan kurva ROC serta menggunakan metode validasi *k-fold cross validation* dengan nilai $k=5$. Hasil pengujian tersebut akan menghasilkan nilai rata-rata dari akurasi, presisi, recall dan AUC.

Berdasarkan hasil pengujian menggunakan *confusion matrix* menghasilkan nilai rata-rata akurasi sebesar 86,56%, presisi sebesar 86,35% dan recall sebesar 86,36%. Sedangkan untuk mengujian menggunakan kurva ROC menghasilkan rata-rata nilai AUC sebesar 0.864 dari nilai FPR dan TPR secara urut sebesar 0.156 dan 0.883. Nilai AUC yang tersebut berada pada rentang nilai 0.80 - 0.90 yang menunjukkan bahwa model CNN yang dihasilkan termasuk ke dalam kategori *good classification*.

Kata kunci : Berita *Hoax*, Klasifikasi Teks, *Deep Learning*, *Word2vec*, *Convolutional Neural Network*, *Confusion Matrix*, Kurva ROC, *K-Fold Cross Validation*

KATA PENGANTAR

Dengan mengucap puji dan syukur kehadirat Allah SWT yang telah memberikan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul “Penerapan *Deep Learning* untuk Klasifikasi Berita *Hoax* menggunakan *Convolutional Neural Network*”. Penyusunan tugas akhir ini bertujuan untuk memenuhi salah satu syarat dalam menyelesaikan pendidikan jenjang Strata Satu (S1) Program Studi Informatika Universitas Pembangunan Nasional “Veteran” Yogyakarta.

Didalam penyusunan tugas akhir ini, penulis menyadari bahwa penulisan ini tidak lepas dari bimbingan, bantuan dan dukungan dari berbagai pihak baik secara langsung maupun tidak langsung. Oleh karena itu, penulis mengucapkan terimakasih kepada:

1. Allah SWT yang senantiasa memberikan petunjuk, kekuatan dan kemudahan selama menyelesaikan tugas akhir.
2. Orangtua (Kasidi dan Salbiyah), *Teteh* (Diah) dan Adik (Trie) yang selalu sabar untuk mendukung dan mendoakan penulis dalam menyelesaikan tugas akhir ini.
3. Bapak Heru Cahya Rustamaji, S.Si., M.T., selaku dosen pembimbing I sekaligus dosen wali yang telah meluangkan waktu, tenaga, dan pikiran dalam memberikan masukan dan bimbingan kepada penulis mulai dari diskusi menentukan tema tugas akhir hingga tugas akhir ini dapat terselesaikan.
4. Bapak Dr. Awang Hendrianto Pratomo, S.T., M.T., selaku dosen pembimbing II yang telah membimbing dan memberikan arahan kepada penulis dalam bentuk saran maupun kritik yang membangun untuk kebaikan tugas akhir ini.
5. Bapak Nur Heri Cahyana, S.T., M.Kom., dan Bapak Budi Santosa, S.Si., M.T., selaku dosen penguji yang telah memberikan masukan baik berupa kritik maupun saran kepada penulis dalam menyelesaikan tugas akhir.

6. Seluruh Dosen dan Pegawai jurusan Informatika atas kebaikan, dukungan dan bantuan yang diberikan kepada penulis.
7. Temanku, Nabila Nurul Aini yang telah bersedia meminjamkan laptopnya sehingga penulis dapat menyelesaikan penyusunan tugas akhir ini dengan lancar.
8. Seluruh teman dan sahabat yang telah mendukung, membantu dan menemani selama perkuliahan dan penggeraan tugas akhir ini Anies, Rizal, Nabila, Rasyid, Irma, Alis, Clarita, Claudy, Risma, Niken, Fachriezal dan Agus.
9. Teman-teman semua khususnya angkatan 2016 yang telah memberikan semangat serta bantuan dalam penyusunan tugas akhir.

Penulis menyadari penyusunan tugas akhir ini tentunya tidak lepas dari kekurangan dan jauh dari kata sempurna. Oleh karena itu, penulis mengharapkan masukan berupa kritik, saran dan tanggapan yang bersifat membangun agar lebih baik untuk kedepannya.

Akhir kata semoga tugas akhir ini bermanfaat bagi semua pihak dan penulis sendiri pada khususnya. *Aamiin.*

Yogyakarta, November 2020

Penulis

DAFTAR ISI

HALAMAN JUDUL	i
HALAMAN PENGESAHAN PEMBIMBING.....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	iv
PERNYATAAN BEBAS PLAGIAT	v
HALAMAN PERSEMBAHAN	vi
ABSTRAK.....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI	x
DAFTAR GAMBAR.....	xiii
DAFTAR TABEL	xvi
DAFTAR MODUL PROGRAM.....	xvii
BAB I PENDAHULUAN	1
1.1 Latar Belakang Masalah	1
1.2 Perumusan Masalah	4
1.3 Batasan Masalah	4
1.3 Tujuan Penelitian	5
1.4 Manfaat Penelitian	5
1.5 Metodologi Penelitian.....	5
1.5.1 Metodologi Pengumpulan Data	5
1.5.2 Metodologi Pengembangan Sistem	6
1.6 Sistematika Penulisan	7
BAB II TINJAUAN PUSTAKA	9
2.1 <i>Hoax</i>	9
2.2 Media Sosial.....	10
2.3 <i>Web Scraping</i>	11
2.4 <i>Text Mining</i>	12
2.5 <i>Text Preprocessing</i>	13
2.5.1 <i>Case Folding</i>	14
2.5.2 <i>Remove Punctuation</i>	14
2.5.3 <i>Remove Number</i>	15
2.5.4 <i>Tokenizing</i>	15
2.5.5 <i>Stopword Removal</i>	16
2.5.6 <i>Stemming</i>	16

2.6	<i>Klasifikasi</i>	19
2.7	<i>Word Embedding</i>	20
2.8	<i>Machine Learning</i>	24
2.9	<i>Deep Learning</i>	25
2.10	<i>Convolutional Neural Network (CNN)</i>	26
2.10.1	<i>Input Layer</i>	27
2.10.2	<i>Convolution Layer</i>	27
2.10.3	<i>Activation Layer</i>	28
2.10.4	<i>Pooling Layer</i>	30
2.10.5	<i>Fully Connected Layer</i>	30
2.11	<i>K-Fold Cross Validation</i>	31
2.12	<i>Confusion Matrix</i>	32
2.13	Penelitian Terdahulu	34
	BAB III METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM.....	35
3.1	Metodologi Penelitian.....	35
3.1.1	Pengumpulan Data.....	35
3.1.2	<i>Labeling</i>	38
3.1.3	<i>Text Preprocessing</i>	38
3.1.3.1	<i>Case Folding</i>	40
3.1.3.2	<i>Remove Punctuation</i>	41
3.1.3.3	<i>Remove Number</i>	42
3.1.3.4	<i>Remove Whitespace</i>	43
3.1.3.5	<i>Tokenizing</i>	45
3.1.3.6	<i>Stopword Removal</i>	45
3.1.3.7	<i>Stemming</i>	47
3.1.4	<i>Word Embedding</i>	48
3.1.5	<i>Convolutional Neural Network</i>	50
3.1.5.1	<i>Input Layer</i>	52
3.1.5.2	<i>Convolutional Layer</i>	52
3.1.5.3	<i>Activation Layer (ReLU)</i>	53
3.1.5.4	<i>Pooling Layer</i>	54
3.1.5.5	<i>Fully Connected Layer</i>	55
3.1.5.6	<i>Softmax</i>	55
3.2	Metodologi Pengembangan Sistem.....	55
3.2.1	Pengumpulan Kebutuhan.....	57
3.2.1.1	Pengumpulan Data	57

3.2.1.2	Analisa Kebutuhan Sistem	57
3.2.2	Proses Desain.....	59
3.2.2.1	Perancangan Sistem.....	59
3.2.2.2	Perancangan Basis Data.....	67
3.2.2.3	Perancangan Antarmuka.....	72
3.3	Perancangan Pengujian	92
3.3.1	Pengujian Model.....	93
3.3.2	Pengujian Sistem	94
BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN	96
4.1	Hasil Penelitian	96
4.1.1	Implementasi Klasifikasi Berita <i>Hoax</i>	96
4.1.1.1	Modul Pengumpulan Data.....	96
4.1.1.2	Modul <i>Labeling</i>	97
4.1.1.3	Modul <i>Text Preprocessing</i>	99
4.1.1.4	Modul <i>Word Embedding</i>	104
4.1.1.5	Modul <i>Convolutional Neural Network</i>	105
4.1.2	Implementasi Aplikasi	111
4.1.2.1	Halaman <i>User</i>	111
4.1.2.2	Halaman <i>Admin</i>	115
4.2	Pengujian.....	127
4.2.1	Pengujian Model.....	127
4.2.1.1	<i>K-Fold Cross Validation</i>	128
4.2.1.2	<i>Confusion Matrix</i>	128
4.2.1.3	Kurva ROC.....	129
4.2.2	Pengujian Sistem	130
4.3	Pembahasan.....	132
BAB V PENUTUP	133
5.1	Kesimpulan	133
5.2	Saran	133
DAFTAR PUSTAKA	xviii
LAMPIRAN	xxi

DAFTAR GAMBAR

Gambar 2.1 Tahapan Text Mining (Hashimi dkk, 2015).....	13
Gambar 2.2 Proses <i>Case Folding</i>	14
Gambar 2.3 Proses <i>Remove Punctuation</i>	15
Gambar 2.4 Proses <i>Remove Number</i>	15
Gambar 2.5 Proses <i>Tokenizing</i>	16
Gambar 2.6 Proses <i>Stopword Removal</i>	16
Gambar 2.7 Proses Stemming	19
Gambar 2.8 Ilustrasi <i>Continous Bag of Words</i> (Abdullah, 2018)	23
Gambar 2.9 Ilustrasi <i>Skip-gram Model</i> (Abdullah, 2018)	24
Gambar 2.10 Diagram <i>Deep Learning</i> (Xing dan Du, 2018)	25
Gambar 2.11 Arsitektur <i>Convolutional Neural Network</i> (Zhang dan Wallace, 2015)	27
Gambar 2.12 <i>Convolutional Layer</i> pada CNN (Yin, 2018).....	28
Gambar 2.13 Contoh Grafik Fungsi Aktivasi (Tarabay, 2019)	29
Gambar 2.14 <i>Pooling Layer</i> (Tandungan, 2019)	30
Gambar 2.15 <i>Fulli Connected Layer</i> (Tandungan, 2019).....	31
Gambar 2.16 Ilustrasi <i>K-Fold Cross Validation</i>	32
Gambar 3.1 Metodologi Penelitian	35
Gambar 3.2 Tampilan Situs Turnbackhoax.id	36
Gambar 3.3 Proses <i>Web Scraping</i> Menggunakan <i>Data Miner</i>	37
Gambar 3.4 <i>Flowchart Text Preprocessing</i>	39
Gambar 3.5 <i>Flowchart Case Folding</i>	40
Gambar 3.6 <i>Flowchart Remove Punctuation</i>	42
Gambar 3.7 <i>Flowchart Remove Number</i>	43
Gambar 3.8 <i>Flowchart Remove Whitespace</i>	44
Gambar 3.9 <i>Flowchart Tokenizing</i>	45
Gambar 3.10 <i>Flowchart Stopword</i>	46
Gambar 3.11 <i>Flowchart Stemming</i>	47
Gambar 3.12 <i>Flowchart Word Embedding</i>	49
Gambar 3.13 Visualisasi <i>Word Embedding</i>	50
Gambar 3.14 <i>Flowchart CNN</i>	51
Gambar 3.15 Arsitektur CNN	51
Gambar 3.16 Proses <i>Convolutional Layer</i>	52
Gambar 3.17 Proses <i>MaxPooling Layer</i>	54
Gambar 3.18 <i>Fully Connected Layer</i>	55

Gambar 3.19 Metodologi Pengembangan Sistem.....	56
Gambar 3.20 Arsitektur Sistem.....	61
Gambar 3.21 <i>Data Flow Diagram Level 0</i>	62
Gambar 3.22 <i>Data Flow Diagram Level 1</i>	63
Gambar 3.23 <i>Data Flow Diagram Level 2 Proses Login</i>	64
Gambar 3.24 <i>Data Flow Diagram Level 2 Proses Cek Hoax</i>	65
Gambar 3.25 <i>Data Flow Diagram Level 2 Proses Olah Data</i>	66
Gambar 3.26 <i>Data Flow Diagram Level 2 Proses Upgrade Model</i>	67
Gambar 3.27 ERD Klasifikasi Berita <i>Hoax</i>	68
Gambar 3.28 RAT Klasifikasi Berita <i>Hoax</i>	69
Gambar 3.29 Struktur Menu	73
Gambar 3.30 Rancangan Halaman <i>Home</i>	74
Gambar 3.31 Rancangan Halaman <i>Cek Hoax</i>	74
Gambar 3.32 Rancangan Halaman Hasil Prediksi	75
Gambar 3.33 Rancangan Halaman Lapor <i>Hoax</i>	76
Gambar 3.34 Rancangan Halaman <i>Disclaimer</i>	76
Gambar 3.35 Rancangan Halaman <i>Contact</i>	77
Gambar 3.36 Rancangan Halaman <i>Login</i>	78
Gambar 3.37 Rancangan Halaman <i>Dashboard</i>	78
Gambar 3.38 Rancangan Halaman Data Admin	79
Gambar 3.39 Rancangan Halaman Tambah Data Admin	80
Gambar 3.40 Rancangan Halaman Edit Data Admin	81
Gambar 3.41 Rancangan Halaman Data Cek <i>Hoax</i>	82
Gambar 3.42 Rancangan Halaman Cek <i>Hoax</i> Admin	82
Gambar 3.43 Rancangan Halaman Hasil Prediksi Admin	83
Gambar 3.44 Rancangan Halaman Detail Cek <i>Hoax</i>	84
Gambar 3.45 Rancangan Halaman Data Lapor <i>Hoax</i>	85
Gambar 3.46 Rancangan Halaman Tambah Data Lapor <i>Hoax</i>	85
Gambar 3.47 Rancangan Halaman Edit Data Lapor <i>Hoax</i>	86
Gambar 3.48 Rancangan Halaman Data Training	87
Gambar 3.49 Rancangan Halaman Detail Data Training	87
Gambar 3.50 Rancangan Halaman Data Testing	88
Gambar 3.51 Rancangan Halaman Detail Data Testing	88
Gambar 3.52 Rancangan Halaman Data Pengujian	89
Gambar 3.53 Rancangan Halaman Model CNN.....	90
Gambar 3.54 Rancangan Halaman Perbaharui Model.....	90

Gambar 3.55 Rancangan Halaman Tambah Dataset	91
Gambar 3.56 Rancangan Halaman Edit Dataset	91
Gambar 3.57 Rancangan Halaman Ganti Password	92
Gambar 3.58 Rancangan Halaman <i>Logout</i>	92
Gambar 4.1 Arsitektur Model CNN Proses <i>Training</i>	107
Gambar 4.2 Halaman <i>Home</i>	111
Gambar 4.3 Halaman <i>Cek Hoax</i>	112
Gambar 4.4 Halaman Hasil Prediksi	113
Gambar 4.5 Halaman Lapor Hoax	113
Gambar 4.6 Halaman <i>Disclaimer</i>	114
Gambar 4.7 Halaman <i>Contact</i>	114
Gambar 4.8 Halaman <i>Login</i>	115
Gambar 4.9 Halaman <i>Dashboard</i>	116
Gambar 4.10 Halaman Data Admin	116
Gambar 4.11 Halaman Tambah Data Admin	117
Gambar 4.12 Halaman Edit Data Admin	117
Gambar 4.13 Halaman Data Cek Hoax	118
Gambar 4.14 Halaman Cek Hoax Admin	119
Gambar 4.15 Halaman Hasil Prediksi Admin	119
Gambar 4.16 Halaman Detail Cek Hoax Admin	120
Gambar 4.17 Halaman Data Lapor Hoax	121
Gambar 4.18 Halaman Tambah Data Lapor Hoax	121
Gambar 4.19 Halaman Edit Data Lapor Hoax	122
Gambar 4.20 Halaman Data Training	122
Gambar 4.21 Halaman Detail Data Training	123
Gambar 4.22 Halaman Data Testing	123
Gambar 4.23 Halaman Detail Data Testing	124
Gambar 4.24 Halaman Data Pengujian	124
Gambar 4.25 Halaman Model CNN	125
Gambar 4.26 Halaman Perbaharui Model	126
Gambar 4.27 Halaman Tambah Dataset	126
Gambar 4.28 Halaman Ganti Password	127
Gambar 4.29 Halaman <i>Logout</i>	127
Gambar 4.30 Kurva ROC	130

DAFTAR TABEL

Tabel 2.1 Kombinasi Awalan dan Akhiran Tidak Dijinkan (Nazief dan Andriani, 1996)	18
Tabel 2.2 Aturan Peluruhan Kata Dasar (Nazief dan Andriani, 1996)	18
Tabel 2.3 <i>Hyperparameter</i> pada <i>Convolutional Layer</i> (Zufar dan Setiyono, 2016)	28
Tabel 2.4 <i>Confusion Matrix</i>	32
Tabel 2.5 Penelitian Terdahulu	34
Tabel 3.1 Contoh Data Hasil <i>Web Scraping</i>	38
Tabel 3.2 Contoh Data Hasil <i>Labeling</i>	38
Tabel 3.3 Rincian Dataset	38
Tabel 3.4 Hasil <i>Word Embedding</i>	49
Tabel 3.5 <i>Input Layer</i>	52
Tabel 3.6 Hasil <i>Feature Map</i>	53
Tabel 3.7 Hasil <i>ReLU</i>	53
Tabel 3.8 Hasil <i>MaxPooling Layer</i>	54
Tabel 3.9 Hasil Klasifikasi.....	55
Tabel 3.10 Kebutuhan Perangkat Keras.....	59
Tabel 3.11 Kebutuhan Perangkat Lunak.....	59
Tabel 3.12 Struktur Tabel Admin	70
Tabel 3.13 Struktur Tabel Cek Hoax	70
Tabel 3.14 Struktur Tabel Dataset	71
Tabel 3.15 Struktur Tabel Lapor Hoax	71
Tabel 3.16 Struktur Tabel Training	71
Tabel 3.17 Struktur Tabel Testing	72
Tabel 3.18 Struktur Tabel Pengujian	72
Tabel 3.19 Tabel <i>Confusion Matrix</i>	93
Tabel 3.20 Tabel Pengujian <i>K-Fold Cross Validation</i>	94
Tabel 3.21 Tabel FPR dan TPR Kurva ROC	94
Tabel 3.22 Tabel AUC Kurva ROC.....	94
Tabel 3.23 Rancangan Pengujian <i>Black Box</i>	95
Tabel 4.1 Tabel <i>Confusion Matrix</i>	129
Tabel 4.2 Tabel Pengujian <i>K-Fold Cross Validation</i>	129
Tabel 4.3 Tabel FPR dan TPR Kurva ROC	129
Tabel 4.4 Tabel AUC Kurva ROC.....	130
Tabel 4.5 Hasil Pengujian <i>Black Boax</i>	131

DAFTAR MODUL PROGRAM

Modul Program 4.1 <i>Pseudocode Element Selector Data Miner</i>	97
Modul Program 4.2 <i>Pseudocode Import Data Proses Labeling</i>	97
Modul Program 4.3 <i>Pseudocode Proses Labeling</i>	98
Modul Program 4.4 <i>Pseudocode Import Data</i>	99
Modul Program 4.5 <i>Pseudocode Proses Case Folding</i>	100
Modul Program 4.6 <i>Pseudocode Proses Remove Punctuation</i>	101
Modul Program 4.7 <i>Pseudocode Proses Remove Number</i>	101
Modul Program 4.8 <i>Pseudocode Proses Remove Whitespace</i>	102
Modul Program 4.9 <i>Pseudocode Proses Tokenizing</i>	102
Modul Program 4.10 <i>Pseudocode Proses Stopword Removal</i>	103
Modul Program 4.11 <i>Pseudocode Proses Stemming</i>	104
Modul Program 4.12 <i>Pseudocode Proses Word Embedding</i>	105
Modul Program 4.13 <i>Pseudocode Import Library Proses Training</i>	105
Modul Program 4.14 <i>Pseudocode Arsitektur CNN Proses Training</i>	106
Modul Program 4.15 <i>Pseudocode Proses Training</i>	108
Modul Program 4.16 <i>Pseudocode Proses Testing</i>	108
Modul Program 4.17 <i>Pseudocode Proses Prediksi</i>	110

BAB I

PENDAHULUAN

1.1 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 disebarluaskan secara masif (Assidik, 2018). Beberapa contoh dari pola berita *hoax* seperti pesan berantai yang terdapat anjuran untuk menyebarkan, penggunaan tata bahasa yang kurang baik, tidak logis dan tidak konsisten serta bertentangan dengan akal sehat (Nasution, 2017). Apabila penyebaran berita *hoax* dibiarkan akan berdampak buruk bagi masyarakat seperti pengaburuan 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 menunjukkan 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 *naive bayes* untuk melakukan klasifikasi berita dari *tweet* menjadi tiga kategori yaitu berita palsu (*hoax*), berita asli (*fact*) dan berita yang tidak terindikasi (Tanjung, 2018). Namun, pada penelitian tersebut memiliki kendala karena tingginya kemiripan antar data membuat sulitnya melakukan klasifikasi berita *hoax* sehingga mempengaruhi tingkat error dan akurasinya. Beberapa penelitian lain melakukan klasifikasi artikel *hoax* dengan berbagai metode seperti *support vector machine* dengan pembobotan TF-IDF (Maulina dan Sagara, 2018) dan membandingkan metode *rocchio* dengan *multinomial naive bayes* (Afriza dan Adisantoso, 2018). Kedua penelitian tersebut telah menghasilkan sistem dengan performa yang baik, namun data yang digunakan relatif sedikit dan belum adanya fitur untuk melakukan *upgrade* model menggunakan data yang baru. Selain itu, ada penelitian yang melakukan klasifikasi pengguna twitter untuk melihat potensi penyebaran berita *hoax* berdasarkan isi konten *tweet* dan perilaku pengguna di media sosial. Penelitian tersebut menggunakan metode *backpropagation* dengan membandingkan algoritma *gradient descent backpropagation* dan *lavenberg-marquad backpropagation*, namun memerlukan waktu yang cukup lama untuk melakukan proses *training* (Lhaksmana dkk, 2017).

Beberapa penelitian tersebut telah menggunakan berbagai metode untuk menyelesaikan permasalahan yang berkaitan tentang berita *hoax* dengan hasil yang baik. Namun, ada beberapa aspek yang perlu ditingkatkan seperti skala data yang digunakan masih relatif rendah, belum adanya fitur untuk *upgrade* model dan tidak dapat menemukan pola berita

hoax karena hasil klasifikasinya tidak berdasarkan pola melainkan kemiripan data. Padahal dalam beberapa kasus, penambahan atau pengurangan beberapa kata dari berita asli akan merubah maksud dan tujuan dari berita tersebut. Maka dari itu, untuk meminimalisir kendala tersebut dapat dilakukan dengan menggunakan metode *deep learning*. Hal tersebut karena metode *deep learning* sangat efektif dan lebih mudah dalam mengidentifikasi pola-pola dari data yang dimasukkan. Selain itu, kelebihan lain dari *deep learning* mampu menangani skala data yang besar dan mengekstraksi fitur dari data tersebut secara otomatis. Salah satu algoritma yang termasuk metode *deep learning* dan mampu mengatasi permasalahan tersebut adalah *Convolutional Neural Network*.

Convolutional Neural Network (CNN) merupakan algoritma yang terdiri dari *neuron* yang memiliki *weight*, *bias* dan *activation function*. Algoritma ini memiliki layer hirarki/bertumpuk-tumpuk dengan struktur dan jumlah jaringan syaraf yang sangat banyak sebagai jaringan tersembunyi (*hidden layer*). CNN akan memanfaatkan proses konvolusi untuk mendapatkan informasi yang merepresentasikan dari data yang digunakan dengan menggerakkan sebuah kernel (filter) berukuran tertentu (Lina, 2019). Pada umumnya algoritma CNN digunakan untuk mengolah data yang berbentuk dua dimensi seperti *image classification*, *object detection* ataupun *face recognition*. Padahal, algoritma ini dapat digunakan dalam lingkup yang lebih luas untuk mengidentifikasi dan mengklasifikasikan data yang berbentuk gambar, video, suara maupun teks. Maka dari itu, algoritma CNN dapat diterapkan pada masalah penelitian ini untuk melakukan klasifikasi berita *hoax*. Beberapa penelitian yang menggunakan algoritma CNN untuk mengatasi data yang berbentuk teks seperti klasifikasi teks medis (Hughes dkk, 2017) dan klasifikasi artikel berita berbahasa indonesia dengan nilai yang baik (Razi, 2017).

Berdasarkan permasalahan yang telah dipaparkan di atas, maka penerapan metode *deep learning* dengan menggunakan *Convolutional Neural Network* menjadi solusi yang

ditawarkan pada penelitian ini. Data yang akan digunakan bersumber dari situs turnbackhoax.id yang telah memvalidasi berita *hoax* yang beredar di media sosial. Data tersebut akan dibagi menjadi data latih dan data uji. Data latih digunakan sebagai bahan belajar untuk menghasilkan model yang mampu mengenali pola-pola berita *hoax*, sedangkan data uji digunakan untuk mengukur tingkat akurasi dalam proses klasifikasinya. Selain itu, penelitian ini juga menguji performa dari algoritma *Convolutional Neural Network* dalam menangani data berbentuk teks yang biasanya algoritma ini digunakan untuk pengolahan citra. Dengan menerapkan algoritma *Convolutional Neural Network* diharapkan dapat menghasilkan suatu model yang dapat dimanfaatkan untuk melakukan klasifikasi berita sehingga dapat diketahui apakah berita tersebut termasuk kategori *hoax* atau bukan.

1.2 Perumusan Masalah

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

1. Bagaimana mengenali pola-pola berita untuk melakukan klasifikasi berita *hoax* menggunakan algoritma *Convolutional Neural Network* ?
2. Bagaimana tingkat performa dari algoritma *Convolutional Neural Network* dalam melakukan klasifikasi berita *hoax* ?

1.3 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*.

1.3 Tujuan Penelitian

Tujuan pada penelitian ini adalah sebagai berikut:

1. Dapat mengenali pola-pola berita untuk klasifikasi berita *hoax* dengan menggunakan algoritma *Convolutional Neural Network*.
2. Dapat mengetahui tingkat performa algoritma *convolutional neural network* untuk klasifikasi berita *hoax*.

1.4 Manfaat Penelitian

Dengan dilakukannya penelitian ini dapat mengetahui performa algoritma *convolutional neural network* dalam 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.

1.5 Metodologi Penelitian

Metodologi penelitian yang digunakan pada penelitian ini adalah sebagai berikut:

1.5.1 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.

2. *Web Scraping*

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 scraping* dengan ketentuan data dari tahun 2015 sampai dengan data tahun 2019.

1.5.2 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.

2. *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.

3. *Construction of Prototype*

Pada proses *construction 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.

4. *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.

1.6 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 digunakan 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* dan *convolutional neural 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

2.1 *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 dkk, 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 difoward 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.

2.2 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 dan 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):

2.3 *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). *Web scraping* berfokus pada pengambilan data dan ekstraksinya. *Web scraping* dapat diartikan sebagai bentuk penyalinan, dimana data spesifik dikumpulkan dan disalin dari suatu web dan disimpan di database lokal atau spreadsheet pusat yang dapat digunakan lagi di proses selanjutnya (Pereira dan Vanitha, 2015). Pendapat lain menyatakan bahwa *web scraping* adalah sebuah proses yang memanfaatkan dokumen berbentuk website dibangun dengan bahasa *markup* seperti HTML atau XHTML berbentuk *semi-structured* yang berasal dari internet dan kemudian dianalisis untuk mendapatkan suatu informasi yang berguna untuk konteks lain (Nikhit, 2015).

Proses *web scraping* pertama kali dilakukan dengan cara manual yaitu dengan *copy paste* dari website ke tempat penyimpanan lokal. Namun, apabila data yang ingin diambil dalam jumlah banyak, maka cara ini kurang efektif karena memerlukan tenaga untuk menyalin dan juga memakan banyak waktu. Selain dengan cara manual, *web scraping* dapat dilakukan dengan otomatis. Beberapa cara untuk melakukan *scraping* secara otomatis dengan menggunakan *coding*, aplikasi ataupun *extension browser*. Sedangkan pada penelitian ini, metode yang digunakan untuk melakukan *scraping* yaitu dengan *extension browser* dari *google* yang bernama *Data Miner*.

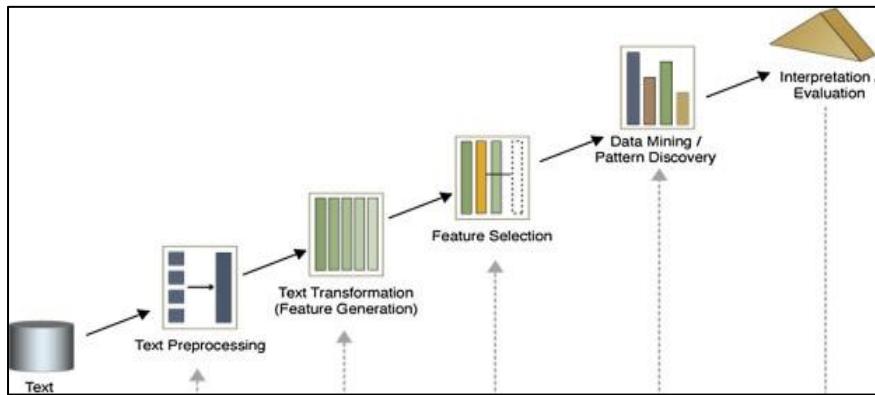
Data miner adalah salah satu *extension chrome* yang berfungsi untuk melakukan proses *scraping* data dari suatu website. *Data miner* akan melakukan *scraping* dengan mengekstrak data dari suatu website dan hasilnya disimpan pada *spreadsheet* atau excel serta dapat dieksport ke dalam bentuk file XLS, CSV, XLSX atau TSV (.xls, .csv, .xlsx, .tsv). Kelebihan dari *data miner* yaitu bisa digunakan dan dipahami oleh semua orang tanpa harus menguasai struktur HTML dari website, adanya fitur *auto pagination* untuk melanjutkan *scraping* ke halaman berikutnya dari website tersebut dan prosesnya *scrapingnya* dilakukan secara otomatis.

2.4 *Text Mining*

Text mining merupakan proses mencari untuk mencari pola pada kumpulan data-data. Proses pencarian pola pada *text mining* dilakukan oleh komputer untuk mendapatkan sesuatu hal yang baru dalam bentuk informasi atau menemukan kembali informasi yang tersirat secara implisit dengan cara melakukan ekstraksi informasi dari sumber data teks yang berbeda-beda (Feldman dan Sanger, 2007). *Text mining* dapat menganalisis dokumen, mengelompokkan dokumen berdasarkan kata yang terkandung didalamnya serta dapat menemukan kesamaan antar dokumen sehingga dapat mengetahui hubungan antar variabelnya.

Penggunaan *text mining* biasanya untuk menganalisis data teks yang semi-terstruktur ataupun data yang tidak terstruktur. Maka dari itu, *text mining* bertanggung jawab untuk merubah data yang awalnya semi-terstruktur atau tidak terstruktur menjadi data yang terstruktur. Berbagai jenis permasalahan yang dapat menggunakan *text mining* seperti klasifikasi, *clustering*, *information extraction* dan *information retrieval* (Berry dan Kogan, 2010). Dari berbagai permasalahan tersebut dapat diaplikasikan pada suatu kasus, seperti klasifikasi artikel berita, analisis klasifikasi, penyaringan spam email dan lain sebagainya. Beberapa tahapan yang biasanya dilakukan untuk melakukan *text mining*, yaitu *text preprocessing*, *text transformation*, *feature selection*, *pattern discovery* dan *interpretation*.

Tahapan *text mining* lebih jelasnya dapat dilihat pada Gambar 2.1.



Gambar 2.1 Tahapan Text Mining (Hashimi dkk , 2015)

2.5 *Text Preprocessing*

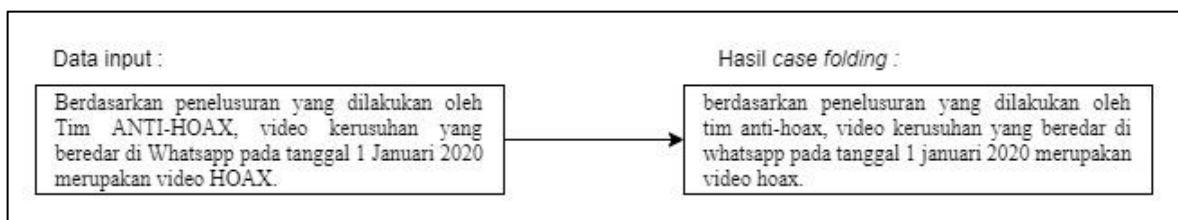
Text preprocessing merupakan salah satu tahapan yang penting pada proses *text mining*. *Text preprocessing* adalah proses untuk membersihkan dan mempersiapkan data sebelum dilakukannya proses klasifikasi (Haddi dkk, 2013). Penggunaan *preprocessing* dikarenakan data yang digunakan pada proses *mining* tidak selamanya dalam kondisi ideal untuk diproses. Tidak jarang data yang akan digunakan memiliki berbagai permasalahan yang dapat mempengaruhi hasil dari proses *mining* itu sendiri. Permasalahan-permasalahan yang sering muncul pada data yang akan diproses seperti *missing value*, *data redundant*, *outliers* ataupun format data yang tidak sesuai dengan sistem. Maka dari itu, proses ini akan

mengubah bentuk data yang awalnya semi-terstruktur ataupun tidak terstruktur menjadi data yang terstruktur.

Berdasarkan penjelasan tersebut, dapat disimpulkan bahwa *text preprocessing* bertujuan untuk mendapatkan bentuk data yang siap diolah atau diproses. Tahapan *text preprocessing* yang akan digunakan meliputi *case folding*, *remove punctuation*, *remove number*, *tokenizing*, *stopword removal* dan *stemming*.

2.5.1 Case Folding

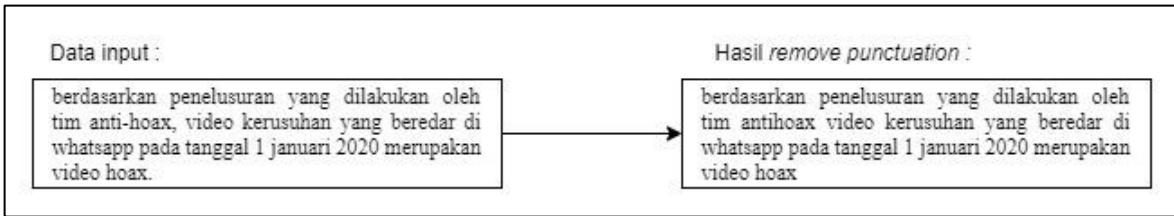
Tahap *case folding* adalah proses merubah semua huruf pada dokumen menjadi huruf kecil (*lowercase*). Huruf yang diterima hanya ‘a’ sampai dengan ‘z’. Perubahan huruf ini bertujuan untuk membentuk suatu standar pada data. Misalnya terdapat data yang berisi “Berita HOAX” akan menjadi “berita hoax”. Contoh lain dari proses *case folding* dapat dilihat pada Gambar 2.2.



Gambar 2.2 Proses *Case Folding*

2.5.2 Remove Punctuation

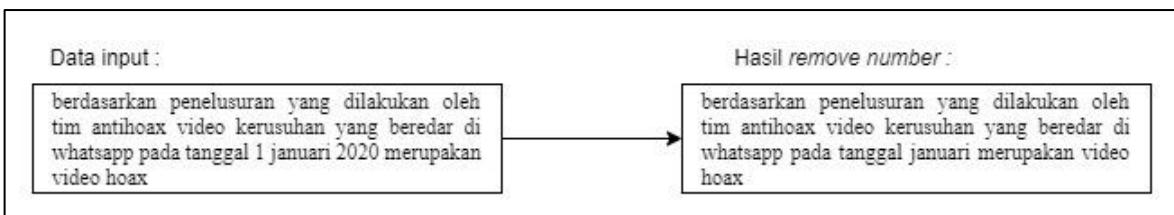
Tahap *remove punctuation* merupakan proses untuk menghapus tanda baca yang terdapat pada suatu data. Pada tahap ini, tanda baca di anggap sebagai *delimiter*. Penghapusan tanda baca pada suatu data akan mengurangi beban pemrosesan. Beberapa contoh tanda baca seperti titik (.), koma (,), tanda tanya (?), tanda seru (!) dan lainnya. Misalnya terdapat data yang berisi “Cek Fakta: Virus Corona tidak bisa masuk Indonesia?” akan berubah menjadi “Cek Fakta Virus Corona tidak bisa masuk Indonesia”. Contoh lain dari proses *remove punctuation* dapat dilihat pada Gambar 2.3.



Gambar 2.3 Proses Remove Punctuation

2.5.3 Remove Number

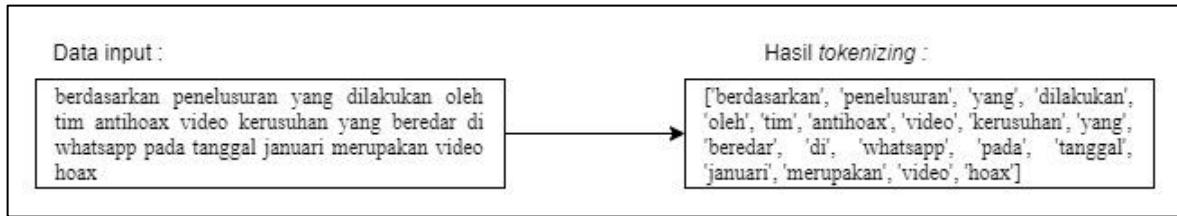
Remove number merupakan tahapan yang hampir sama dengan *remove punctuation*. Bedanya pada bagian objek yang akan dihapus. Pada *remove punctuation* menghapus tanda baca, sedangkan *remove number* akan menghapus semua angka pada suatu data. Angka akan dihapus karena dianggap tidak memiliki arti dan termasuk *delimiter*. Misalnya ada data yang berisi “terdapat 2 akun media sosial yang menyebar berita hoax” akan diubah menjadi “terdapat akun media sosial yang menyebar berita hoax”. Contoh lain penggunaan dari *remove number* dapat dilihat pada Gambar 2.4.



Gambar 2.4 Proses Remove Number

2.5.4 Tokenizing

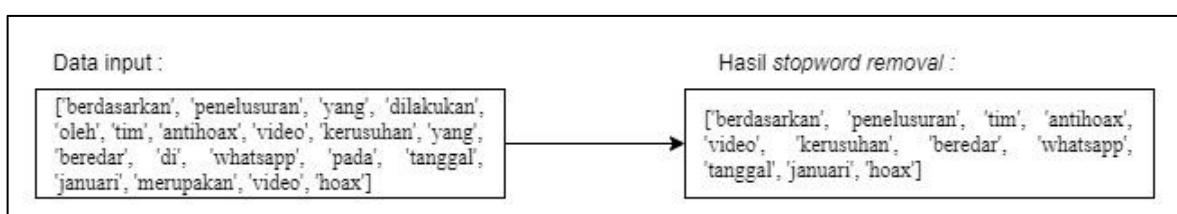
Tahap *tokenizing* atau tokenisasi adalah proses pemisahan teks dari paragraph atau kalimat berdasarkan per kata. Proses *tokenizing* biasanya berpatokan pada karakter *whitespace* seperti spasi, enter ataupun tab sebagai pemisah atau pemotong antar kata. Tiap kata yang akan dihasilkan pada proses ini akan disebut sebagai token. Contoh tokenisasi dari kalimat ”komparasi metode deep learning untuk klasifikasi berita hoax” akan menghasilkan 8 token, yaitu: ”komparasi”, ”metode”, ”deep”, ”learning”, ”untuk”, ”klasifikasi”, ”berita”, dan ”hoax”. Contoh lain yang dapat memperjelas proses *tokenizing* dapat dilihat pada Gambar 2.5.



Gambar 2.5 Proses Tokenizing

2.5.5 Stopword Removal

Stopword removal adalah proses penghapusan kata *stopword*. *Stopword* adalah kata-kata umum yang sering muncul pada suatu dokumen atau data namun tidak memiliki makna (tidak berkaitan dengan tema dokumen). *Stopword* biasanya berupa kata ganti orang atau kata penghubung. Beberapa contoh dari kata *stopword* seperti ‘aku’, ‘kamu’, ‘yang’, ‘di’, ‘dan’, ‘atau’, ‘tetapi’ dan lain sebagainya. Jadi, *stopword removal* merupakan proses pembuangan kata-kata kurang penting (*stopword*) dari token yang telah dihasilkan pada tahap sebelumnya. Pembuangan *stopword* tidak akan mengubah makna atau isi dari suatu data. Bahkan dengan menghapus *stopword* dapat mengurangi ukuran index, waktu pemrosesan dan *noise* dari suatu data. Contoh penerapan dari *stopword removal* dengan inputan hasil dari proses *tokenizing* dapat dilihat pada Gambar 2.6.



Gambar 2.6 Proses Stopword Removal

2.5.6 Stemming

Stemming adalah suatu proses untuk mentransformasi kata-kata yang terdapat pada suatu dokumen menjadi kata dasar (*root word*). Biasanya proses yang dilakukan dengan menghapus awalan, akhiran maupun sisipan. Hal ini bertujuan untuk mengurangi varian kata dengan makna yang hampir sama pada suatu dokumen. Selain itu, proses *stemming* dapat meningkatkan kinerja pengambilan data karena jumlah indeks dengan istilah yang berbeda menjadi berkurang. Contoh proses transformasi kata pada *stemming* pada kata

“menghasilkan” menjadi “hasil”. Terdapat beberapa algoritma yang telah dikembangkan untuk melakukan proses *stemming*. Penggunaan algoritma *stemming* harus disesuaikan dengan bahasa yang digunakan. Algoritma yang digunakan untuk teks dokumen yang menggunakan bahasa Indonesia akan berbeda dengan teks yang berbahasa Inggris. Pada teks berbahasa Indonesia dapat menggunakan beberapa algoritma, salah satunya Algoritma Nazief dan Adriani.

Algoritma Nazief dan Adriani adalah algoritma *stemming* untuk bahasa Indonesia. Algoritma ini memiliki beberapa aturan morfologi untuk merubah suatu kata menjadi kata dasar yaitu dengan menghilangkan *affiks* (awalan, imbuhan dan lain-lain) dan kemudian dicocokkan dengan kamus kata dasar. Kamus kata dasar merupakan hal yang paling utama pada algoritma ini karena semakin banyak kamus kata dasarnya maka semakin tinggi pula akurasinya. Algoritma ini memiliki tahap-tahap sebagai berikut (Nazief dan Adriani, 1996):

1. Cari kata yang akan di-*stem* di dalam kamus, jika data tersebut ditemukan maka kata tersebut adalah kata dasar dan algoritma berhenti. Jika tidak ada maka lanjutkan ke langkah-2.
2. Hilangkan *inflectional suffix* (imbuhan infleksional) yaitu (“-lah”, ”-kah”, “-tah”, “-ku”, “-mu”, “-nya”).
3. Hapus *derivation suffix* (imbuhan turunan) yaitu (“-i”, “-an”, atau “-kan”). Jika kata ditemukan di kamus, maka algoritma berhenti. Jika tidak maka ke langkah-3a.
 - a. Jika “-an” telah dihapus dan huruf terakhir dari kata tersebut adalah “-k” maka “-k” juga ikut dihapus. Jika kata tersebut ditemukan dalam kamus maka algoritma berhenti. Jika tidak ditemukan maka lakukan langkah-3b.
 - b. Akhiran yang dihapus (“-i”, “-an” atau “-kan”) dikembalikan dan lanjut ke langkah-4.

4. Hapus *derivation prefix* (awalan turunan) yaitu (“be-“, “di-“, “ke-“, “me-“, “pe-“, “se-“, “te-“). Jika pada langkah 3 ada *suffix* yang dihapus maka pergi ke langkah-4a, jika tidak maka pergi ke langkah-4b.
- a. Periksa tabel kombinasi awalan-akhiran yang tidak diizinkan. Jika ditemukan maka algoritma berhenti, jika tidak pergi ke langkah-4b. Kombinasi awalan dan akhiran yang tidak diijinkan dapat dilihat pada Tabel 2.1.

Tabel 2.1 Kombinasi Awalan dan Akhiran Tidak Diijinkan (Nazief dan Adriani, 1996)

Awalan	Akhiran yang tidak diijinkan
be-	-i
di-	-an
ke-	-i, -kan
me-	-an
se-	-i, -kan

- b. Untuk $i=1$ sampai 3, tentukan tipe awalan kemudian hapus awalan. Jika kata dasar belum ditemukan juga lakukan langkah-5, jika sudah maka algoritma berhenti. Proses ini harus dilihat terlebih dahulu aturan peluruhan kata pada Tabel 2.2.

Tabel 2.2 Aturan Peluruhan Kata Dasar (Nazief dan Adriani, 1996)

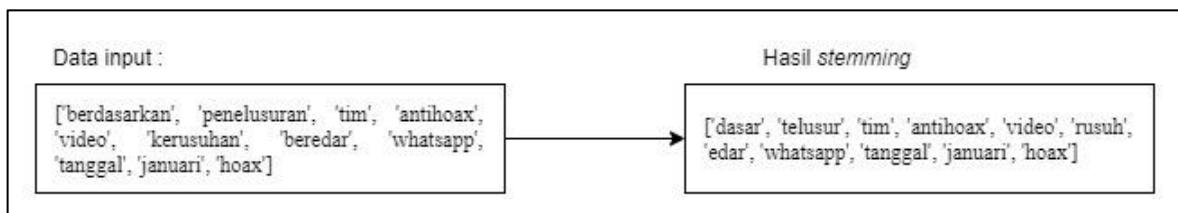
Aturan	Awalan	Peluruhan
1	berV...	ber-V.. be-rV..
2	Belajar	bel-ajar
3	berClerC2	be-ClerC2.. dimana C1!= {'r' 'l'}
4	terV...	ter-V.. te-rV..
5	terCer...	terCer... dimana C!=r'
6	teClerC2	te-CleC2... dimana C1!=r'
7	me{I r w y}V...	me-{I r w y}V...
8	mem{b f v}...	mem-{b f v}...
9	mempe...	m-pe..
10	mem{r V V}...	me-m{r V V}... me-p{r V V}...
11	men{c d j z}...	Men-{c d j z}...
12	menV...	Me-nV... me-tV...
13	meng{g h q k}...	Meng-{g h q k}...
14	mengV...	Meng-V... meng-kV...
15	mengeC...	Meng-C...
16	menyV...	Me-ny... men-sV...
17	memV...	Mem-pV...
18	pe{w y}V...	Pe-{w y}V...
19	perV...	Per-V... pe-rV...
20	pem{b f v}...	Pem-{b f v}...
21	pem{rV V}...	Pe-m{rV V}... pe-p{rV V}...
22	pen{c d j z}...	Pen-{c d j z}...
23	penV...	Pe-nV... pe-tV...
24	peng{g h q}...	Peng-{g h q}
25	pengV...	Peng-V peng-kV
26	penyV...	Pe-nya peny-sV

27	pelV...	Pe-IV...; kecuali untuk kata "pelajar"
28	peCP...	Pe-CP... dimana C!=\{r w y I m n\} dan P!=\'er'
29	perCerV...	Per-CerV... dimana C!=\{r w y I m n\}

Tipe awalan ditentukan melalui langkah-langkah sebagai berikut:

1. Jika awalannya adalah: "di-", "ke-" atau "se-" maka tipe awalannya secara berturut-turut adalah "di-", "ke-" atau "se-".
2. Jika awalannya adalah: "te-", "me-", "be-" atau "pe-" maka dibutuhkan sebuah proses tambahan untuk menentukan tipe awalannya.
3. Jika dua karakter pertama bukan "di-", "ke-", "se-", "te-", "be-", "me-" atau "pe-" maka berhenti.
4. Jika tipe awalan adalah "none" maka berhenti. Hapus awalan jika ditemukan.
5. Lakukan *recording*. Proses *recording* dilakukan dengan menambah karakter *recording* di awal kata yang dipenggal dengan mengacu pada Tabel 2.5.
6. Jika semua langkah telah selesai tetapi tidak juga berhasil maka kata awal diasumsikan sebagai kata dasar. Proses selesai.

Sebagai contoh dari proses *stemming*, misalnya terdapat kata "menghasilkan" menjadi "hasil" atau kata "berasal" akan menjadi "asal". Contoh lain untuk memperjelas proses *stemming* dapat dilihat pada Gambar 2.7.



Gambar 2.7 Proses *Stemming*

2.6 Klasifikasi

Klasifikasi adalah proses untuk menemukan sebuah model atau fungsi yang menggambarkan dan membedakan antar kelas data. Klasifikasi didefinisikan sebagai pengkategorian dokumen secara otomatis ke dalam satu atau lebih kelas yang telah

ditentukan berdasarkan isinya (Sebastiani, 2002). Dokumen-dokumen yang memiliki isi yang sama atau relevan akan dikelompokkan pada kategori yang sama. Dengan kata lain bahwa klasifikasi merupakan pemberian kategori yang telah didefinisikan kepada dokumen yang belum memiliki kategori (Goller, 2000). Tujuan dari klasifikasi yaitu untuk memperkirakan kelas dari suatu objek yang belum diketahui kelasnya (Gaikwad dkk, 2014). Selain itu, akan memudahkan mencari suatu informasi dengan isi yang relevan.

Klasifikasi memiliki dua proses yaitu proses *training* dan *testing*. Proses *training* adalah proses untuk melatih algoritma yang digunakan dengan dokumen yang telah memiliki kategori. Proses *training* akan menghasilkan suatu model yang nantinya dapat digunakan untuk melakukan prediksi kategori dari suatu data. Sedangkan, *proses testing* yaitu proses untuk menguji algoritma dengan data yang belum memiliki kategori untuk memprediksi data tersebut termasuk kategori yang mana. Pada proses *testing* akan diketahui tingkat akurasi dari model yang telah dibangun dari proses *training*.

2.7 ***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. Contoh sederhana untuk merubah teks menjadi vektor angka dengan menggunakan *one-hot encoding*. Misalnya ada kalimat berisi “Indonesia adalah negara berkembang” maka akan dirubah menjadi vektor untuk masing masing kata dimana posisi kata direpresentasikan dengan nilai 1. Vektor yang merepresentasikan kata “Indonesia” adalah [1,0,0,0], kata “adalah” dengan vektor [0,1,0,0], “negara” direpresentasikan dengan [0,0,1,0] dan kata

“berkembang” menjadi [0,0,0,1]. Selain menggunakan *one-hot encoding*, ada metode lain yang sering digunakan untuk melakukan *word embedding* yaitu *word2vec*.

Word2vec merupakan metode yang digunakan untuk melakukan transformasi dari teks menjadi suatu vektor angka. Proses *word embedding* dengan metode *word2vec* menggunakan konsep *neural network* yang memetakan kata ke variabel target (Abdullah, 2018). Sama halnya dengan *neural network*, proses yang dilakukan dengan metode *word2vec* ini menggunakan *weight* sebagai representasi vektor kata. *Word2vec* akan mengetahui hubungan semantik/sintatikal antar kata diruang vektor. Pada implementasinya, *word2vec* memiliki dua teknik untuk melakukan proses *word embedding* yaitu *Continous Bag of Words* dan *Skip-Gram Model*. Misalnya ingin membuat model dari suatu data training yang terdiri nilai input berupa kata “king” dan “brave” maka hasilnya adalah “man”. Alur kerja yang dilakukan untuk membuat model tersebut apabila menggunakan *continous bag of words* adalah sebagai berikut:

1. Lakukan proses transformasi pada layer input dan layer output (target) menggunakan *one-hot encoding* sehingga vektor yang dihasilkan berukuran $[1 \times V]$. Dimana nilai V adalah jumlah kata dari hasil *tokenizing*. Misalnya pada contoh ini memiliki 6 kata maka matriks input berukuran $2[1 \times V]$ dimana $V = 6$ dan 2 adalah kata “king” dan “brave”. Bentuk matriks input yang dihasilkan dapat dilihat pada Formula 2.1.

$$\begin{aligned} \text{king} &= (1 \ 0 \ 0 \ 0 \ 0 \ 0) \\ \text{brave} &= (0 \ 1 \ 0 \ 0 \ 0 \ 0) \end{aligned} \quad (2.1)$$

Sedangkan matriks dari layer output berukuran $1[1 \times V]$ seperti pada Formula 2.2.

$$\text{man} = (0 \ 0 \ 1 \ 0 \ 0 \ 0) \quad (2.2)$$

2. Tahap selanjutnya yaitu melakukan pembobotan. Proses pembobotan dilakukan dua kali yaitu di antara *input layer* dengan *hidden layer* dan di antara *hidden layer* dengan *output layer*. Matriks *input hidden layer* berukuran $[V \times N]$ dan matriks dari *output hidden layer*

berukuran $[N \times V]$. Dimana nilai N merupakan jumlah *neuron*. Misalnya jumlah *neuronnya* adalah 4 maka matriks *input hidden layer* memiliki ukuran $[6 \times 4]$ seperti pada Formula 2.3.

$$\text{input hidden layer} = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 \end{pmatrix} \quad (2.3)$$

Sedangkan ukuran matriks dari *output hidden layer* adalah $[4 \times 6]$ dapat dilihat pada Formula 2.4.

$$\text{output hidden layer} = \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 & 1.0 & 1.1 & 1.2 \\ 1.3 & 1.4 & 1.5 & 1.6 & 1.7 & 1.8 \\ 1.9 & 2.0 & 2.1 & 2.2 & 2.3 & 2.4 \end{pmatrix} \quad (2.4)$$

3. Kemudian *input layer* dikalikan dengan *weight* dari *input hidden layer*. Hasil dari proses ini disebut dengan *hidden activation*. Berikut persamaan untuk menghasilkan *hidden activation* pada Formula 2.5. Pada iterasi pertama, bobot akan diberikan secara random.

$$\begin{aligned} \text{hidden activation} &= \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \\ 9 & 10 & 11 & 12 \\ 13 & 14 & 15 & 16 \\ 17 & 18 & 19 & 20 \\ 21 & 22 & 23 & 24 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 2 & 3 & 4 \\ 5 & 6 & 7 & 8 \end{pmatrix} \end{aligned} \quad (2.5)$$

Setelah itu, cari nilai rata-rata dari vektor *hidden activation*. Hasil rata-rata *hidden activation* pada Formula 2.6.

$$\text{avg hidden activation} = (3 \quad 4 \quad 5 \quad 6) \quad (2.6)$$

4. Lakukan proses perkalian antara *hidden activation* dengan *hidden-output weight* untuk memperoleh *output*. Proses perkalian tersebut seperti pada Formula 2.7.

$$\begin{aligned}
 output &= (3 \quad 4 \quad 5 \quad 6) \begin{pmatrix} 0.1 & 0.2 & 0.3 & 0.4 & 0.5 & 0.6 \\ 0.7 & 0.8 & 0.9 & 1.0 & 1.1 & 1.2 \\ 1.3 & 1.4 & 1.5 & 1.6 & 1.7 & 1.8 \\ 1.9 & 2.0 & 2.1 & 2.2 & 2.3 & 2.4 \end{pmatrix} \\
 &= (21 \quad 22.8 \quad 24.6 \quad 26.4 \quad 28.2 \quad 30) \tag{2.7}
 \end{aligned}$$

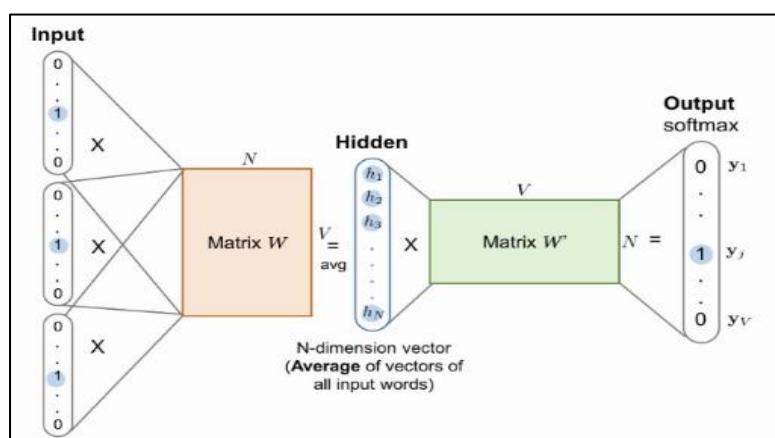
5. Setelah mendapat nilai *output*, maka *output* tersebut ditransformasikan menggunakan fungsi *softmax* untuk mendapatkan nilai probabilitasnya. Berikut hasil probabilitas dari *output* pada Formula 2.8.

$$output \text{ softmax} = \begin{pmatrix} 0.0001030124 \\ 0.0006231887 \\ 0.003770072 \\ 0.02280761 \\ 0.137978 \\ 0.8347181 \end{pmatrix} \tag{2.8}$$

6. Hitung nilai *error* antara *output* dengan target kata. Setelah itu dilakukan *backpropagation* untuk *re-adjust weightnya* untuk memperbaiki nilainya. Cara menghitung yaitu matriks target dikurang matriks *output* seperti pada Formula 2.9.

$$error = \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0.0001030124 \\ 0.0006231887 \\ 0.003770072 \\ 0.02280761 \\ 0.137978 \\ 0.8347181 \end{pmatrix} = \begin{pmatrix} -0.0001030124 \\ -0.0006231887 \\ 0.9962299 \\ -0.02280761 \\ -0.137978 \\ -0.8347181 \end{pmatrix} \tag{2.9}$$

Berdasarkan alur kerja tersebut, maka dapat diilustrasikan proses dari teknik *continuous bag of words* seperti pada Gambar 2.8.

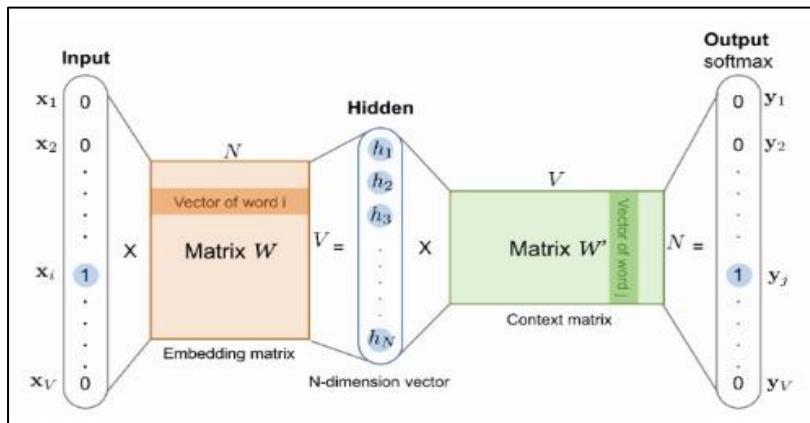


Gambar 2.8 Ilustrasi *Continuous Bag of Words* (Abdullah, 2018)

Sedangkan apabila menggunakan teknik *Skip-Gram Model*, alur kerja yang dilakukan sebagai berikut (Abdullah, 2018):

1. Lakukan transformasi dari nilai input menjadi berukuran $[1 \times V]$, matriks *input hidden weight* berukuran $[V \times N]$ dimana N merupakan jumlah *neuron* dan ukuran matriks dari *hidden output* adalah $C[1 \times V]$.
2. C adalah jumlah konteks katanya.
3. Kemudian nilai matriks dari *hidden activation* dikalikan dengan *weight* yang dilakukan di antara *hidden layer* dan *output layer* untuk menghasilkan *output* dari prediksi.
4. Hasil dari proses sebelumnya berupa nilai *output* ditransformasikan menggunakan fungsi *softmax* untuk mendapatkan nilai probabilitasnya.
5. Hitung *error* atau selisih antara nilai *output* dengan target. Kemudian dilakukan proses *backpropagation* untuk *re-adjust weightnya*.

Alur kerja dari *skip-gram model* lebih jelasnya dapat dilihat ilustrasi pada Gambar 2.9.



Gambar 2.9 Ilustrasi *Skip-gram Model* (Abdullah, 2018)

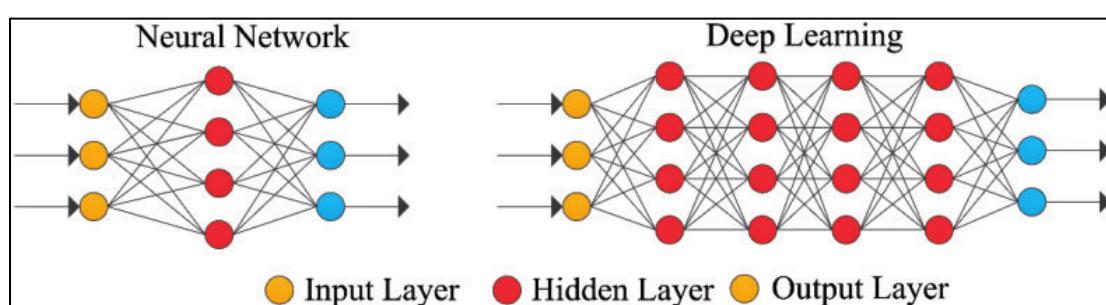
2.8 Machine Learning

machine learning sebagai cabang ilmu komputer yang meneliti bagaimana suatu mesin dapat menyelesaikan masalah tanpa diprogram secara eksplisit. *Machine learning* menggunakan algoritme untuk mencerna rangkaian data, mengambil kesimpulan berdasarkan data yang dianalisis, serta menggunakan kesimpulan tersebut

untuk menyelesaikan tugasnya dengan cara paling efektif. *Machine learning* bertujuan untuk membawa kecerdasan buatan melalui belajar dari data. Data digunakan *machine learning* sebagai kode untuk komputasi tradisional. Cara lain untuk memperoleh kecerdasan dalam mesin bisa melalui pemrograman logis, penalaran induktif berdasarkan aturan dasar dan sebagainya. Dengan demikian *machine learning* dapat dianggap sebagai salah satu pendekatan menuju kecerdasan buatan.

2.9 Deep Learning

Deep learning adalah salah satu bidang dari pembelajaran mesin (*machine learning*) yang berbasis jaringan syaraf tiruan atau *neural network*. Sama halnya dengan *neural network*, *deep learning* juga akan mengajarkan kepada komputer untuk melakukan tindakan seperti manusia, dimana tindakan atau keputusan tersebut berdasarkan hasil belajar dari contoh yang diberikan. *Deep learning* merupakan sebuah pendekatan dalam penyelesaian masalah pada sistem pembelajaran komputer yang menggunakan konsep hirarki (Goodfellow dkk, 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. Secara umum, lapisan dari *deep learning* terdiri dari tiga bagian yaitu *input layer*, *hidden layer* dan *output layer*. Pada bagian *hidden layer* ini lah yang dapat dibuat berlapis-lapis untuk menemukan komposisi algoritma yang tepat agar meminimalisir *error* pada *output* (Primas, 2017). Arsitektur dari *deep learning* dapat diilustrasikan seperti pada Gambar 2.10.

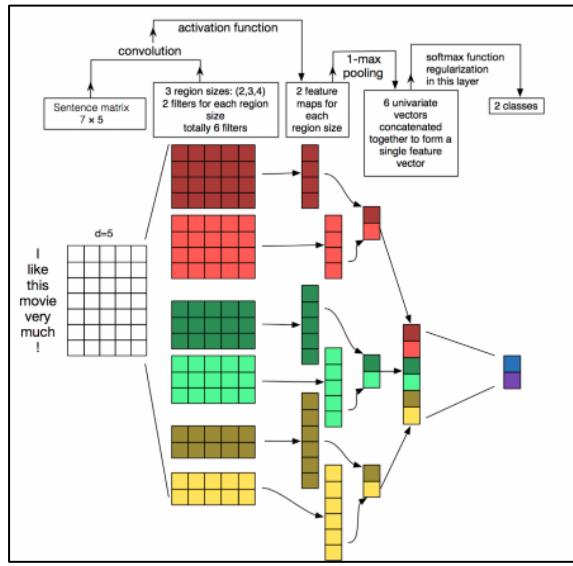


Gambar 2.10 Diagram *Deep Learning* (Xing dan Du, 2018)

Deep learning dapat digunakan untuk menangani berbagai permasalahan dengan data berskala besar seperti *computer vision*, *speech recognition* dan *natural language processing*. Kelebihan dari *deep learning* yaitu memiliki *feature engineering* yang berfungsi untuk mengekstrak pola yang penting dari suatu data sehingga dapat memudahkan model untuk membedakan suatu kelas. Model merupakan hasil dari proses pelatihan pada *deep learning* yang nantinya akan digunakan untuk melakukan prediksi. Algoritma yang digunakan pada *feature engineering* harus disesuaikan dengan kumpulan data dan jenis data yang digunakan. Pemilihan algoritma tersebut berdampak pada baik atau buruknya model untuk melakukan prediksi. Ada beberapa algoritma yang menerapkan konsep dari *deep learning* untuk melakukan *feature engineering* antara lain *Convolutional Neural Network* (CNN), *Recurrent Neural Network* (RNN) dan *Hierarchical Attention Network* (HAN).

2.10 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 yaitu *input layer*, *convolutional layer*, *activation layer*, *pooling layer* dan *fully connected layer*. Arsitektur dari CNN 1D dapat dilihat pada Gambar 2.11:



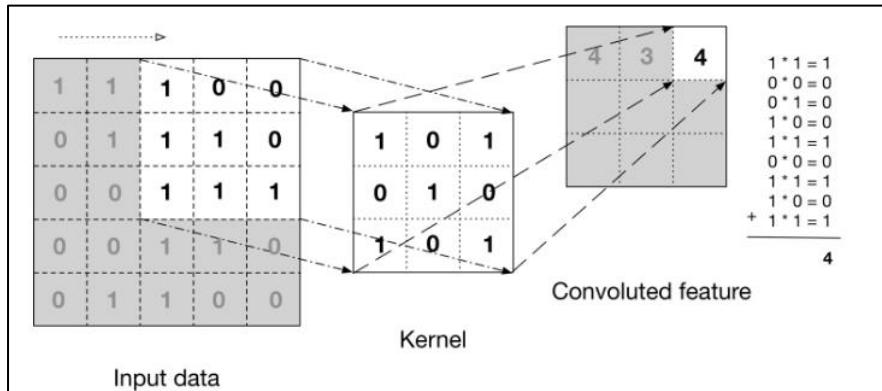
Gambar 2.11 Arsitektur *Convolutional Neural Network* (Zhang dan Wallace, 2015)

2.10.1 *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 5×100 .

2.10.2 *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 5×5 , 3×3 , atau 1×1 . Pada proses ini akan menghasilkan *feature map* yang menunjukkan fitur atau ciri dari kumpulan vektor inputan. Hasil *feature map* merupakan jumlah dari proses perkalian antara input data dengan kernel dengan indeks yang sama dan melakukan pergeseran kernel untuk nilai dari indeks berikutnya. Proses perkalian antara input data dengan kernel untuk menghasilkan *feature map* dapat dilihat pada Gambar 2.12.



Gambar 2.12 Convolution Layer pada CNN (Yin, 2018)

Berdasarkan penjelasan tersebut dapat dirumuskan persamaan untuk menghasilkan *feature map* dengan inputan data x , kernel dilambangkan dengan w dan h sebagai *feature map* yang dihasilkan pada Formula 2.10:

$$h_j^n = \sum_{k=1}^K x_k^{n-1} w_{kj}^n \quad (2.10)$$

Convolution layer memiliki *hyperparameter* yang nilainya mempengaruhi *feature map* yang akan dihasilkan. *Hyperparameter* yang digunakan pada proses konvolusi yaitu *depth*, *stride* dan *zero-padding*. Penjelasan dari masing-masing *hyperparameter* dapat dilihat pada Tabel 2.3.

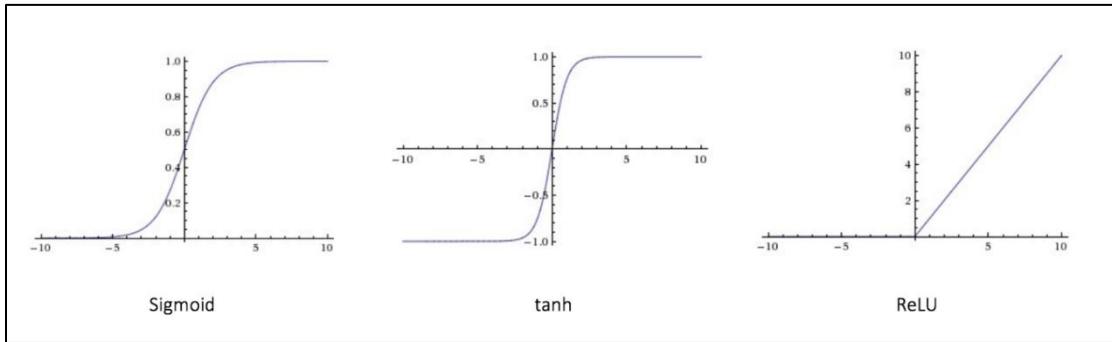
Tabel 2.3 Hyperparameter pada convolutional layer (Zufar dan Setiyono, 2016)

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

2.10.3 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 yaitu sigmoid,

tanh dan ReLu: Fungsi aktivasi tersebut memiliki hasil grafik yang berbeda-beda, visualisasi grafik dari masing-masing fungsi aktivasi seperti pada Gambar 2.13



Gambar 2.13 Contoh Grafik Fungsi Aktivasi (Tarabay, 2019)

Fungsi sigmoid terbentuk dari pembagian antara nilai 1 dengan penjumlahan nilai 1 dan eksponensial pangkat minus x . Persamaan fungsi sigmoid dapat dibentuk seperti Formula 2.11:

$$\sigma_x = \frac{1}{1+e^{-x}} \quad (2.11)$$

Fungsi selanjutnya yaitu Tanh. Persamaan dari fungsi ini adalah selisih eksponensial dari input dengan eksponensial minus nilai input dibandingkan dengan jumlah dari eksponensial nilai input dan eksponensial minus nilai input. Persamaan tersebut dapat ditulis seperti pada Formula 2.12:

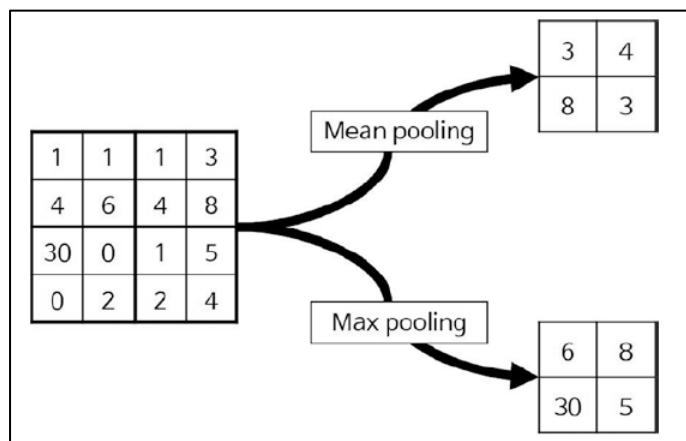
$$\tanh_x = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad (2.12)$$

ReLU didefinisikan sebagai fungsi yang 2 kondisi untuk menghasilkan nilainya. Jika nilai x kurang dari 0 maka fungsi dari x tersebut bernilai 0. Sedangkan apabila nilai x lebih dari sama dengan 0 maka nilai dari fungsi x adalah nilai x tersebut. Persamaan dari fungsi ReLu dapat ditulis seperti pada Formula 2.13

$$ReLU_x = \begin{cases} 0 & \text{if } x < 0 \\ x & \text{if } x \geq 0 \end{cases} \quad (2.13)$$

2.10.4 Pooling Layer

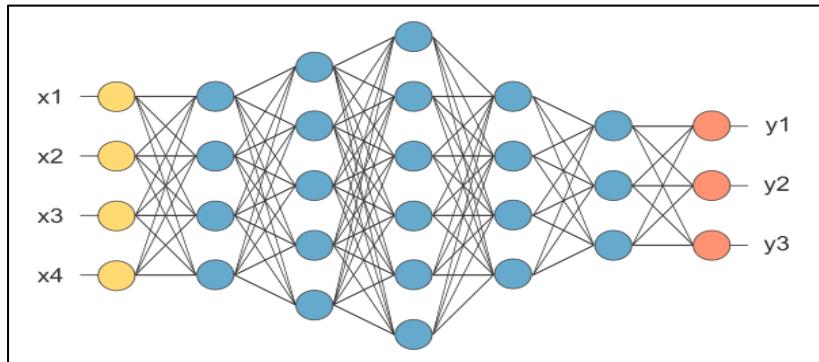
Pooling layer dapat disebut juga dengan *subsampling* atau *downsampling*. Hal tersebut karena *pooling layer* akan mengurangi dimensi dari *feature map* yang telah melalui *activation layer*. Pengurangan dimensi pada tahap ini tidak mempengaruhi informasi penting yang ada didalamnya. Proses *pooling* memiliki beberapa macam, seperti *maxpooling*, *meanpooling*, *sumpooling* dan sebagainya. Proses pada *pooling* dimulai dengan menentukan ukuran *downsampling*, misalnya 2×2 . Setelah itu akan dilakukan proses *pooling* pada *feature map*, sebagai contoh *feature map* yang digunakan berukuran 4×4 . Apabila menggunakan *maxpooling*, maka dari ukuran *downsampling* yang 2×2 diambil nilai tertinggi untuk dijadikan nilai *feature map*. Sedangkan apabila menggunakan *meanpooling* maka nilai rata rata dari *downsampling* yang dijadikan nilai *feature map*. Proses *pooling* diilustrasikan seperti pada Gambar 2.14.



Gambar 2.14 Pooling Layer (Tandungan, 2019)

2.10.5 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 terhubung dengan *neuron-neuron* pada layer tetangganya. Berikut bentuk dari *full connected layer* dapat dilihat pada Gambar 2.15:



Gambar 2.15 Fully Connected Layer (Tandungan, 2019)

Layer ini akan menentukan hasil klasifikasi berdasarkan data yang telah dimasukkan. Proses penentuan hasil klasifikasi menggunakan fungsi aktivasi *softmax*. Fungsi aktivasi *softmax* yaitu rasio eksponensial dari nilai input dengan jumlah nilai eksponensial. Persamaan dari fungsi aktivasi *softmax* dapat dirumuskan seperti pada Formula 2.14:

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^K e^{z_k}} \quad (2.14)$$

dimana :

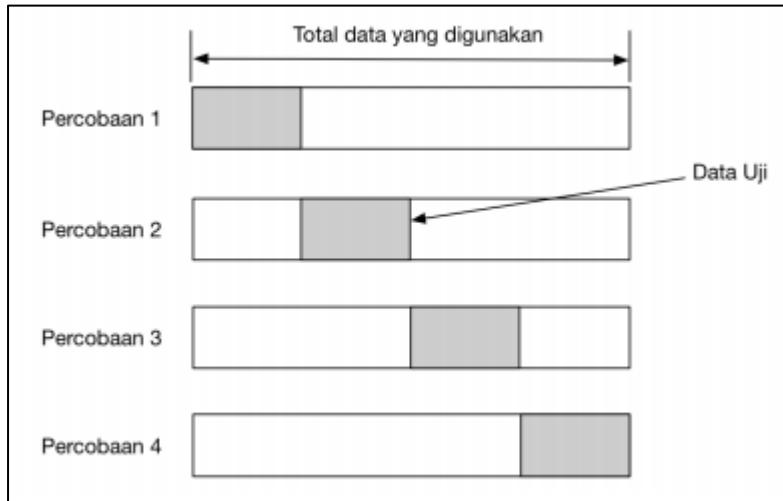
σ = probabilitas input tersebut merupakan class j

z = persamaan linear fungsi pre-aktivasi

j = salah satu class dari K class yang ada

2.11 *K-Fold Cross Validation*

K-fold cross validation merupakan salah satu metode yang digunakan untuk mengukur rata-rata tingkat keberhasilan dari suatu model. Hasil tersebut dilakukan dengan melakukan perulangan untuk menguji model dengan data yang acak. Awalnya, data yang dimiliki akan dibagi sejumlah *n-fold* yang diinginkan sehingga membentuk kelompok kelompok secara acak. Lalu akan dilakukan perulangan terhadap data yang telah dibagi menjadi data latih dan data uji secara bergantikan. Jadi setiap kelompok pernah menjadi data latih dan juga data uji. Berikut ilustrasi dari proses *k-fold cross validation* pada Gambar 2.16.



Gambar 2.16 Ilustrasi K-Fold Cross Validation

2.12 Confusion Matrix

Confusion matrix merupakan metode yang sering digunakan untuk mengukur tingkat performa dari suatu model klasifikasi. *Confusion matrix* akan memberikan hasil perbandingan antara hasil klasifikasi yang dilakukan oleh model dengan hasil klasifikasi yang sebenarnya. Terdapat empat istilah yang merepresentasikan hasil dari proses klasifikasi pada *confusion matrix* yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), dan *False Negative* (FN). Tabel *confusion matrix* dapat dilihat pada Tabel 2.4

Tabel 2.4 Confusion Matrix

		Actual Values	
		Positive	Negative
Predicted Values	Positive	True Positive (TP)	False Positive (FP)
	Negative	False Negative (FN)	True Negative (TN)

Keterangan:

True Positive (TP) : Jumlah data positif yang diprediksi benar

True Negative (TN) : Jumlah data negatif yang diprediksi benar

False Positive (FP) : Jumlah data negatif yang diprediksi salah

False Negative (FN) : Jumlah data positif yang diprediksi salah

Berdasarkan tabel *confusion matrix* tersebut dapat menghitung *performance matrix* untuk mengukur model yang telah dibuat. *Performance matrix* yang paling sering digunakan yaitu akurasi, presisi dan recall.

a. Akurasi

Akurasi merupakan nilai yang menggambarkan seberapa akurat sebuah model dalam mengklasifikasikan dengan benar. Nilai akurasi ini merepresentasikan nilai kedekatan antara nilai prediksi dan nilai yang sebenarnya. Persamaan untuk menghasilkan nilai akurasi yaitu dengan membandingkan nilai prediksi benar dengan keseluruhan data. Rumus menghitung nilai akurasi dapat dilihat pada persamaan 2.15

$$\text{accuracy} = \frac{TP+TN}{TP+FP+FN+TN} \times 100\% \quad (2.15)$$

b. Presisi

Presisi merupakan rasio prediksi benar positif dibandingkan dengan keseluruhan hasil yang diprediksi positif. Rumus untuk menghitung nilai presisi dapat dilihat pada persamaan 2.16

$$\text{precision} = \frac{TP}{TP+FP} \times 100\% \quad (2.16)$$

c. Recall

Recall merupakan perbandingan antara prediksi benar positif dengan keseluruhan data yang benar positif. Untuk menghitung nilai recall dapat menggunakan rumus yang dapat dilihat pada persamaan 2.17

$$\text{recall} = \frac{TP}{TP+FN} \times 100\% \quad (2.17)$$

2.13 Penelitian Terdahulu

Adapun beberapa penelitian yang telah dilakukan dan dijadikan referensi dalam penelitian ini dapat dilihat pada Tabel 2.5

Tabel 2.5 Penelitian Terdahulu

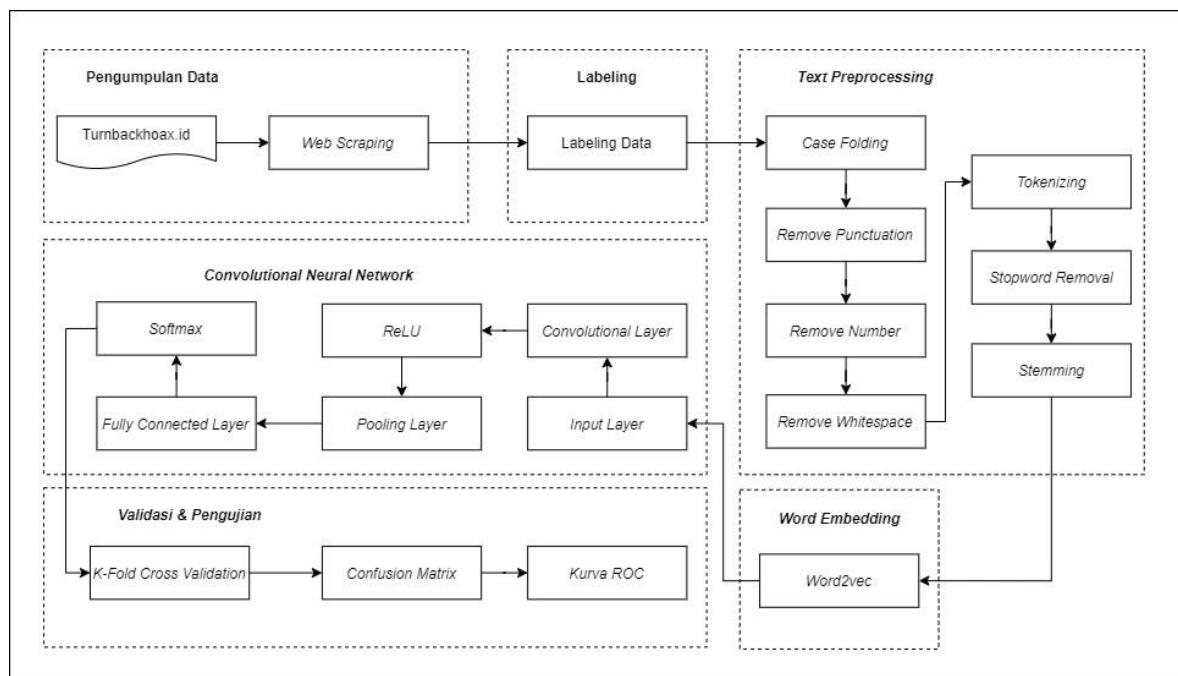
No	Penulis	Judul	Intisari
1	Budi Setiawan Tanjung (2018)	Pendekatan <i>Text Mining</i> sebagai Sistem Pendekripsi Pemberitaan Palsu yang Tersebar dalam Twitter	Sistem telah berhasil mengklasifikasi berita ke dalam tiga kategori yaitu <i>hoax</i> , <i>fact</i> dan <i>grey</i> . Namun, tingginya kemiripan antar data membuat sulit menemukan pola berita <i>hoax</i> sehingga mempengaruhi tingkat <i>error</i> dan akurasinya.
2	Dina Maulina dan Rofie Sagara (2018)	Klasifikasi Artikel <i>Hoax</i> Menggunakan <i>Support Vector Machine Linear</i> dengan Pembobotan <i>Term Frequency-Inverse Document Frequency</i>	Penelitian ini telah menghasilkan sistem dengan performa yang baik dan memiliki kecepatan training 1.37 detik untuk 240 vektor data. Namun, data yang digunakan relatif sedikit sehingga wawasan yang dihasilkan masih kurang optimal.
3	Aulia Afriza dan Julio Adisantoso (2018)	Metode Klasifikasi <i>Rocchio</i> untuk Analisis <i>Hoax</i>	Hasil penelitiannya menunjukkan bahwa algoritma <i>rocchio</i> memiliki akurasi lebih tinggi dibandingkan multinomial naive bayes. Karena kenadala naive bayes yaitu jika probabilitas kondisionalnya sama dengan 0.
4	Kemas Muslim Lhaksmana, Fhira Nhita dan Ageng Budhiarto (2017)	Klasifikasi Pengguna Media Sosial Twitter dalam Persebaran <i>Hoax</i> Menggunakan Metode <i>Backpropagation</i>	Sistem yang dihasilkan memiliki performa yang cukup baik dengan akurasi 72% namun mendapatkan error yang cukup besar karena karakteristik yang hampir sama pada masing-masing kelas. Selain itu membutuhkan waktu yang relatif lama saat proses pelatihan.
5	Mark Hughes, Irene Li, Spyros Kotoulas dan Toyotaro Suzumura (2017)	<i>Medical Text Classification using Convolutional Neural Network</i>	Hasil penelitian menunjukkan bahwa <i>convolutional neural network</i> dapat melakukan klasifikasi teks dengan hasil yang cukup meskipun biasanya metode ini digunakan untuk klasifikasi citra. Selain itu, dapat menghasilkan fitur yang lebih optimal dibandingkan metode pembelajaran yang dangkal.
6	Ar Razi (2017)	Klasifikasi Artikel Berita Berbahasa Indonesia Menggunakan <i>Convolutional Neural Network</i>	Penelitian yang dihasilkan dapat melakukan klasifikasi dengan baik. Namun, dalam penelitian ini hanya menggunakan dataset yang relatif sedikit dan pada bagian <i>preprocessing</i> tidak menggunakan <i>stemming</i> serta <i>stopword</i> sehingga hasilnya kurang optimal.

BAB III

METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM

3.1 Metodologi Penelitian

Pada bagian ini akan membahas tentang metodologi penelitian yang digunakan pada penelitian ini. Metodologi penelitian berisi tahapan-tahapan perencanaan yang akan dilakukan selama proses pembuatan sistem pada penelitian ini. Metodologi penelitian bertujuan untuk menjadi bahan acuan dalam meminimalkan risiko kegagalan sehingga mencapai tujuan yang diinginkan. Adapun tahapan metodologi penelitian yang dilakukan dapat dilihat pada Gambar 3.1.



Gambar 3.1 Metodologi Penelitian

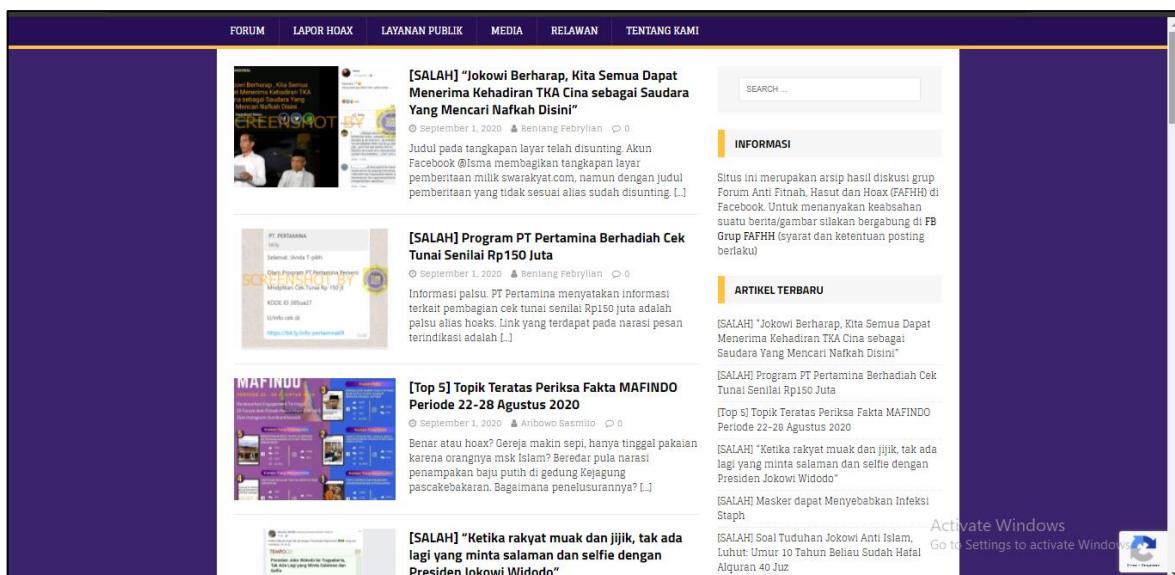
Secara garis besar, terdapat 6 tahapan yang akan dilakukan yaitu pengumpulan data, labeling, *text preprocessing*, *word embedding* menggunakan *word2vec*, algoritma *convolutional neural network* serta validasi & pengujian.

3.1.1 Pengumpulan Data

Data merupakan salah satu hal yang paling penting dalam sebuah penelitian. Tanpa adanya data, suatu penelitian tidak dapat dilakukan. Suatu penelitian memerlukan data untuk

dolah sehingga menjadi sebuah informasi yang dapat membantu untuk pengambilan keputusan atau menghasilkan teori. Maka dari itu, data yang diperlukan dalam penelitian harus dikumpulkan terlebih dahulu. Selain itu, data yang digunakan juga harus dapat dipertanggungjawabkan kebenarannya.

Pada penelitian ini akan menggunakan data berita yang mengandung unsur *hoax* dan berita yang valid kebenarannya. Data yang digunakan adalah data sekunder yang bersumber dari situs turnbackhoax.id. Data yang diambil mulai dari berita tahun 2015 sampai dengan 2019 yang nantinya akan dikategorikan menjadi dua kelas yaitu *hoax* dan fakta.. Alasan situs tersebut menjadi sumber data penelitian karena pada situs tersebut telah memvalidasi kebenaran dari berita-berita yang tersebar di media sosial. Situs tersebut akan melakukan penelusuran berdasarkan aduan dari masyarakat yang menemukan berita *hoax* di media sosial. Berikut tampilan awal situs turnbackhoax.id dapat dilihat pada Gambar 3.2.



Gambar 3.2 Tampilan Situs Turnbackhoax.id

Selanjutnya, akan dilakukan proses pengambilan data dari situs tersebut menggunakan metode *web scraping*. *Web scraping* merupakan teknik pengumpulan data dengan mengekstraksi data dan informasi dari suatu website lalu disimpan ke dalam format tertentu. Dengan menggunakan metode *web scraping* bertujuan untuk memudahkan

pengambilan data dari situs tersebut karena dilakukan secara otomatis tanpa harus menyalin data satu per satu. *Web scraping* memiliki beberapa metode yang dapat digunakan, salah satunya menggunakan *extension browser* yang dilakukan pada penelitian ini. *Extension browser* yang digunakan yaitu *Data Miner* dari *browser* Google.

Proses mengambil data dengan *data miner* cukup mudah karena dapat dilakukan dan dipahami oleh siapapun tanpa harus menguasai pemrograman. Pertama, *install extension data miner* terlebih dahulu di *browser google*. Kedua, buka situs yang ingin diambil datanya, dan pada penelitian ini pada situs turnbackhoax.id. Setelah itu, buka *data miner* dan pilih bagian website yang memuat data dan informasi yang ingin diambil. Selanjutnya, pilih link *pagination* agar data yang diambil dilakukan secara berulang ke data berikutnya. Proses *web scraping* menggunakan *data miner* dapat dilihat pada Gambar 3.3.



Gambar 3.3 Proses Web Scraping Menggunakan *Data Miner*

Selanjutnya, hasil data yang telah diambil dapat dilihat apakah sudah sesuai atau belum. Apabila data yang *discrepe* telah sesuai keinginan, maka data tersebut dapat disimpan dalam format csv. Berikut bentuk dataset yang telah diubah ke format csv yang bersumber dari data turnbackhoax.id.

Tabel 3.1 Contoh Data Hasil *Web Scraping*

Konten
Terlibat pembuatan virus corona di China, FBI menangkap 10 profesor Harvard.
PT Kereta Api Indonesia (KAI) melarang pemakaian masker scuba dan buff_di kereta api jarak jauh. Pasalnya, scuba dan buff dinilai hanya satu lapisan dan tak begitu efektif menangkal virus.

3.1.2 *Labeling*

Labeling merupakan tahapan untuk memberikan label atau kategori pada setiap data.

Semua data akan dikategorikan menjadi 2 kelas yaitu *hoax* atau fakta. Proses pemberian label pada data sesuai dengan judul dari postingan turnbackhoax.id, apabila pada judul tertera tanda “[SALAH]” maka narasi dari data tersebut termasuk kategori *hoax* dan bagian dari penjelasan akan diberi label fakta. Dan apabila ada postingan yang tidak memiliki tanda “[SALAH]” maka akan diberi label fakta. Berikut hasil perubahan dataset setelah dilakukan proses *labeling* dapat dilihat pada Tabel 3.2.

Tabel 3.2 Contoh Data Hasil *Labeling*

Konten	Kelas
Terlibat pembuatan virus corona di China, FBI menangkap 10 profesor Harvard.	Hoax
PT Kereta Api Indonesia (KAI) melarang pemakaian masker scuba dan buff_di kereta api jarak jauh. Pasalnya, scuba dan buff dinilai hanya satu lapisan dan tak begitu efektif menangkal virus.	Fakta

Berdasarkan hasil dari proses *labeling* dapat diketahui rincian dari dataset yang akan digunakan. Data yang didapatkan terdiri dari 1899 berita yang berkategori *hoax* dan 2450 berita yang berkategori fakta. Jadi apabila diakumulasikan maka total dataset yang digunakan pada penelitian ini berjumlah 4349 data. Rincian dari jumlah dataset pada penelitian ini dapat dilihat pada Tabel 3.3

Tabel 3.3 Rincian Dataset

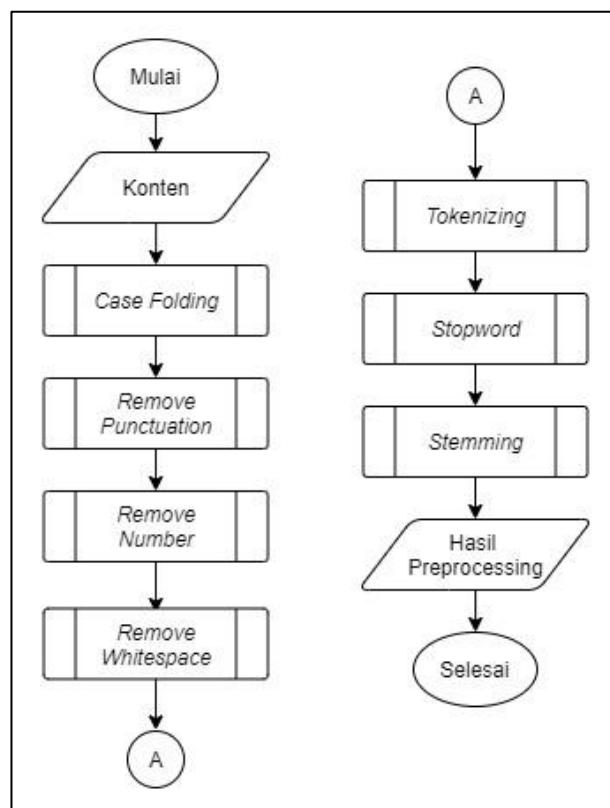
No	Label Klasifikasi	Jumlah Data
1	Hoax	1899
2	Fakta	2450
Total Data		4349

3.1.3 *Text Preprocessing*

Pada bagian ini akan menggambarkan alur proses pada tahap *text preprocessing*. *Text preprocessing* ini bertujuan untuk membersihkan data konten yang menjadi *input* dari

sistem. Hal tersebut membuat data yang awalnya semi-terstruktur atau bahkan tidak terstruktur menjadi data yang terstruktur dan siap untuk digunakan. *Text preprocessing* memiliki beberapa tahapan yang menjadi sub proses diantaranya *case folding*, *remove punctuation*, *remove number*, *remove whitespace*, *tokenizing*, *stopword* dan *stemming*.

Flowchart dari *text preprocessing* dapat dilihat pada Gambar 3.4.



Gambar 3.4 Flowchart Text Preprocessing

Berikut adalah contoh dari hasil *text preprocessing* terhadap sebuah berita berbahasa Indonesia setelah melalui semua proses *text preprocessing*.

Text awal:

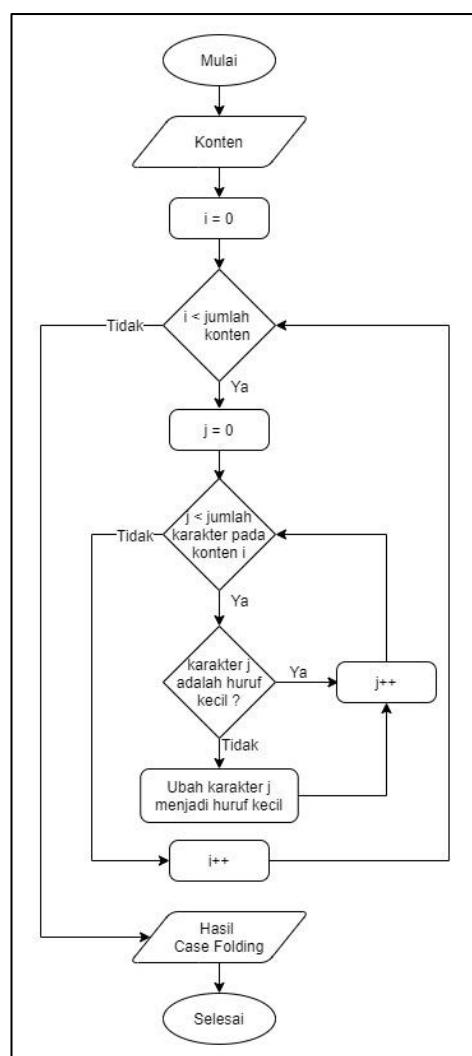
“Terlibat pembuatan virus corona di China, FBI menangkap 10 profesor Harvard.”

Hasil *Preprocessing*:

“['libat', 'buat', 'virus', 'corona', 'china', 'fbi', 'tangkap', 'profesor', 'harvard']”

3.1.3.1 Case Folding

Case folding yaitu proses merubah semua karakter pada dokumen menjadi huruf kecil (*lowercase*). *Case folding* akan menerima *input* berupa konten dan inisialisasi i yang mewaliki jumlah dan j mewakili jumlah karakter. Prosesnya yaitu melakukan pengecekan ke setiap karakter dari awal hingga akhir. Jika karakter merupakan huruf kecil, maka karakter tersebut disimpan. Tetapi apabila menemukan karakter yang huruf besar maka dirubah terlebih dahulu karakter tersebut menjadi huruf kecil, lalu perubahan tersebut di simpan. Ketika semua karakter telah diperiksa, maka akan mengeluarkan *output* berupa hasil *case folding* yang akan digunakan pada proses berikutnya. *Flowchart* dari proses *case folding* dapat dilihat pada Gambar 3.5.



Gambar 3.5 Flowchart Case Folding

Berikut contoh hasil dari proses *case folding* terhadap suatu konten berita yang ingin diperiksa.

Text awal:

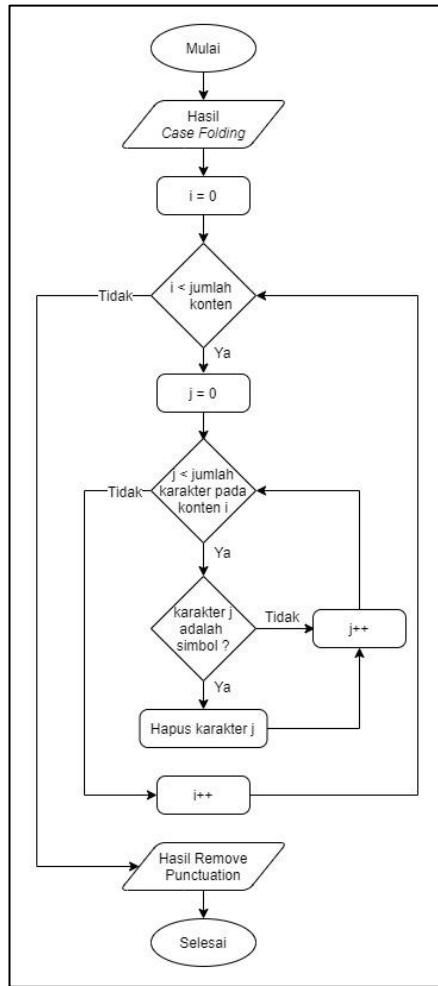
“Terlibat pembuatan virus corona di China, FBI menangkap 10 profesor Harvard.”

Hasil *case folding*:

“terlibat pembuatan virus corona di china, fbi menangkap 10 profesor harvard.”

3.1.3.2 Remove Punctuation

Remove punctuation adalah proses menghapus tanda baca pada suatu dokumen karena dianggap sebagai *delimiter*. Penghapusan tanda baca ini akan mengurangi beban pemrosesan. Beberapa contoh tanda baca seperti titik (.), koma (,), tanda tanya (?), tanda seru (!) dan lainnya. Proses yang terjadi yaitu melakukan pengecekan jumlah dokumen dan jumlah karakternya. Data yang digunakan pada proses ini merupakan data hasil dari proses *case folding*. Kemudian, setiap karakter dari data tersebut akan dicek apakah karakter merupakan simbol atau tanda baca. Jika karakter merupakan tanda baca maka akan dihapus, tetapi jika bukan tanda baca maka karakter tersebut akan disimpan. Apabila semua karakter telah diperiksa, akan memberikan *output* berupa hasil *remove punctuation*. Alur dari proses *remove punctuation* dapat dilihat lebih detail pada Gambar 3.6.



Gambar 3.6 Flowchart Remove Punctuation

Berikut contoh hasil proses *remove punctuation* terhadap data yang merupakan hasil dari *case folding*.

Text *input* (hasil *case folding*):

“terlibat pembuatan virus corona di china, fbi menangkap 10 profesor harvard.”

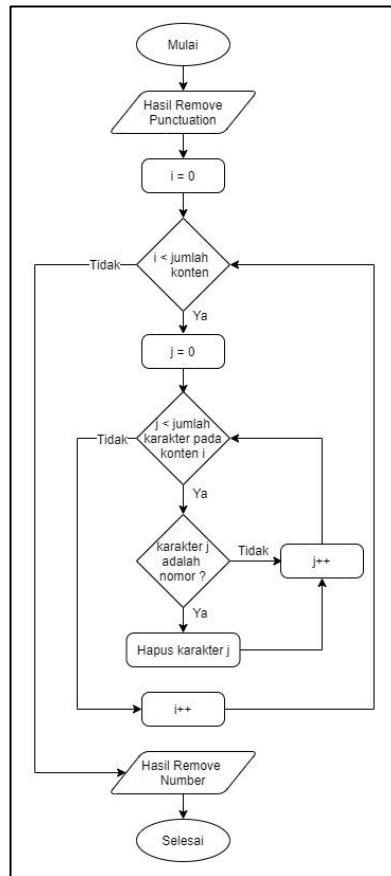
Text hasil *remove punctuation*:

“terlibat pembuatan virus corona di china fbi menangkap 10 profesor harvard”

3.1.3.3 Remove Number

Remove number tidak jauh berbeda dengan *remove punctuation*, perbedaannya hanya pada karakter yang dihapus. Pada *remove number* ini akan menghapus karakter yang berbentuk angka. Pada proses ini akan menerima *input* berupa hasil dari proses *remove*

punctuation, kemudian akan diperiksa per karakternya. Apabila menemukan karakter berbentuk angka, maka karakter tersebut akan dihapus. *Flowchart* dari *remove number* dapat dilihat pada Gambar 3.7.



Gambar 3.7 Flowchart Remove Number

Berikut contoh hasil proses *remove number* terhadap data yang merupakan hasil dari *remove punctuation*.

Text *input* (hasil *remove punctuation*):

“terlibat pembuatan virus corona di china fbi menangkap 10 profesor harvard”

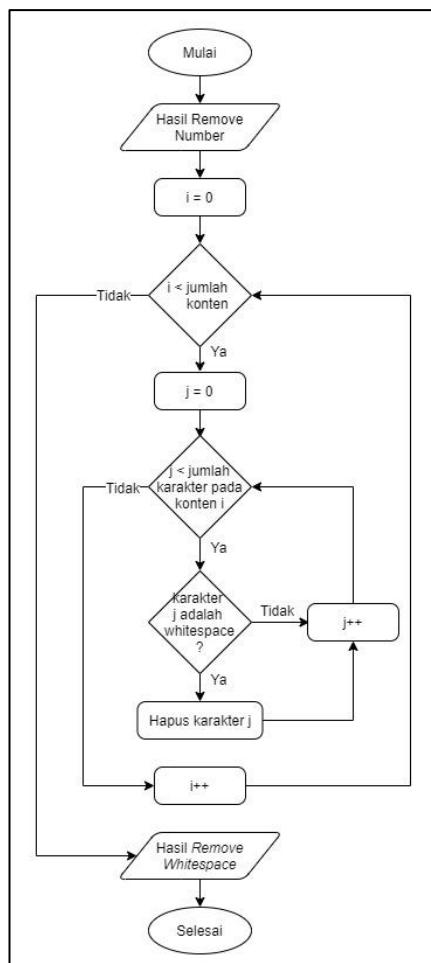
Text hasil *remove number*:

“terlibat pembuatan virus corona di china fbi menangkap profesor harvard”

3.1.3.4 Remove Whitespace

Remove whitespace merupakan proses penghapusan *whitespace* dari data. Proses ini akan mengecek setiap karakter apakah termasuk *whitespace* atau bukan. Jika karakter

merupakan *whitespace* maka karakter tersebut akan dihapus, sebaliknya jika tidak maka karakter akan disimpan. Data yang akan diproses merupakan hasil dari *remove number* dan hasil dari proses ini menghasilkan hasil *remove whitespace*. *Flowchart* dari *remove whitespace* dapat dilihat pada Gambar 3.8.



Gambar 3.8 Flowchart Remove Whitespace

Berikut contoh hasil proses *remove whitespace* terhadap data yang merupakan hasil dari *remove number*.

Text *input* (hasil *remove number*):

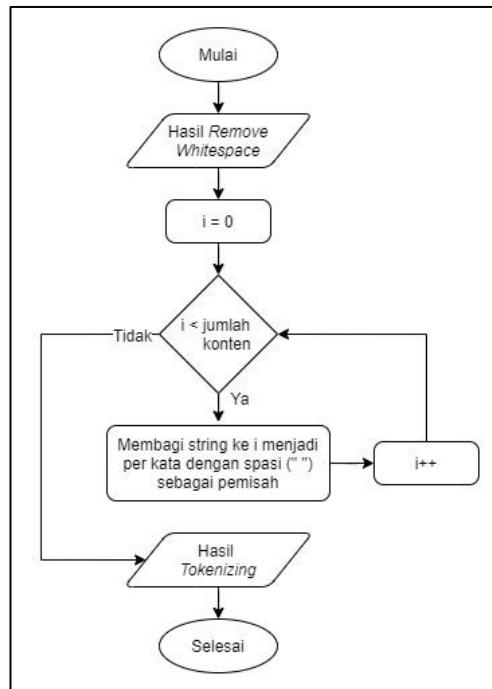
“terlibat pembuatan virus corona di china fbi menangkap profesor harvard”

Text hasil *remove whitespace*:

“terlibat pembuatan virus corona di china fbi menangkap profesor harvard”

3.1.3.5 Tokenizing

Pada bagian ini akan menjelaskan proses *tokenizing*. Proses ini akan memotong data menjadi per kata dengan spasi (“ ”) sebagai pemisahnya. Data yang akan diproses dari hasil *remove whitespace* dan akan mengeluarkan hasil *tokenizing* dalam bentuk token-token (kata). *Flowchart* dari proses *tokenizing* dapat dilihat pada Gambar 3.9.



Gambar 3.9 Flowchart Tokenizing

Berikut contoh hasil proses *tokenizing* terhadap data yang merupakan hasil dari *remove whitespace*.

Text *input* (hasil *remove whitespace*):

“terlibat pembuatan virus corona di china fbi menangkap profesor harvard”

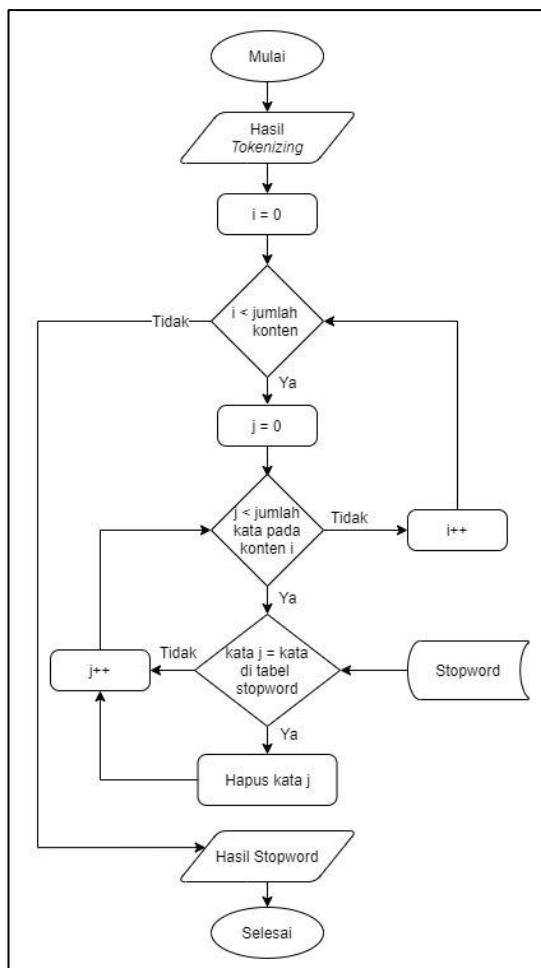
Text hasil *tokenizing*:

“['terlibat', 'pembuatan', 'virus', 'corona', 'di', 'china', 'fbi', 'menangkap', 'profesor', 'harvard']”

3.1.3.6 Stopword Removal

Stopword removal adalah proses pembuangan kata-kata yang kurang penting (*stopword*) pada data. Contoh kata yang kurang penting seperti kata ganti orang, kata penghubung dan sebagainya. Proses pada tahap ini yaitu melakukan pengecekan per kata

dari hasil *tokenizing*. Lalu, setiap kata akan dicek apakah kata tersebut ada di kamus *stopword*. Apabila kata di data ada di kamus *stopword*, maka kata tersebut dihapus. Sebaliknya, jika tidak ada di kamus *stopword* maka akan disimpan. Proses *stopword* dapat dilihat pada Gambar 3.10.



Gambar 3.10 Flowchart Stopword

Berikut contoh hasil proses *stopword removal*; terhadap data yang merupakan hasil dari *tokenizing*.

Text *input* (hasil *tokenizing*):

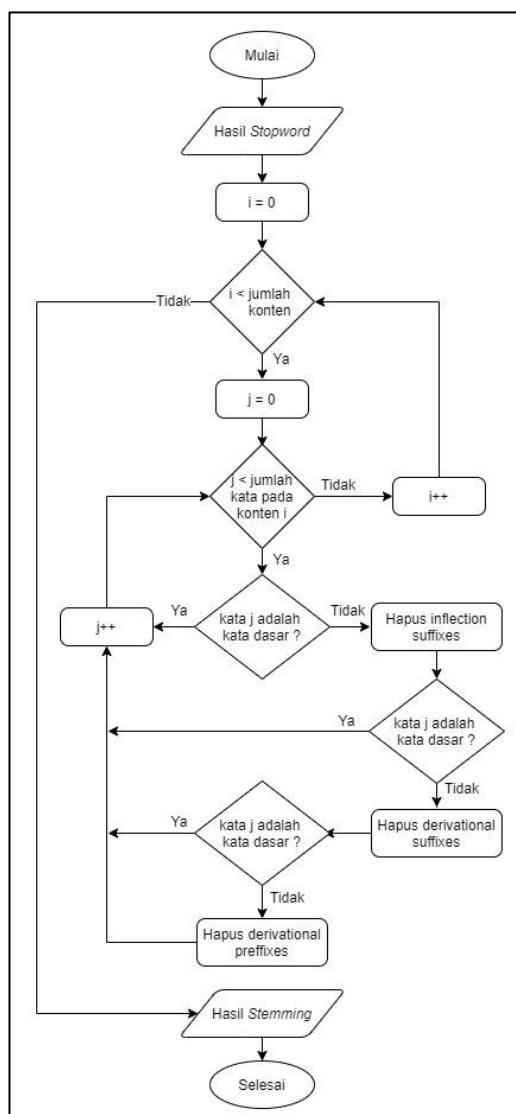
“['terlibat', 'pembuatan', 'virus', 'corona', 'di', 'china', 'fbi', 'menangkap', 'profesor', 'harvard']”

Text hasil *stopword removal*:

“['terlibat', 'pembuatan', 'virus', 'corona', 'china', 'fbi', 'menangkap', 'profesor', 'harvard']”

3.1.3.7 Stemming

Pada bagian ini akan menjelaskan terkait alur dari proses *stemming*. Proses ini akan menggunakan data hasil dari proses *stopword* yang akan diperiksa per kata. *Stemming* akan mencari kata dasar dari setiap kata dengan cara menghapus imbuhan pada kata. Jika kata merupakan kata dasar maka kata tersebut akan disimpan. Apabila kata bukan kata dasar, maka akan dihapus *inflection suffixes*, *derivational suffixes* dan atau *derivational prefixes* yang ada pada kata sehingga kata tersebut menjadi kata dasar. Setelah itu kata disimpan dan *flowchart* ini akan menghasilkan *output* berupa hasil *stemming* sekaligus menjadi hasil dari proses *text preprocessing*. Berikut *flowchart* dari *stemming* dapat dilihat pada Gambar 3.11.



Gambar 3.11 Flowchart Stemming

Berikut contoh hasil proses *stemming* terhadap data yang merupakan hasil dari *stopword removal*.

Text *input* (hasil *stopword removal*):

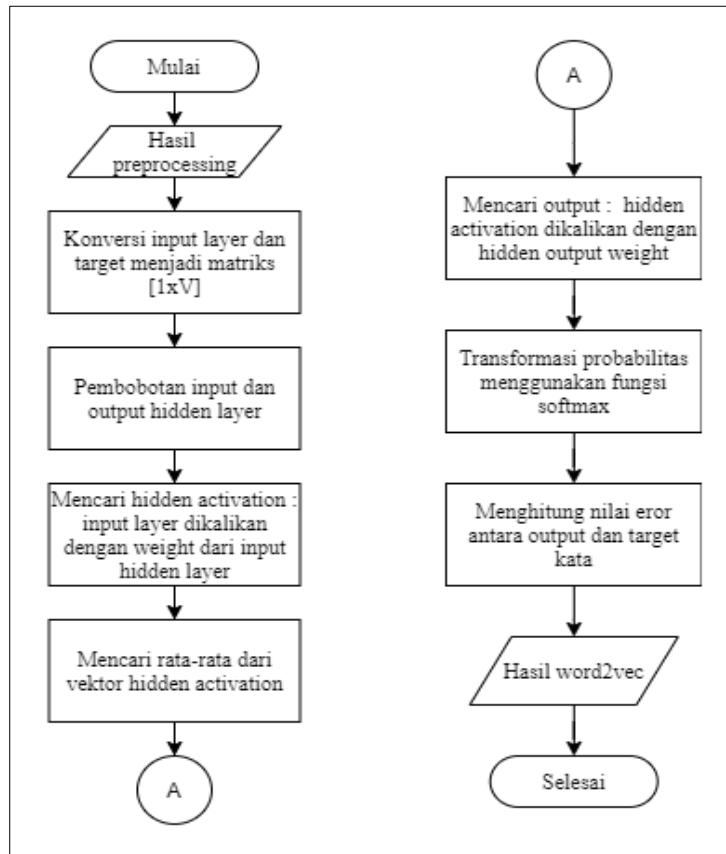
“['terlibat', 'pembuatan', 'virus', 'corona', 'di', 'china', 'fbi', 'menangkap', 'profesor', 'harvard']”

Text hasil *stemming* dan hasil *preprocessing*:

“['libat', 'buat', 'virus', 'corona', 'china', 'fbi', 'tangkap', 'profesor', 'harvard']”

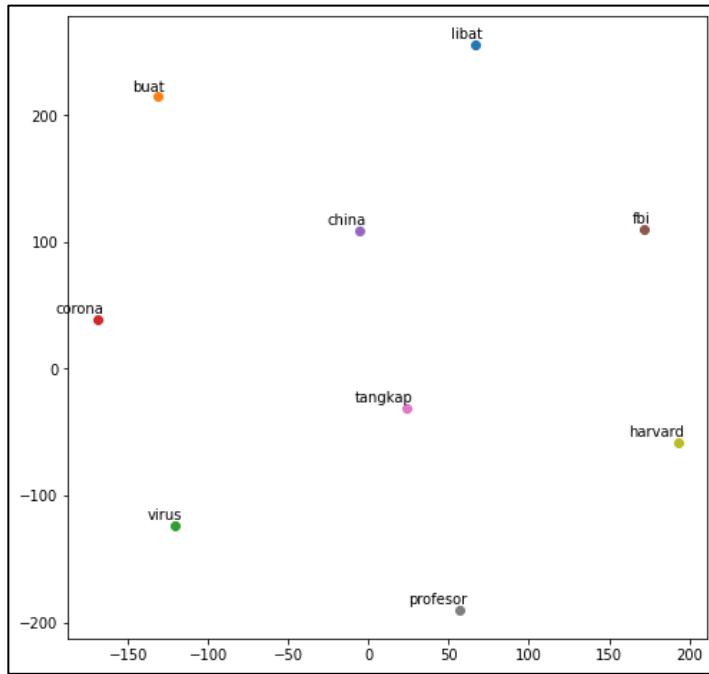
3.1.4 Word Embedding

Pada bagian ini akan dijelaskan terkait proses dari *word embedding* menggunakan *word2vec*. Proses ini akan merubah suatu kata menjadi suatu vektor yang berisi kumpulan angka. Kata yang menjadi nilai *input* pada *word embedding* berasal dari hasil proses *stemming*. Vektor yang dihasilkan dari proses ini akan merepresentasikan dari setiap kata yang ada. Vektor yang dihasilkan memiliki panjang tertentu yang dapat diatur, semakin panjang suatu vektor maka akan semakin teliti representasi yang dihasilkan. Panjang vektor pada proses *word embedding* dapat disebut dengan *dimensi*. Pada penelitian ini, dimensi dari vektor yang dihasilkan yaitu 100. Namun, untuk memudahkan dalam memberikan contoh maka akan ditampilkan vektor dari setiap kata dengan jumlah dimensi 5. Berikut *flowchart* dari proses *word embedding* dapat dilihat pada Gambar 3.12 dan hasil dari proses *word embedding* dapat dilihat pada Tabel 3.4.

**Gambar 3.12 Flowchart Word Embedding****Tabel 3.4 Hasil Word Embedding**

Kata	Dimensi				
	1	2	3	4	5
libat	0.07452912	0.071270086	0.011775045	-0.0683471	-0.00932195
buat	-0.0835347	0.0141729405	-0.028569078	-0.017134266	-0.017073445
virus	-0.045051266	-0.06300815	-0.039852243	0.06623562	-0.07374446
corona	-0.018552037	-0.05871475	-0.0063648676	-0.06412901	-0.085089624
china	0.041457042	-0.015513414	0.038284097	-0.046403337	0.074265204
fbi	-0.061272316	-0.055307325	0.026977174	0.089745976	0.057193857
tangkap	-0.08523997	0.058822744	0.007000594	0.067885585	-0.095681004
profesor	0.08066033	-0.045340743	-0.089614525	-0.0053708707	-0.028951226
harvard	-0.064775534	-0.0042433734	0.05336014	0.084353596	0.006926856

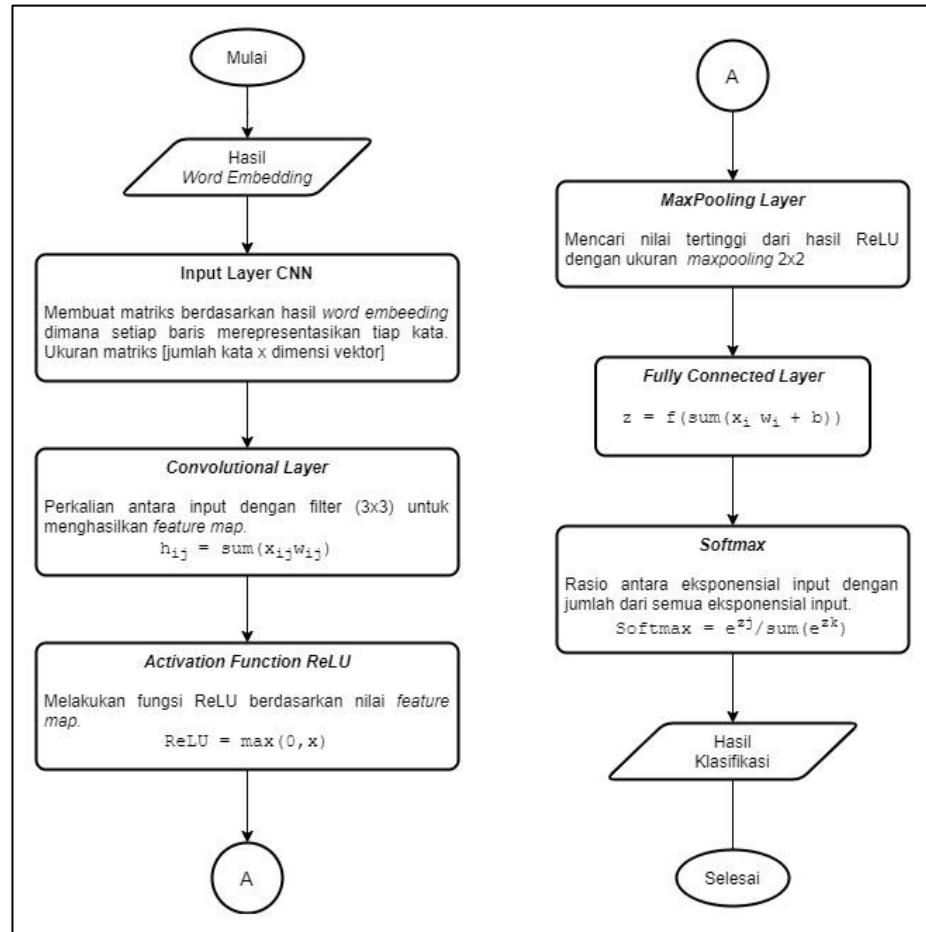
Berdasarkan vektor-vektor tersebut dapat dibuat sebuah visualisasi yang merepresentasikan kedekatan antar kata ada. Untuk melakukan visualisasi dari data hasil word embedding dapat menggunakan model t-SNE. Model tersebut akan mengkonversi kesamaan antara titik data. Berikut hasil visualisasi dari vektor *word embedding* dapat dilihat pada Gambar 3.13.



Gambar 3.13 Visualisasi *Word Embedding*

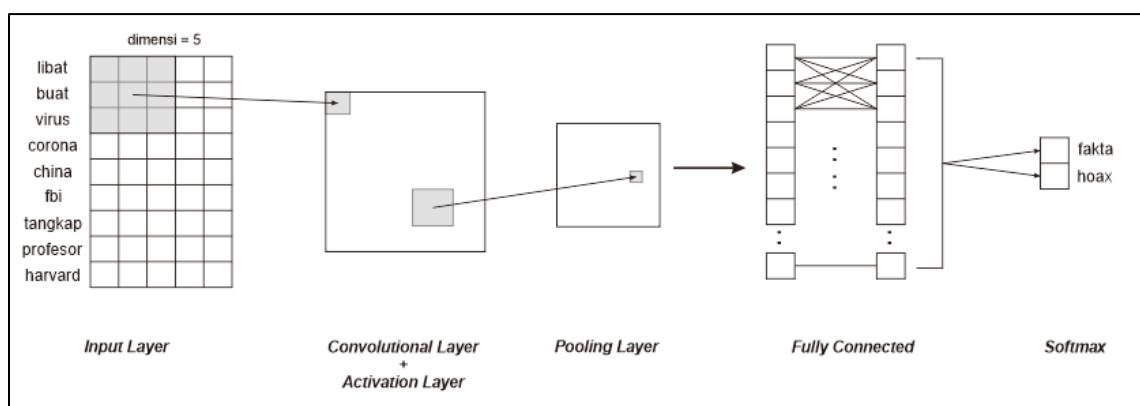
3.1.5 Convolutional Neural Network

Pada bagian ini akan menggambarkan proses klasifikasi menggunakan algoritma *convolutional neural network*. Hasil dari proses *word embedding* akan menjadi data *input* yang akan dimasukkan pada matriks *input layer* dengan setiap barisnya merepresentasikan setiap kata. Setelah itu, *input layer* akan dikalikan dengan filter ukuran 3×3 untuk menghasilkan *feature map* yang merupakan proses dari *convolutional layer* dan masing-masing hasil tersebut dilakukan fungsi aktivasi ReLu. Kemudian, akan dilakukan operasi *pooling* dimana pada penelitian ini akan menggunakan *MaxPooling* dengan ukuran 2×2 . Hasil dari operasi *pooling* tersebut akan masuk ke bagian *fully connected*. Pada bagian ini merupakan *neural network* yang terdiri dari *input layer*, *hidden layer* dan *output layer*. Selanjutnya akan dilakukan proses *softmax* untuk menghasilkan probabilitas dari hasil klasifikasi. Berikut *flowchart* CNN dapat dilihat pada Gambar 3.14.



Gambar 3.14 Flowchart CNN

Berdasarkan *flowchart* tersebut maka dapat dibuat sebuah arsitektur CNN yang akan digunakan pada penelitian ini untuk memperjelas alur dari proses dari algoritma *convolutional neural network*. Berikut arsitektur CNN dapat dilihat pada Gambar 3.15.



Gambar 3.15 Arsitektur CNN

3.1.5.1 Input Layer

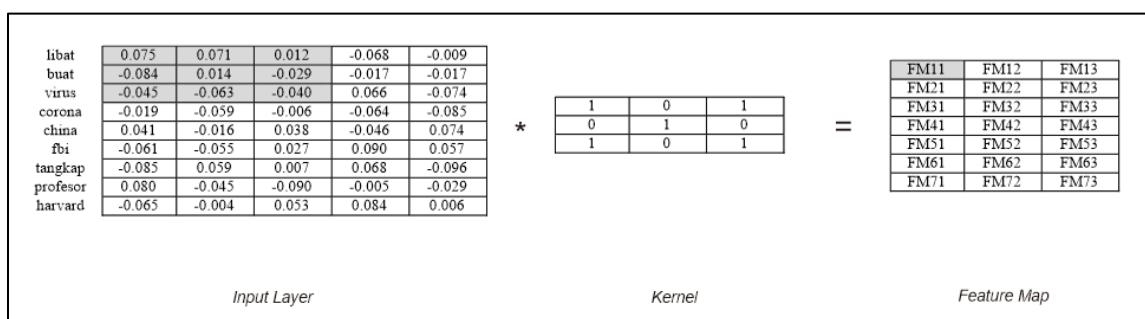
Input layer merupakan proses pertama yang berfungsi sebagai tempat untuk menampung data inputan di algoritma CNN. Data yang tampung merupakan hasil dari proses *word embedding* yang telah berbentuk vektor yang merepresentasikan masing-masing kata. Layer ini berbentuk seperti matriks dengan ukuran banyaknya kata dikali panjang vektor atau dimensi vektor. Apabila datanya sebanyak 9 kata dan masing-masing kata direpresentasikan dengan 5 dimensi vektor, maka layer input berukuran 9×5 . Berikut contoh *input layer* dengan data dari hasil *word embedding* yang dibulatkan untuk memudahkan perhitungan.

Tabel 3.5 Input Layer

libat	0.075	0.071	0.012	-0.068	-0.009
buat	-0.084	0.014	-0.029	-0.017	-0.017
virus	-0.045	-0.063	-0.040	0.066	-0.074
corona	-0.019	-0.059	-0.006	-0.064	-0.085
china	0.041	-0.016	0.038	-0.046	0.074
fbi	-0.061	-0.055	0.027	0.090	0.057
tangkap	-0.085	0.059	0.007	0.068	-0.096
profesor	0.080	-0.045	-0.090	-0.005	-0.029
harvard	-0.065	-0.004	0.053	0.084	0.006

3.1.5.2 Convolutional Layer

Convolutional layer merupakan proses utama dari CNN. Data dari input layer akan menjadi masukan di layer konvolusi ini. Proses dari *convolutional layer* yaitu jumlah dari perkalian antara *input layer* dengan kernel. Kernel ini merupakan array dua dimensi berukuran 3×3 . Kernel atau filter ini akan bergeser ke indeks selanjutnya pada *input layer*. Apabila proses konvolusi telah selesai akan menghasilkan kumpulan vektor yang disebut dengan *feature map*. Berikut proses dari *convolutional layer*.



Gambar 3.16 Proses Convolutional Layer

Matriks *feature map* dihasilkan dari jumlah perkalian *input layer* filter 3×3 dengan kernel pada indeks yang sama. Berdasarkan formula 2.10 untuk menghasilkan FM11 sebagai berikut:

$$\text{FM11} = (0.075*1) + (0.071*0) + (0.012*1) + (-0.084*0) + (0.014*1) + (-0.029*0) + (-0.045*1) + (-0.063*0) + (-0.040*1) = 0.025$$

Berdasarkan perhitungan tersebut dan dilakukan berulang seusai pergeseran filter akan menghasilkan *feature map* seperti pada Tabel 3.6.

Tabel 3.6 Hasil *Feature Map*

0.025	-0.023	-0.128
-0.201	-0.166	-0.071
-0.065	-0.046	-0.066
-0.075	-0.050	-0.053
-0.054	0.092	0.113
0.015	-0.008	0.033
-0.135	0.117	-0.035

3.1.5.3 Activation Layer (*ReLU*)

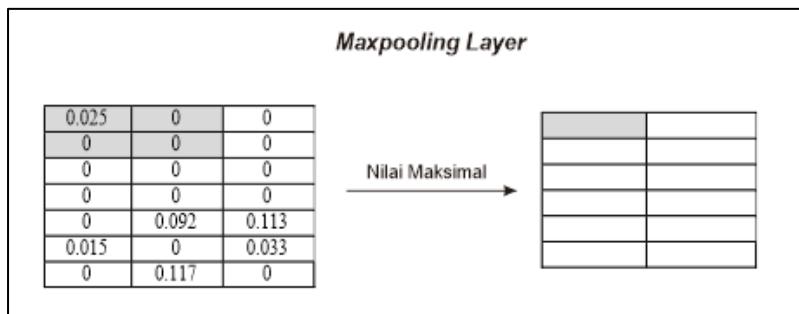
ReLU merupakan salah satu fungsi aktivasi. Proses pada layer aktivasi ini yaitu dengan memasukkan nilai *feature map* yang telah dihasilkan pada tahap sebelumnya ke dalam fungsi *ReLU*. Fungsi aktivasi ini bertujuan untuk mengubah nilai pada *feature map* pada range tertentu dan meneruskan nilai yang menjadi fitur dominan layer selanjutnya. Fungsi *ReLU* yaitu $\max(0, x)$. Jadi apabila nilai *feature map* kurang dari 0 maka hasil dari fungsi *ReLU* sama dengan 0. Tetapi, jika nilainya lebih dari 0 maka hasil dari fungsi *ReLU*nya nilai itu sendiri. Hasil proses *ReLU* dari data *feature map* sebelumnya dapat dilihat pada Tabel 3.7.

Tabel 3.7 Hasil *ReLU*

0.025	0	0
0	0	0
0	0	0
0	0	0
0	0.092	0.113
0.015	0	0.033
0	0.117	0

3.1.5.4 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 Pada penelitian ini menggunakan *maxpooling*. Proses pada *maxpooling* dimulai dengan menentukan ukuran *downsampling*, misalnya 2×2 . Setelah itu akan dilakukan proses *pooling* pada *feature map* dengan mengambil nilai tertingginya. Proses *maxpooling* dapat dilihat pada Gambar 3.17.



Gambar 3.17 Proses *MaxPooling Layer*

Proses *maxpooling* dilakukan seperti layer konvolusi dengan filter 2×2 bergeser secara berulang. Apabila telah dilakukan hingga indeks *feature map* terakhir maka akan dihasilkan *feature map* baru yang telah melalui proses *maxpooling*. Hasil *feature map* baru tersebut dapat dilihat pada Tabel 3.8.

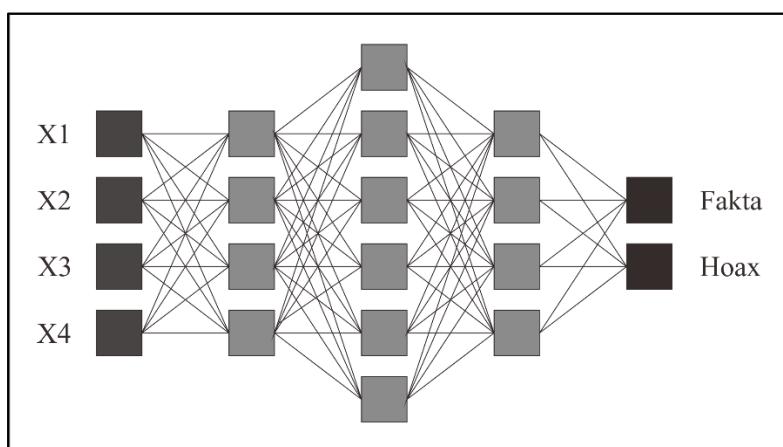
Tabel 3.8 Hasil *MaxPooling Layer*

0.025	0
0	0
0	0
0	0
0	0.092
0.015	0
0	0.117

0	0
0	0
0	0
0.092	0.113
0.092	0.113
0.117	0.117

3.1.5.5 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 terhubung dengan *neuron-neuron* pada layer tetangganya. Berikut bentuk dari *full connected layer* dapat dilihat pada Gambar 3.18:



Gambar 3.18 Fully Connected Layer

3.1.5.6 Softmax

Layer ini akan menentukan hasil klasifikasi berdasarkan data yang telah dimasukkan. Proses penentuan hasil klasifikasi menggunakan fungsi aktivasi *softmax*. Fungsi aktivasi *softmax* yaitu rasio eksponensial dari nilai input dengan jumlah nilai eksponensial. Nilai probabilitas tertinggi menjadi hasil klasifikasi kelas yang diprediksi. Hasil klasifikasi dapat dilihat pada Tabel 3.9.

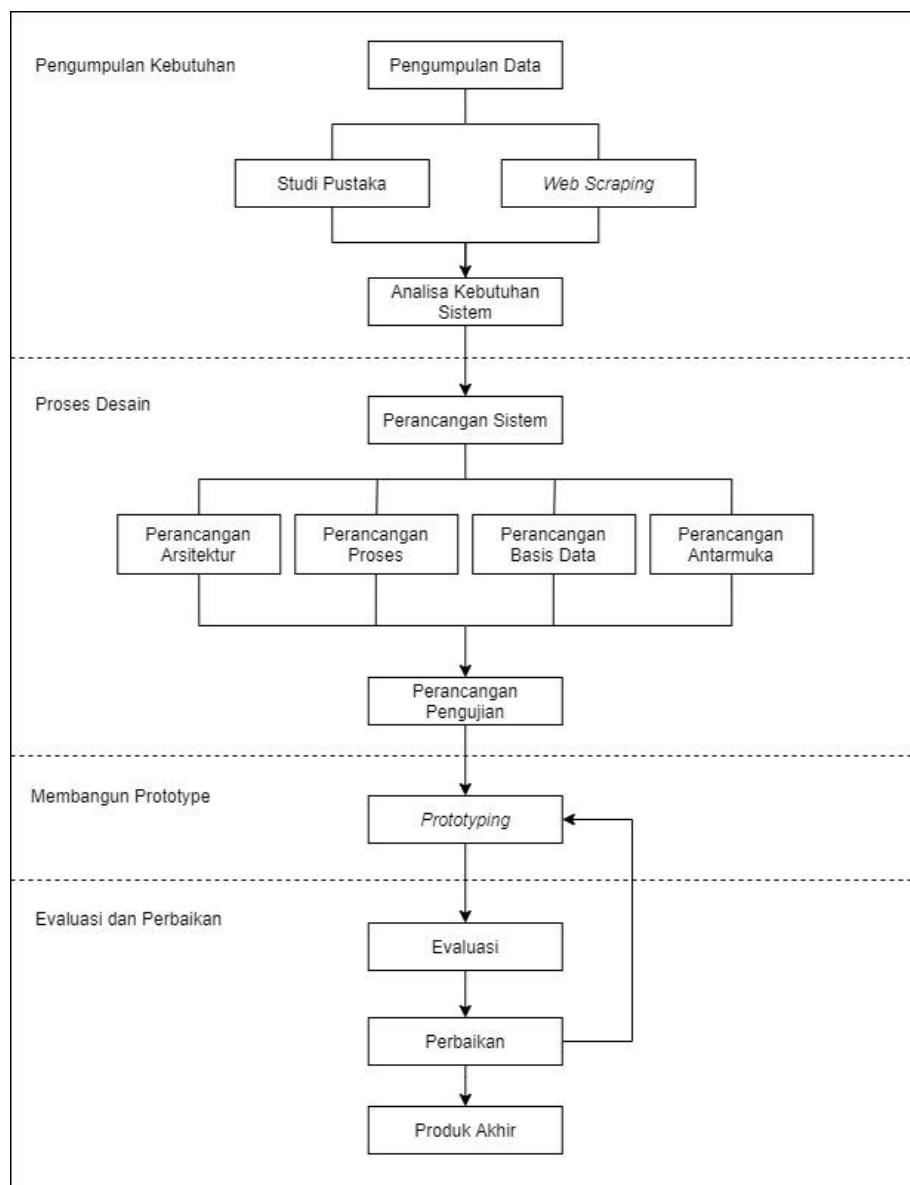
Tabel 3.9 Hasil Klasifikasi

0.352	Fakta
0.648	Hoax

3.2 Metodologi Pengembangan Sistem

Metode pengembangan sistem yang digunakan pada penelitian ini adalah metode *prototype*. Penggunaan metode *prototype* ini karena pada metode ini lebih menekankan pada komunikasi antara pengembang dan pelanggan. Selanjutnya kebutuhan dari pengguna

diterjemahkan dalam bentuk model (*prototype*). Ada beberapa kelebihan dari penggunaan metode *prototype* yaitu terjalin hubungan yang baik antara pengguna dan pengembang, menghemat waktu dan biaya dalam pengembangan sistem, serta pelanggan berperan aktif dalam proses pengembangan sistem sehingga hasil yang didapatkan akan lebih baik karena sesuai dengan keinginan pelanggan. Metode *prototype* terdiri dari empat tahapan yaitu pengumpulan kebutuhan, proses desain yang cepat, membangun prototipe serta evaluasi dan perbaikan (Purnomo, 2017). Metodologi pengembangan sistem yang dibuat dapat dilihat pada Gambar 3.19.



Gambar 3.19 Metodologi Pengembangan Sistem

Pada tahapan pengumpulan dilakukan pengumpulan data yang bersumber dari studi pustaka dan *web scraping*, dilanjutkan dengan analisa dari data tersebut serta analisa kebutuhan sistem. Kemudian, pada bagian proses desain akan membuat rancangan sistem seperti rancangan arsitektur, proses basis data dan rancangan antarmuka. Selanjutnya membangun *prototype* berdasarkan rancangan yang telah dibuat dengan penulisan program (*coding*). *Prototype* yang dihasilkan akan diuji untuk mengetahui tingkat performa dari algoritma *Convolutional Neural Network* dalam menangai permasalahan klasifikasi berita *hoax*, lalu akan dievaluasi dan dilakukan perbaikan sesuai kebutuhan.

3.2.1 Pengumpulan Kebutuhan

Pada bagian akan dibahas mengenai hal-hal yang dibutuhkan baik berupa data maupun informasi lain untuk dianalisis dan digunakan dalam penelitian ini. Pengumpulan kebutuhan terdiri dari beberapa tahapan, yaitu pengumpulan data, analisa data dan analisa kebutuhan sistem.

3.2.1.1 Pengumpulan Data

Data merupakan salah satu hal yang paling penting dalam sebuah penelitian. Tanpa adanya data, suatu penelitian tidak dapat dilakukan. Penelitian memerlukan data untuk diolah sehingga menjadi sebuah informasi yang dapat membantu untuk pengambilan keputusan atau menghasilkan teori. Maka dari itu, data yang diperlukan dalam penelitian harus dikumpulkan terlebih dahulu. Untuk melakukan proses pengumpulan data, ada beberapa metode yang dapat digunakan. Metode pengumpulan data yang dilakukan pada penelitian ini terdiri dari dua bagian, yaitu studi pustaka dan *web scraping*.

3.2.1.2 Analisa Kebutuhan Sistem

Pada bagian ini akan dilakukan analisis terhadap kebutuhan-kebutuhan yang perlu disiapkan untuk mengembangkan sistem pada penelitian ini. Analisis ini bertujuan untuk mendefinisikan masalah dan menyiapkan sistem sehingga kendala yang berkaitan dengan

sumber daya dapat diminimalisir. Analisis kebutuhan sistem ini terbagi menjadi dua bagian yaitu kebutuhan fungsional dan kebutuhan non-fungsional.

a. Kebutuhan Fungsional

Kebutuhan fungsional adalah kebutuhan-kebutuhan yang berkaitan langsung dengan sistem. Pada kebutuhan fungsional akan memuat gambaran besar dari keseluruhan proses yang dapat dilakukan oleh sistem. Berikut adalah kebutuhan fungsional yang akan dilakukan oleh sistem:

- a. Sistem dapat melakukan proses *preprocessing* terhadap dataset.
- b. Sistem dapat melakukan proses *word embedding*.
- c. Sistem dapat melakukan proses pembuatan model klasifikasi dengan metode *Convolutional Neural Network*.
- d. Sistem dapat melakukan proses *training* dan *testing* menggunakan model klasifikasi.
- e. Sistem dapat melakukan proses klasifikasi berita.
- f. Sistem dapat menampilkan hasil klasifikasi berita *hoax* atau bukan.
- g. Sistem dapat melakukan proses *login* dan *logout* untuk *admin*.
- h. Sistem dapat memperbarui model klasifikasi dengan *training* data baru yang dilakukan melalui *admin*.
- i. Sistem dapat melakukan pengecekan dan pelaporan berita *hoax* melalui *user*.

b. Kebutuhan Non-Fungsional

Kebutuhan non-fungsional adalah kebutuhan yang tidak berkaitan langsung dengan fitur yang terdapat pada sistem. Pada kebutuhan non-fungsional terdiri dari kebutuhan perangkat keras (*hardware*) dan perangkat lunak (*software*).

1. Perangkat Keras (*Hardware*)

Perangkat keras yang dibutuhkan untuk mengembangkan sistem pada penelitian ini memiliki spesifikasi seperti pada Tabel 3.10.

Tabel 3.10 Kebutuhan Perangkat Keras

No	Perangkat Keras	Keterangan
1	<i>Processor</i>	Intel Core i3-5005U 3th Gen
2	<i>RAM</i>	4GB DDR4
3	<i>Storage</i>	Harddisk 500GB
4	<i>Graphic</i>	Intel HD Graphics
5	Perangkat <i>Input</i> dan <i>Output</i>	<i>Keyboard, Mouse</i> dan <i>Monitor</i>
6	Koneksi Internet	Wifi atau Kuota Internet

2. Perangkat Lunak (*Software*)

Adapun perangkat lunak yang dibutuhkan untuk mengembangkan sistem pada penelitian ini dapat dilihat pada Tabel 3.11.

Tabel 3.11 Kebutuhan Perangkat Lunak

No	Perangkat Lunak	Keterangan
1	Operating System	Windows 10 dengan 64 bit
2	Bahasa Python	Bahasa pemrograman
3	Bahasa PHP	Bahasa pemrograman
4	Sublime Text	Perangkat lunak untuk <i>code editor</i>
5	Google Colab	<i>Code editor</i> berbasis online
6	XAMPP	Perangkat lunak sebagai server lokal
7	MySQL	Tempat penyimpanan database
8	Google Chrome	<i>Web browser</i>
9	Draw.io	Website untuk membuat desain diagram
10	CorelDraw	Perangkat lunak untuk rancangan <i>interface</i>

3.2.2 Proses Desain

Proses desain merupakan tahapan yang membahas mengenai proses pembuatan rancangan sementara untuk disajikan kepada pelanggan berdasarkan hasil komunikasi antara pengembang dan pelanggan. Pembuatan rancangan ini bertujuan untuk memberikan gambaran dan memudahkan proses pembuatan sistem yang akan dibangun. Pada proses desain ini akan dibuat perancangan sistem dan perancangan pengujian.

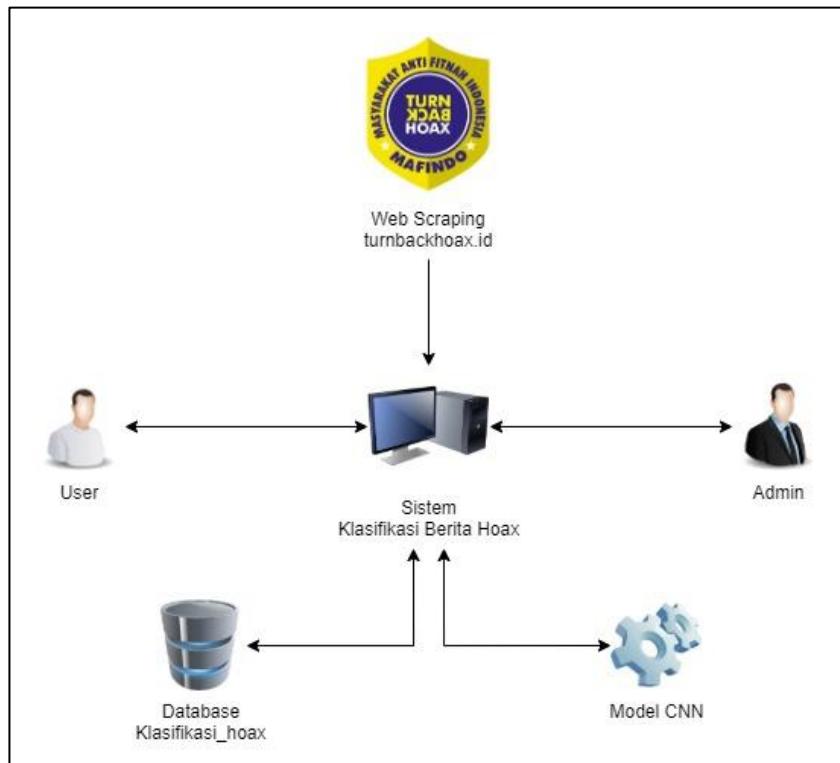
3.2.2.1 Perancangan Sistem

Pada bagian ini akan dibahas tentang rancangan sistem yang akan dibangun. Beberapa perancangan yang akan dibuat untuk membangun sistem klasifikasi berita *hoax* antara lain perancangan arsitektur, perancangan proses, perancangan basis data dan perancangan antarmuka.

a. Perancangan Arsitektur

Arsitektur sistem yang akan dibuat terdiri dari pengguna, database, proses *scraping*, model CNN dan sistem itu sendiri yang akan melakukan klasifikasi berita *hoax*. Terdapat dua aktor pengguna pada sistem ini yaitu *user* dan *admin*. *User* merupakan semua orang secara umum yang menggunakan sistem ini. *User* dapat memberikan *input* kepada sistem berupa konten berita yang ingin diperiksa kebenarannya dan membuat laporan apabila menemukan berita *hoax* di media sosial. Sedangkan *admin* merupakan orang tertentu yang memiliki hak akses khusus dan memiliki hak yang lebih pula di sistem. *Admin* dapat mengolah berbagai data seperti data admin, data cek hoax, data lapor hoax, data dataset, data training, data testing dan data pengujian. Selain itu, *admin* juga dapat melakukan pengecekan berita *hoax* dan membuat laporan *hoax*.

Komponen lain yaitu sistem klasifikasi berita hoax. Semua proses akan terjadi di sistem. Sistem mendapatkan data berasal dari proses *scraping* dari web turnbackhoax.id yang setelah itu diolah menggunakan *text preprocessing* sehingga akan menghasilkan suatu dataset yang siap digunakan. Sistem akan melakukan proses klasifikasi berdasarkan model CNN yang telah dibuat sebelumnya. Model tersebut telah melalui proses training dan testing menggunakan algoritma *Convolutional Neural Network* (CNN). Setelah itu, sistem akan memberikan hasil klasifikasinya kepada *user* atau *admin*. Sistem juga akan memberikan *output* berupa data-data yang berasal dari *database* ataupun menyimpan data ke *database* berdasarkan aksi yang dilakukan oleh *admin*. Ilustrasi dari arsitektur sistem tersebut dapat dilihat pada Gambar 3.20.



Gambar 3.20 Arsitektur Sistem

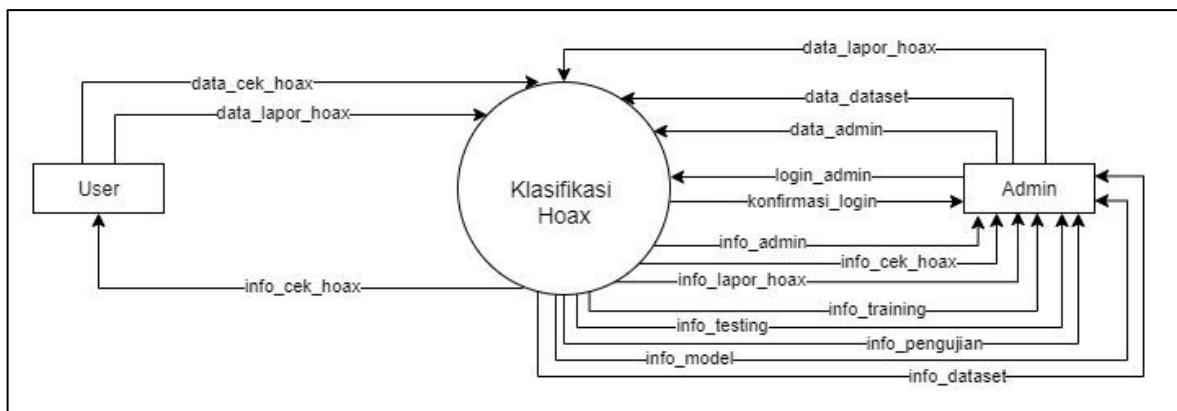
b. Perancangan Proses

Pada tahap perancangan proses ini akan menggambarkan alur dari proses yang ada pada sistem. Perancangan proses ini digambarkan *Data Flow Diagram* (DFD) dari level 0 hingga level 2 serta *flowchart* pada beberapa proses.

1. Data Flow Diagram Level 0

Data Flow Diagram (DFD) *Level 0* merupakan gambaran besar dari proses beserta alur pergerakan data yang ada pada sistem ini. Pada penelitian ini terdapat dua pengguna yang berinteraksi dengan sistem yaitu *user* dan *admin*. Dimana masing-masing pengguna tersebut memiliki batasan yang berbeda terkait *input* dan *output* pada sistem. Pada bagian *user* dapat memberikan *input* kepada sistem berupa data cek hoax dan data lapor hoax. Selain itu, *user* juga dapat menerima *output* dari sistem berupa informasi cek hoax. Sedangkan dari sisi *admin* mempunyai hak yang lebih besar terhadap sistem. Karena *admin* dapat memberikan *input* kepada sistem berupa login admin, data admin, data dataset dan

data lapor hoax. *Admin* juga menerima *output* dari sistem berupa konfirmasi login, informasi admin, informasi cek hoax, informasi lapor hoax, informasi training, informasi testing, informasi pengujian, informasi model dan informasi dataset. Gambaran dari *data flow diagram level 0* dapat dilihat pada Gambar 3.21.



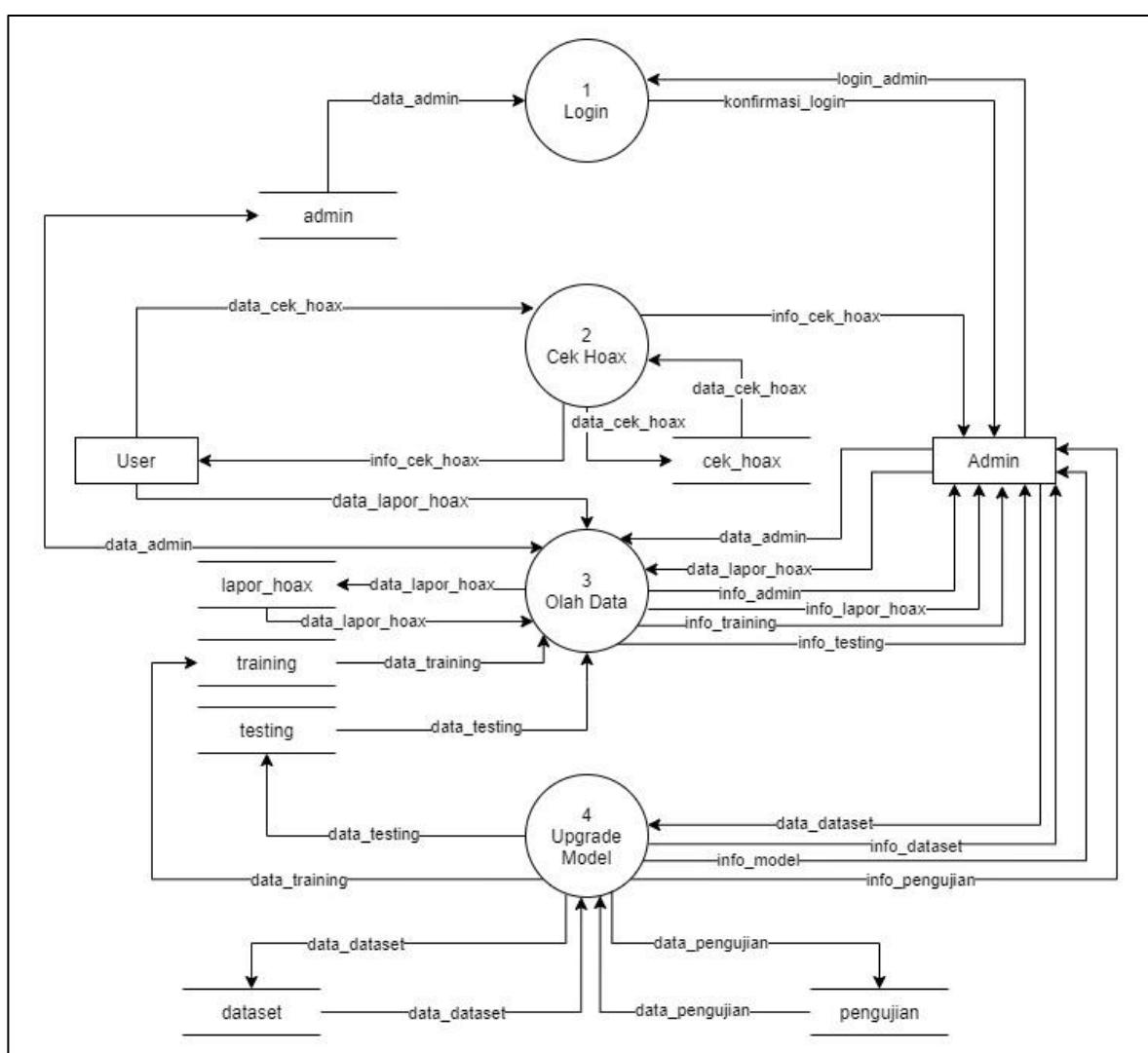
Gambar 3.21 Data Flow Diagram Level 0

2. Data Flow Diagram Level 1

Data Flow Diagram (DFD) Level 1 merupakan turunan dari DFD level 0 yang akan menjabarkan semua proses yang terdapat pada sistem. DFD level 1 terdiri dari empat proses yaitu proses *login*, proses cek hoax, proses olah data dan proses *upgrade* model. Setiap proses akan menggambarkan aliran data serta interaksi dengan pengguna (*user* dan *admin*) dan atau basis data.

Pada proses *login*, sistem akan menerima *input* dari *admin* berupa login admin dan data admin dari tabel admin. Lalu, proses *login* akan mengeluarkan *output* berupa konfirmasi login kepada admin. Proses kedua pada sistem ini yaitu proses cek hoax. Proses cek hoax dapat menerima *input* dari *user* berupa data cek hoax dan menyimpan data tersebut pada tabel cek hoax. Selain itu, proses cek hoax akan mengeluarkan *output* berupa informasi cek hoax kepada *admin* dan *user*. Selanjutnya proses olah data yang dapat menerima *input* data admin dari *admin* dan data lapor hoax dari *admin* dan *user* lalu disimpan pada tabel lapor hoa. Proses olah data juga akan mengambil data dari tabel training dan tabel testing

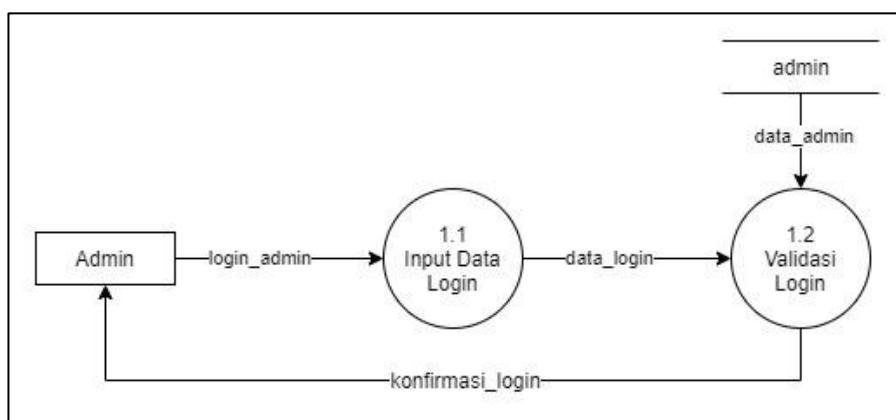
untuk memberikan *output* berupa informasi training dan testing, serta informasi admin dan lapor hoax kepada *admin*. Terakhir, proses *upgrade* model yang akan menerima *input* data dataset dari *admin* lalu membagi dataset tersebut menjadi data training dan testing untuk diproses dan disimpan di tabel training dan testing. Proses *upgrade* model akan melakukan pengujian yang hasilnya akan disimpan pada tabel pengujian. Proses ini akan mengeluarkan *output* berupa informasi dataset, informasi model dan informasi pengujian. Berikut *data flow diagram level 1* yang dapat dilihat pada Gambar 3.22.



Gambar 3.22 Data Flow Diagram Level 1

3. Data Flow Diagram Level 2 Proses Login

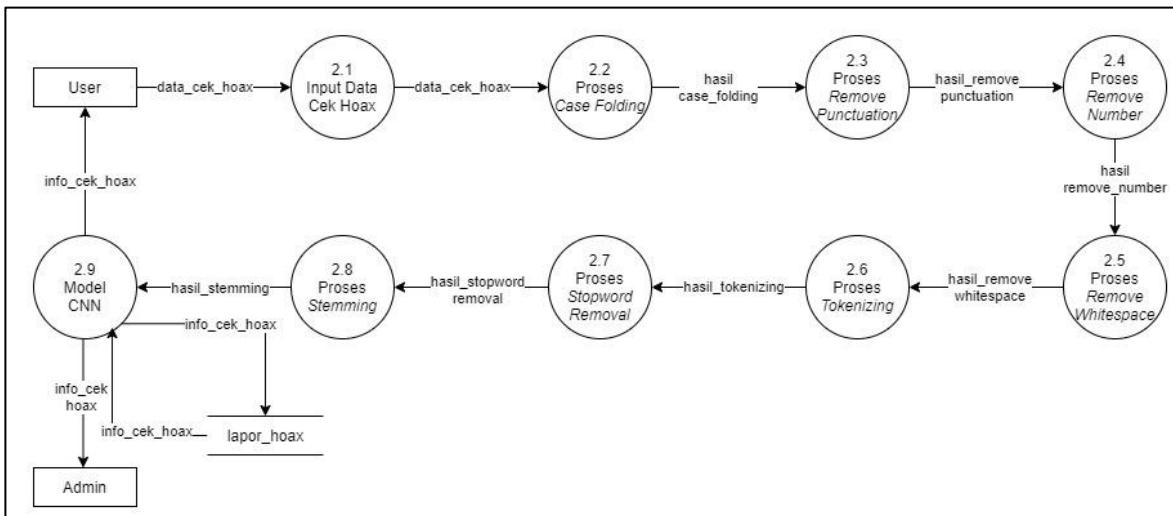
Data Flow Diagram (DFD) Level 2 Proses Login merupakan penjelasan lebih rinci dari proses login di DFD level 1. Pada DFD level 2 proses login, *admin* akan mengirimkan login admin pada sub proses input data login dan sub proses tersebut akan menghasilkan data login yang akan dikirim ke subproses validasi login. Pada sub proses validasi logi akan mengambil data dari tabel admin untuk memvalidasi data login. Hasil validasi tersebut akan dikeluarkan oleh subproses validasi login berupa konfirmasi login kepada *admin*.



Gambar 3.23 Data Flow Diagram Level 2 Proses Login

4. Data Flow Diagram Level 2 Proses Cek Hoax

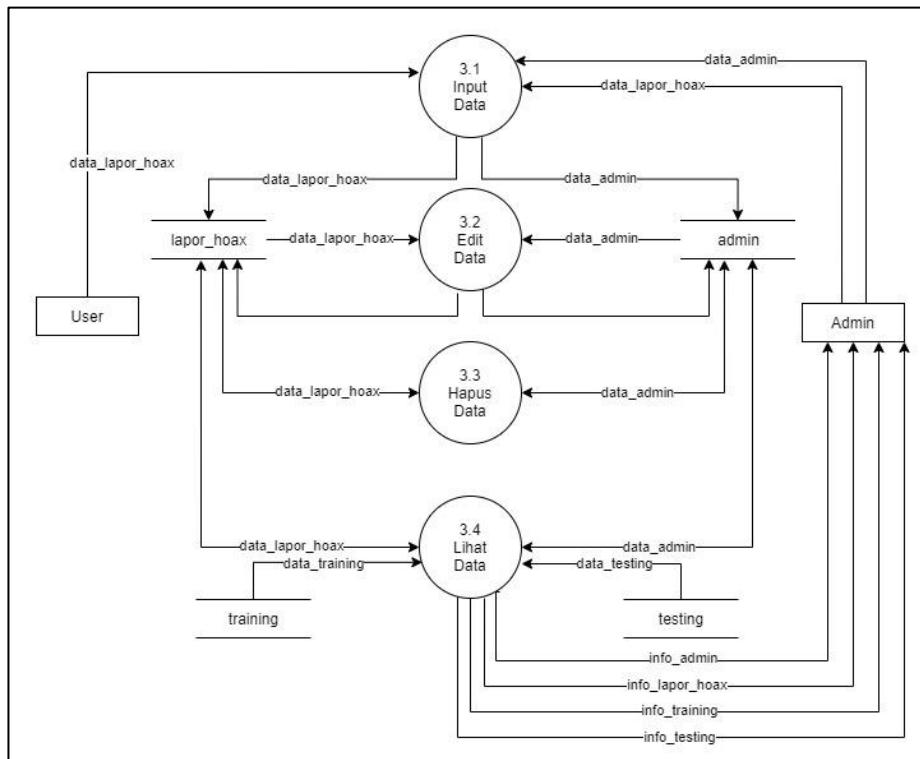
Data Flow Diagram (DFD) Level 2 Proses Cek Hoax merupakan penjelasan lebih rinci dari proses cek hoax pada DFD level 1. Pada DFD level 2 proses cek hoax akan menerima *input* berupa data cek hoax dan memiliki beberapa sub proses diantaranya *input data hoax*, *case folding*, *remove punctuation*, *remove number*, *remove whitespace*, *tokenizing*, *stopword*, *stemming* dan model CNN. Pada sub proses model CNN akan menghasilkan informasi cek hoax yang akan dikeluarkan ke *user* dan *admin* serta informasi tersebut di tabel cek hoax. DFD Level 2 Proses Cek Hoax dapat dilihat pada Gambar 3.24.



Gambar 3.24 Data Flow Diagram Level 2 Proses Cek Hoax

5. Data Flow Diagram Level 2 Proses Olah Data

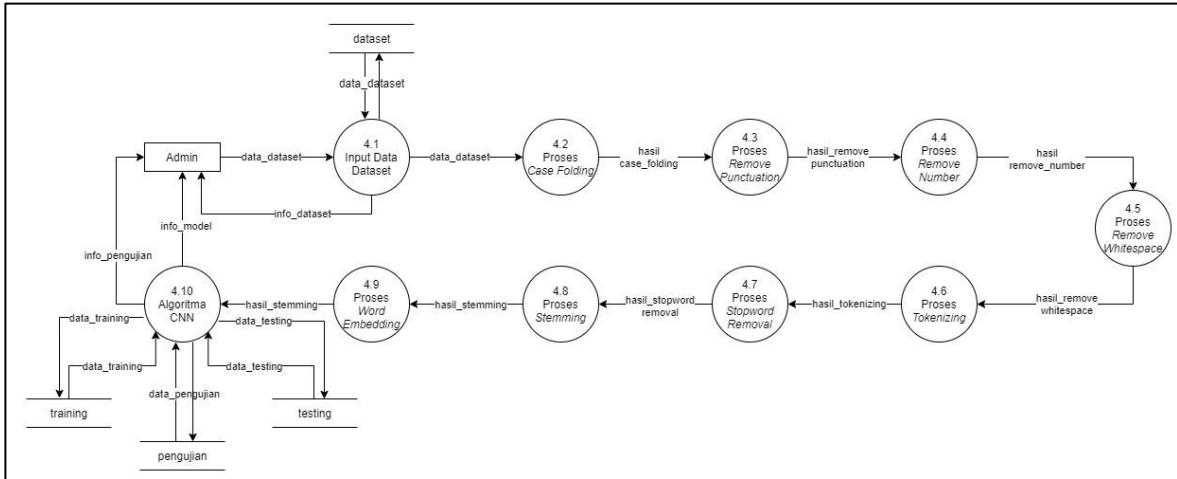
Data Flow Diagram (DFD) Level 2 Proses Olah Data adalah penjabaran yang lebih rinci dari proses olah data di DFD level 1. Pada DFD level 2 proses olah data terdapat empat sub proses yang memiliki fungsi masing-masing. Pertama, sub proses input data yang akan menerima *input* dari *admin* berupa data admin dan data lapor hoax serta data lapor hoax dari *user*. Kemudian data admin dan data lapor hoax tersebut akan disimpan di tabel admin dan tabel lapor hoax. Kedua, sub proses edit data. Sub proses ini akan mengambil data dari tabel admin dan tabel lapor hoax untuk diubah datanya dan akan disimpan kembali pada tabel masing-masing. Ketiga, sub proses hapus data yang akan mengambil data dan menghapusnya dari tabel admin dan atau tabel lapor hoax. Terakhir, sub proses lihat data. Sub proses ini mengambil data dari tabel admin, tabel lapor hoax, tabel training dan testing yang selanjutnya mengeluarkan *output* berupa informasi admin, informasi lapor hoax, informasi training dan informasi testing. DFD level 2 proses olah data ini dapat dilihat pada Gambar 3.25



Gambar 3.25 Data Flow Diagram Level 2 Proses Olah Data

6. Data Flow Diagram Level 2 Proses Upgrade Model

Data Flow Diagram (DFD) Level 2 Proses Upgrade Model merupakan penjabaran dari proses *upgrade* model pada DFD level 1. Pada bagian ini akan menerima *input* dari *admin* berupa data dataset dan memberikan *output* informasi dataset kepada *admin*. Data dataset akan diproses pada beberapa sub proses yang terdiri dari *case folding*, *remove punctuation*, *remove number*, *remove whitespace*, *tokenizing*, *stopword*, *stemming*, *word embedding*, dan algoritma CNN. Hasil dari sub proses tersebut berupa data training, data testing dan data pengujian yang akan disimpan di masing-masing tabel serta informasi pengujian dan informasi model yang akan diberikan kepada *admin*. DFD level 2 proses *upgrade* model dapat dilihat pada Gambar 3.26.



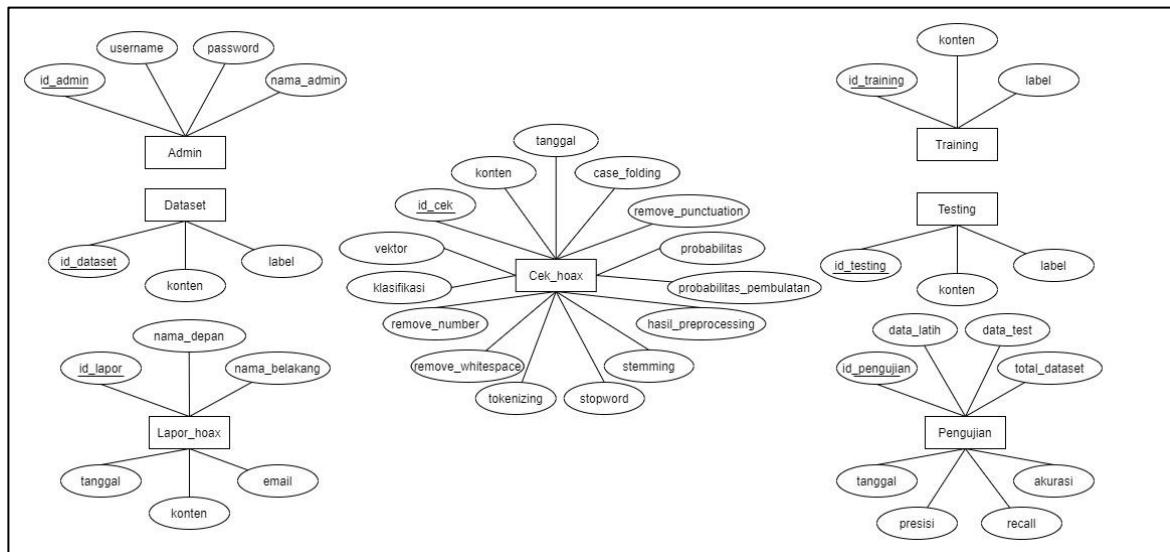
Gambar 3.26 Data Flow Diagram Level 2 Proses Upgrade Model

3.2.2.2 Perancangan Basis Data

Dalam memenuhi kebutuhan sistem, maka diperlukan suatu basis data yang berfungsi sebagai tempat penyimpanan seluruh data yang dibutuhkan. Data-data yang akan disimpan seperti data admin, data cek hoax, dataset, data training dan data testing, data laporan serta data pengujian. Basis data pada sistem ini menggunakan *database MySQL* dan *Server HTTP Apache*. Tetapi sebelum membuat basis data, ada beberapa rancangan yang perlu dipersiapkan terlebih dahulu diantaranya perancangan *Entity Relationship Diagram* (ERD), Relasi Antar Tabel (RAT) dan Struktur Tabel.

a. *Entity Relationship Diagram (ERD)*

Entity Relationship Diagram (ERD) adalah suatu model yang bertujuan untuk menjelaskan hubungan antar data dalam basis data. Proses ini akan memodelkan struktur data dan hubungan antar datanya dengan menggunakan notasi dan simbol. Ada tiga komponen utama pada ERD yaitu entitas (objek), atribut (karakteristik objek), dan relasi (hubungan antar objek). Rancangan ERD pada Penerapan *Deep Learning* Untuk Klasifikasi Berita Hoax Menggunakan *Convolutional Neural Network* dapat dilihat pada Gambar 3.27.



Gambar 3.27 ERD Klasifikasi Berita Hoax

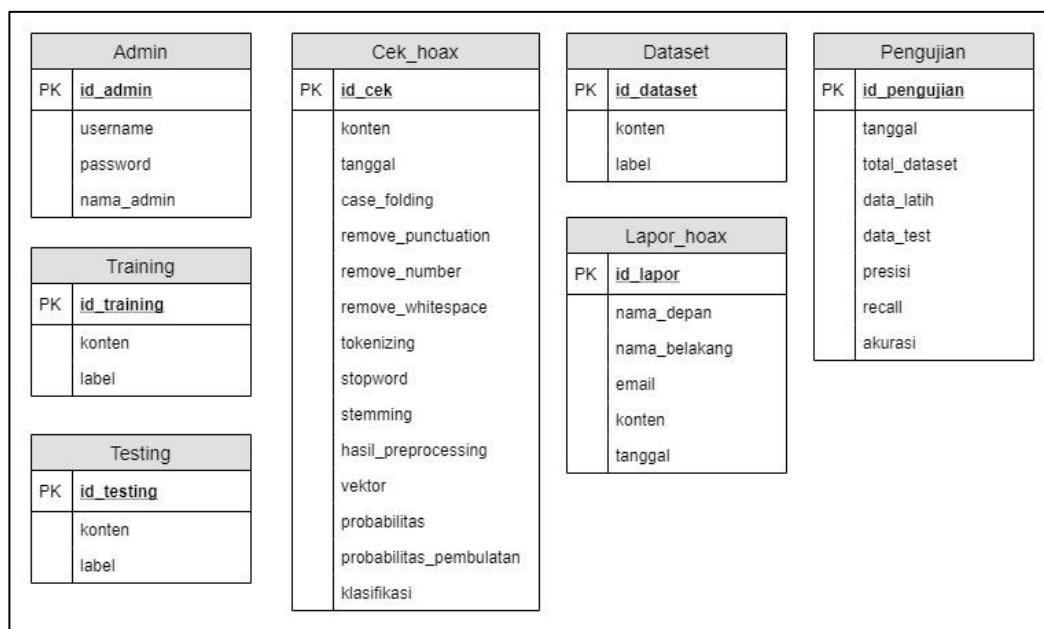
ERD dalam sistem ini terdapat tujuh entitas yaitu admin, cek hoax, dataset, lapor hoax, training, testing dan pengujian yang masing-masing entitas akan dijabarkan sebagai berikut:

1. Entitas admin, entitas ini memiliki empat atribut diantaranya id admin, username, password dan nama admin. Id admin merupakan *primary key*.
2. Entitas cek hoax, entitas ini memiliki beberapa atribut yaitu id cek, konten, tanggal, case folding, remove punctuation, remove number, remove whitespace, tokenizing, stopword, stemming, hasil preprocessing, vektor, probabilitas, probabilitas pembulatan dan klasifikasi dimana id cek sebagai *primary key*.
3. Entitas dataset, entitas ini memiliki tiga atribut yaitu id dataset yang merupakan *primary key*, konten dan label.
4. Entitas lapor hoax, entitas ini memiliki beberapa atribut seperti id lapor, nama depan, nama belakang, email, konten dan tanggal. Id lapor merupakan *primary key*.
5. Entitas training, entitas ini terdapat tiga atribut diantaranya id training, konten dan label dimana id training berperan sebagai *primary key*.

6. Entitas testing, entitas ini mempunyai tiga atribut antara lain id testing, konten dan label dimana id testing berperan sebagai *primary key*.
7. Entitas pengujian, entitas ini memiliki beberapa atribut yaitu id pengujian, tanggal, total dataset, data latih, data test, presisi, recall dan akurasi. Id pengujian pada entitas ini adalah *primary key*.

b. Relasi Antar Tabel (RAT)

Relasi Antar Tabel (RAT) akan merepresentasikan hubungan antar tabel yang berfungsi untuk mengatur operasi suatu basis data. Rancangan RAT pada Penerapan *Deep Learning* Untuk Klasifikasi Berita *Hoax* Menggunakan *Convolutional Neural Network* dapat dilihat pada Gambar 3.28.



Gambar 3.28 RAT Klasifikasi Berita *Hoax*

c. Struktur Tabel

Basis data dalam penelitian ini terdiri dari beberapa tabel yang masing-masing tabel dapat dilihat sebagai berikut:

1. Tabel Admin

Tabel admin berfungsi untuk menyimpan semua data yang berhubungan dengan admin. Data tersebut digunakan sebagai hak akses untuk masuk ke sistem. Pada tabel admin mempunyai 4 buah atribut yaitu *id_admin*, *username*, *password* dan *nama_admin* dimana *id_admin* sebagai *primary key*. Struktur tabel admin dapat dilihat pada Tabel 3.12.

Tabel 3.12 Struktur Tabel Admin

No	Nama Field	Tipe Data	Constraint	Keterangan
1	<i>id_admin</i>	Int (11)	<i>Primary Key, Auto Increment</i>	Id admin
2	<i>username</i>	Varchar (50)	<i>None</i>	<i>Username admin</i>
3	<i>password</i>	Varchar (50)	<i>None</i>	<i>Password admin</i>
4	<i>nama_admin</i>	Varchar (50)	<i>None</i>	Nama admin

2. Tabel Cek Hoax

Tabel cek hoax berfungsi untuk menyimpan data berupa berita yang ingin dicek atau diprediksi kebenarannya. Selain itu, tabel ini juga berisi hasil dari setiap tahapan dalam proses prediksi. Tabel cek hoax terdiri dari 15 atribut yang salah satunya sebagai *primary key* yaitu *id_cek*. Struktur tabel cek hoax dapat dilihat pada Tabel 3.13.

Tabel 3.13 Struktur Tabel Cek Hoax

No	Nama Field	Tipe Data	Constraint	Keterangan
1	<i>id_cek</i>	Int (11)	<i>Primary Key, Auto Increment</i>	Id cek
2	Konten	Text	<i>None</i>	Konten yang ingin dicek
3	Tanggal	Date	<i>None</i>	Tanggal pengecekan
4	<i>case_folding</i>	Text	<i>None</i>	Hasil case folding
5	<i>remove_punctuation</i>	Text	<i>None</i>	Hasil remove punctuation
6	<i>remove_number</i>	Text	<i>None</i>	Hasil remove number
7	<i>remove_whitespace</i>	Text	<i>None</i>	Hasil remove whitespace
8	<i>tokenizing</i>	Text	<i>None</i>	Hasil tokenizing
9	<i>stopword</i>	Text	<i>None</i>	Hasil stopword
10	<i>stemming</i>	Text	<i>None</i>	Hasil stemming
11	<i>hasil_preprocessing</i>	Text	<i>None</i>	Hasil preprocessing
12	<i>vektor</i>	Text	<i>None</i>	Hasil vektorisasi
13	<i>probabilitas</i>	Varchar (50)	<i>None</i>	Probabilitas prediksi
14	<i>probabilitas_pembulatan</i>	Varhcar (50)	<i>None</i>	Probabilitas dibulatkan
15	<i>klasifikasi</i>	Varchar (10)	<i>None</i>	Hasil klasifikasi cek hoax

3. Tabel Dataset

Tabel dataset berfungsi untuk menyimpan data yang akan digunakan sebagai dataset saat admin ingin melakukan proses *upgrade* model prediksi. Tabel dataset terdiri dari tiga

atribut yaitu id_dataset, konten dan label. Struktur dari tabel dataset dapat dilihat lebih jelas pada Tabel 3.14.

Tabel 3.14 Struktur Tabel Dataset

No	Nama Field	Tipe Data	Constraint	Keterangan
1	id_dataset	Int (11)	Primary Key, Auto Increment	Id dataset
2	konten	Text	None	Konten dataset
3	label	Varchar (10)	None	Label konten dataset

4. Tabel Lapor Hoax

Tabel lapor hoax merupakan tabel yang berisi kumpulan data dari laporan *user*. Jika *user* menemukan suatu berita *hoax* atau fakta maka *user* dapat melaporkan berita tersebut melalui sistem dan data laporannya akan disimpan pada tabel ini. Atribut dari tabel lapor hoax antara lain id_lapor, nama_depan, nama_belakang, email, konten dan tanggal. Id_lapor merupakan atribut dari tabel lapor hoax yang bertindak sebagai *primary key*. Untuk lebih jelasnya, tabel lapor hoax dapat dilihat pada Tabel 3.15.

Tabel 3.15 Struktur Tabel Lapor Hoax

No	Nama Field	Tipe Data	Constraint	Keterangan
1	id_lapor	Int (11)	Primary Key, Auto Increment	Id lapor
2	nama_depan	Varchar (50)	None	Nama depan pelapor
3	nama_belakang	Varchar (50)	None	Nama belakang pelapor
4	email	Varchar(100)	None	Email pelapor
5	konten	Text	None	Isi konten laporan
6	tanggal	Date	None	Tanggal pelaporan

5. Tabel Training

Tabel training berisi dataset yang bertindak sebagai data training saat pembuatan model atau saat *upgrade* model prediksi. Maka dari itu, setelah data pada tabel dataset digunakan untuk *upgrade* model maka data pada tabel dataset tersebut yang berperan sebagai data training akan pindah ke tabel training. Tabel training ini memiliki tiga atribut yaitu id_training, konten dan label. Tabel training dapat dilihat pada Tabel 3.16.

Tabel 3.16 Struktur Tabel Training

No	Nama Field	Tipe Data	Constraint	Keterangan
1	id_training	Int (11)	Primary Key, Auto Increment	Id Training
2	konten	Text	None	Konten training
3	label	Varchar (10)	None	Label konten training

6. Tabel Testing

Sama halnya dengan tabel training, tabel testing juga berisi data dari tabel dataset yang telah digunakan untuk pembuatan atau *upgrade* model prediksi. Data dari tabel dataset yang akan pindah ke tabel testing hanya data yang berperan sebagai data testing saat proses pembuatan atau *upgrade* model. Tabel testing ini memiliki tiga atribut yaitu id_testing, konten dan label. Tabel testing dapat dilihat pada Tabel 3.17.

Tabel 3.17 Struktur Tabel Testing

No	Nama Field	Tipe Data	Constraint	Keterangan
1	id_testing	Int (11)	<i>Primary Key, Auto Increment</i>	Id testing
2	konten	Text	<i>None</i>	Konten testing
3	label	Varchar (10)	<i>None</i>	Label konten testing

7. Tabel Pengujian

Tabel pengujian berfungsi untuk menyimpan data yang berhubungan dengan hasil dari proses pengujian model prediksi. Tabel pengujian ini terdiri dari delapan atribut yaitu id_pengujian, tanggal, total_dataset, data_latih, data_test, presisi, recall dan akurasi. Id_pengujian pada tabel ini sebagai *primary key*. Struktur dari tabel pengujian dapat dilihat pada Tabel 3.18.

Tabel 3.18 Struktur Tabel Pengujian

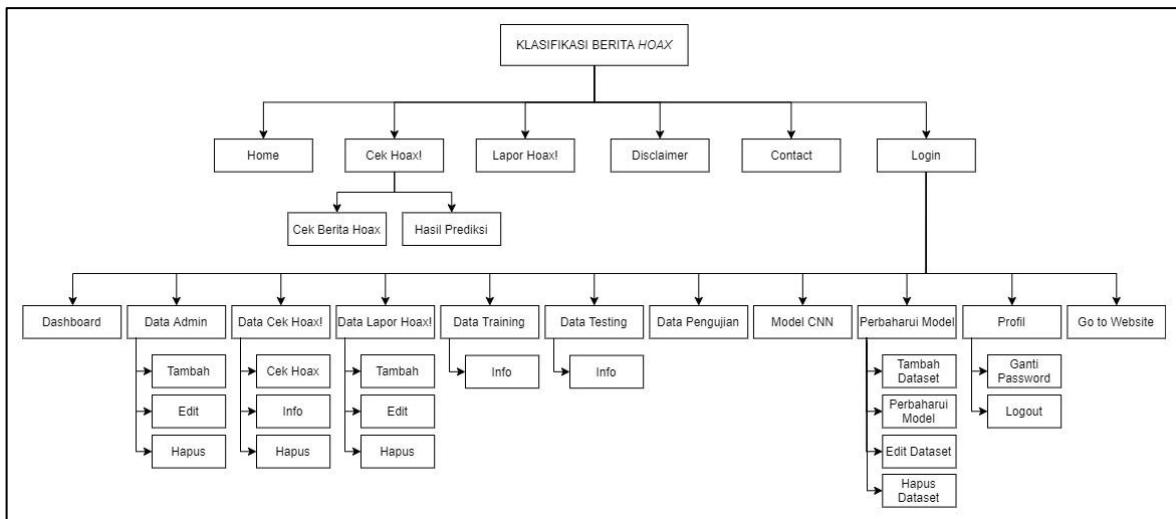
No	Nama Field	Tipe Data	Constraint	Keterangan
1	id_pengujian	Int (11)	<i>Primary Key, Auto Increment</i>	Id pengujian
2	tanggal	Date	<i>None</i>	Tanggal pengujian
3	total_dataset	Int (11)	<i>None</i>	Total dataset yang diuji
4	data_latih	Int (11)	<i>None</i>	Total data latih
5	data_test	Int (11)	<i>None</i>	Total data test
6	presisi	Float	<i>None</i>	Hasil presisi pengujian
7	recall	Float	<i>None</i>	Hasil recall pengujian
8	akurasi	Float	<i>None</i>	Hasil akurasi pengujian

3.2.2.3 Perancangan Antarmuka

Pada bagian ini akan memuat hal yang berkaitan dengan perancangan antarmuka sistem. Pada perancangan antarmuka sistem pada penelitian ini meliputi perancangan struktur menu dan *user interface*.

a. Perancangan Struktur Menu

Bagian ini akan menggambarkan menu-menu yang terdapat pada sistem Klasifikasi Berita *Hoax* beserta sub-menunya. Secara lebih rinci, struktur menu yang dihasilkan dapat dilihat pada Gambar 3.29.



Gambar 3.29 Struktur Menu

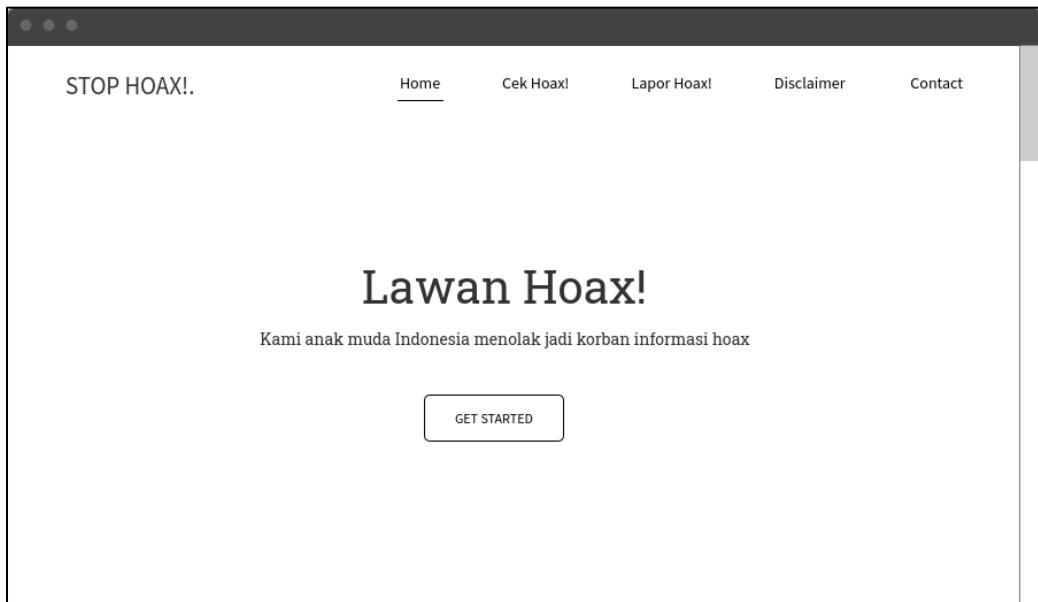
b. Perancangan User Interface

Perancangan *user interface* merupakan model mekanisme komunikasi antara pengguna dengan sistem. Antarmuka atau *user interface* dapat menerima informasi dan memberikan informasi pada pengguna dalam bentuk tampilan yang interaktif. Secara garis besar, *user interface* pada sistem ini akan dibagi menjadi dua aktor, yaitu *user* dan *admin*.

Berikut merupakan rancangan *user interface* pada aktor *user* yang terdiri dari beberapa bagian, diantaranya adalah sebagai berikut:

1. Rancangan Halaman *Home*

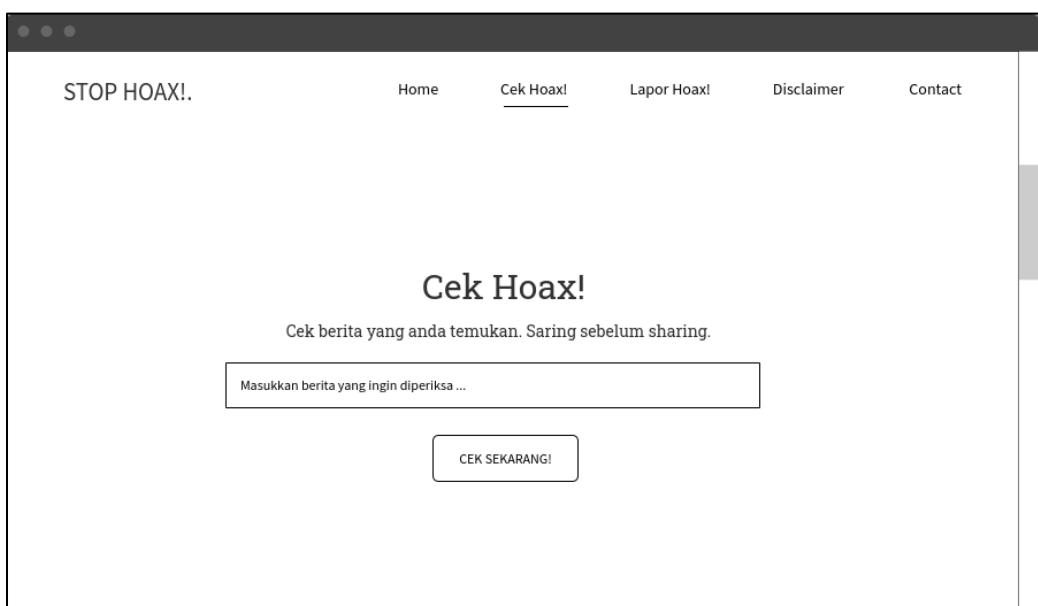
Halaman *home* merupakan halaman yang akan tampil pertama kali saat diakses oleh *user*. Pada halaman ini berisi *header* yang memuat judul dan slogan. Selain itu, terdapat *button* *Get Started* yang berfungsi untuk mengarahkan *user* ke halaman *Cek Hoax!*. Rancangan halaman *home* dapat dilihat pada Gambar 3.30.



Gambar 3.30 Rancangan Halaman *Home*

2. Rancangan Halaman Cek Hoax!

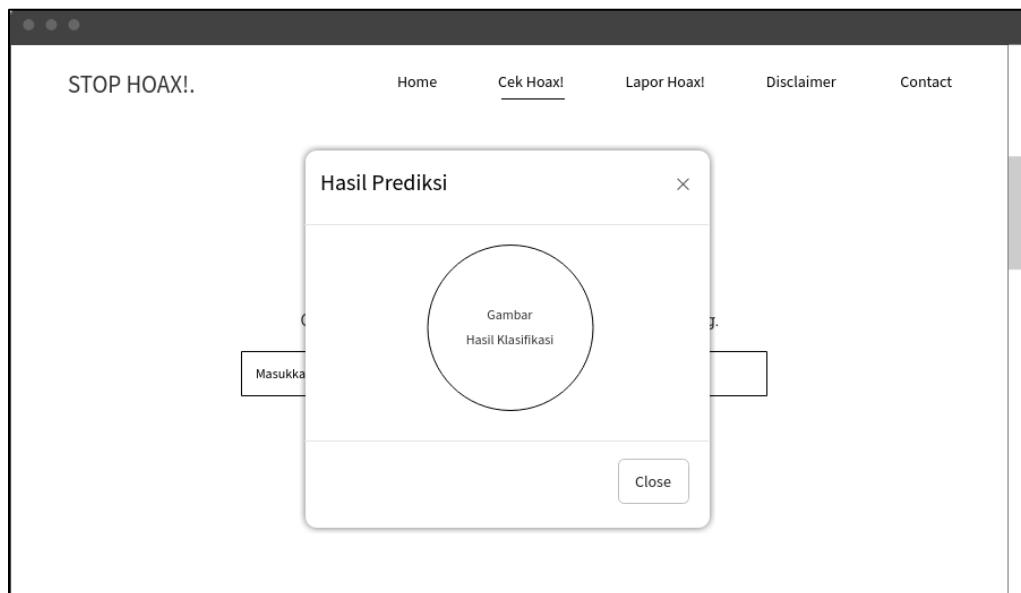
Halaman cek hoax merupakan halaman yang berisi *field* untuk melakukan pengecekan atau prediksi suatu konten berita, apakah berita tersebut *hoax* atau fakta. Pengecekan atau prediksi akan diproses setelah menekan *button* Cek Sekarang. Rancangan dari halaman cek hoax dapat dilihat pada Gambar 3.31.



Gambar 3.31 Rancangan Halaman Cek Hoax!

3. Rancangan Halaman Hasil Prediksi

Apabila menekan tombol Cek Sekarang! pada halaman cek hoax maka akan muncul halaman hasil prediksi. Halaman ini akan menampilkan gambar yang menunjukkan bahwa berita yang dicek termasuk berita *hoax* atau fakta. Rancangan halaman hasil prediksi seperti pada Gambar 3.32.



Gambar 3.32 Rancangan Halaman Hasil Prediksi

4. Rancangan Halaman Lapor Hoax!

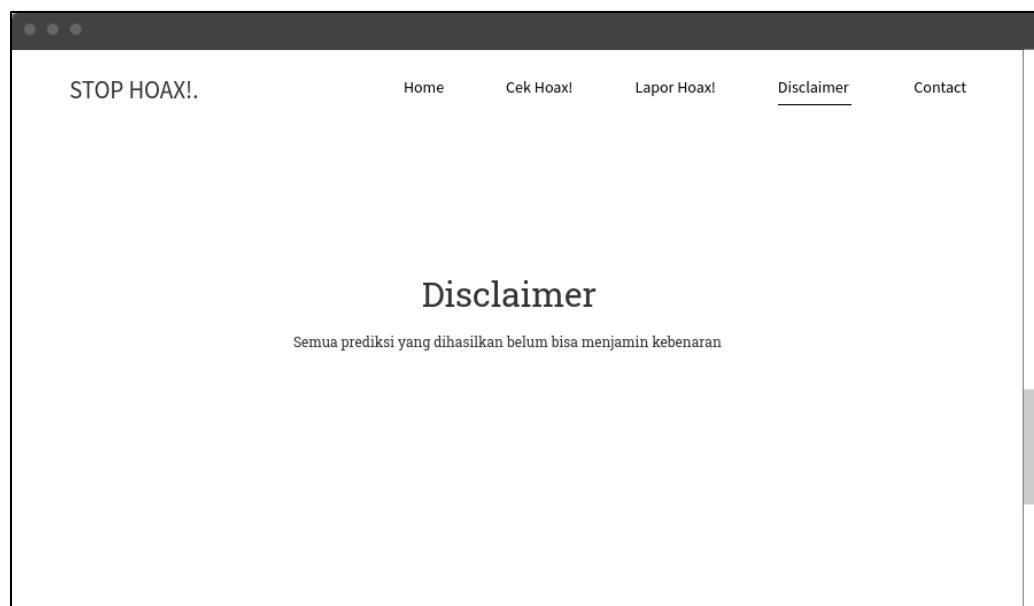
Halaman lapor hoax adalah halaman yang memuat fitur untuk melapor apabila menemukan berita *hoax* atau fakta di media sosial atau sebagainya dengan membuat sebuah form laporan. Pada form laporan ini, *user* mengisi data pribadi seperti nama depan, nama belakang dan email *user* serta konten berita yang ingin dilaporkan. *Laporan* ini akan dikirim ke *admin* untuk ditindaklanjuti. Rancangan dari halaman lapor hoax dapat dilihat pada Gambar 3.33.

The screenshot shows a web page titled "STOP HOAX!. Home Cek Hoax! Lapor Hoax! Disclaimer Contact". The main title is "Lapor Hoax!". Below it is a message: "Anda menemukan berita hoax ? **Lapor Sekarang!**". There are four input fields: "Nama Depan" and "Nama Belakang" in separate boxes, followed by a combined "Email" field, and a large text area labeled "Tulis konten berita ...". At the bottom is a button labeled "KIRIM LAPORAN".

Gambar 3.33 Rancangan Halaman Lapor Hoax!

5. Rancangan Halaman *Disclaimer*

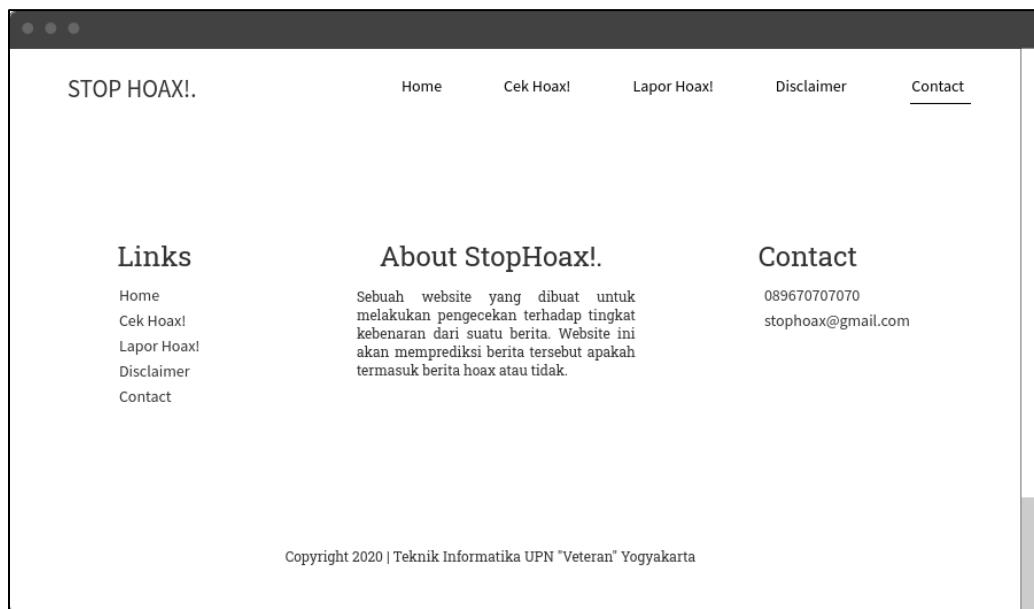
Halaman *disclaimer* ini berisi peringatan berupa batasan dan ruang lingkup bahwa sistem ini hanya menampilkan hasil prediksi dan tidak menjamin kebenaran prediksi tersebut. Rancangan dari halaman *disclaimer* dapat dilihat pada Gambar 3.34.



Gambar 3.34 Rancangan Halaman *Disclaimer*

6. Rancangan Halaman *Contact*

Halaman *contact* adalah halaman yang berisi informasi tentang sistem ini seperti link menuju ke menu-menu *user*, penjelasan singkat tentang sistem yang dibuat dan kontak yang dapat dihubungi. Rancangan halaman *contact* dapat dilihat pada Gambar 3.35.

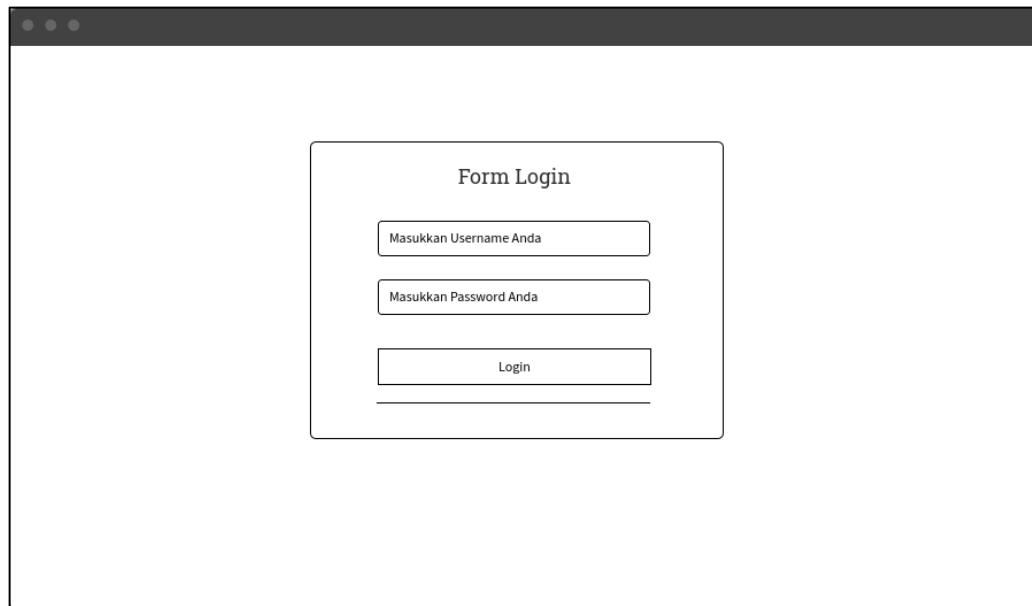


Gambar 3.35 Rancangan Halaman *Contact*

Setelah *user interface* untuk aktor *user*, berikut akan ditampilkan *user interface* pada aktor *admin* yang terdiri dari beberapa bagian, diantaranya yaitu:

1. Rancangan Halaman *Login*

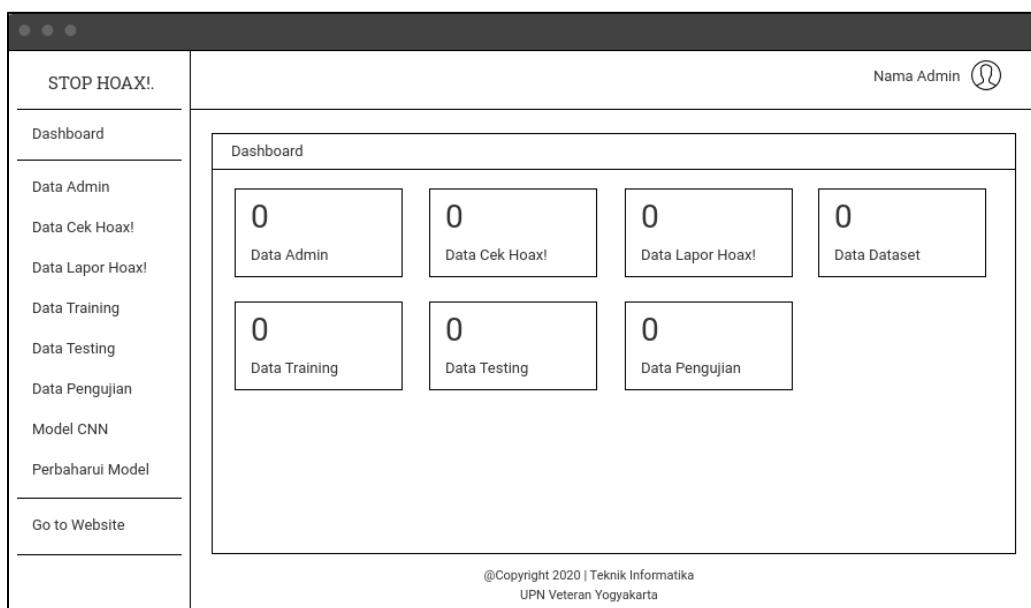
Halaman *login* adalah halaman yang akan tampil pertama kali apabila berperan sebagai aktor *admin* jika ingin masuk ke dalam sistem. Admin mengisi *username* dan *password* yang benar pada *form login*. Rancangan dari halaman *login* dapat dilihat pada Gambar 3.36.



Gambar 3.36 Rancangan Halaman *Login*

2. Rancangan Halaman *Dashboard*

Setelah *admin* berhasil melakukan *login*, maka akan tampil halaman *dashboard*. Halaman *dashboard* berisi jumlah data dari masing-masing tabel pada sistem ini yaitu jumlah data admin, jumlah data cek hoax, lapor hoax, dataset, data training dan data testing serta jumlah data pengujian. Berikut rancangan dari halaman *dashboard* pada Gambar 3.37.



Gambar 3.37 Rancangan Halaman *Dashboard*

3. Rancangan Halaman Data Admin

Halaman data admin akan menampilkan data-data admin yang divisualisasikan dalam bentuk tabel. Ada beberapa fitur yang tersedia pada halaman ini seperti fitur pencarian dan pengurutan data, navigator tabel dan jumlah baris tabel yang ingin ditampilkan pada setiap halaman tabel. Selain itu, pada halaman ini juga memiliki beberapa fitur untuk mengolah data seperti menambah data admin, mengedit data admin serta untuk menghapus data admin.

Rancangan untuk halaman data admin dapat dilihat pada Gambar 3.38.

No	Username	Password	Nama Admin	Aksi
1	
2	
3	
4	

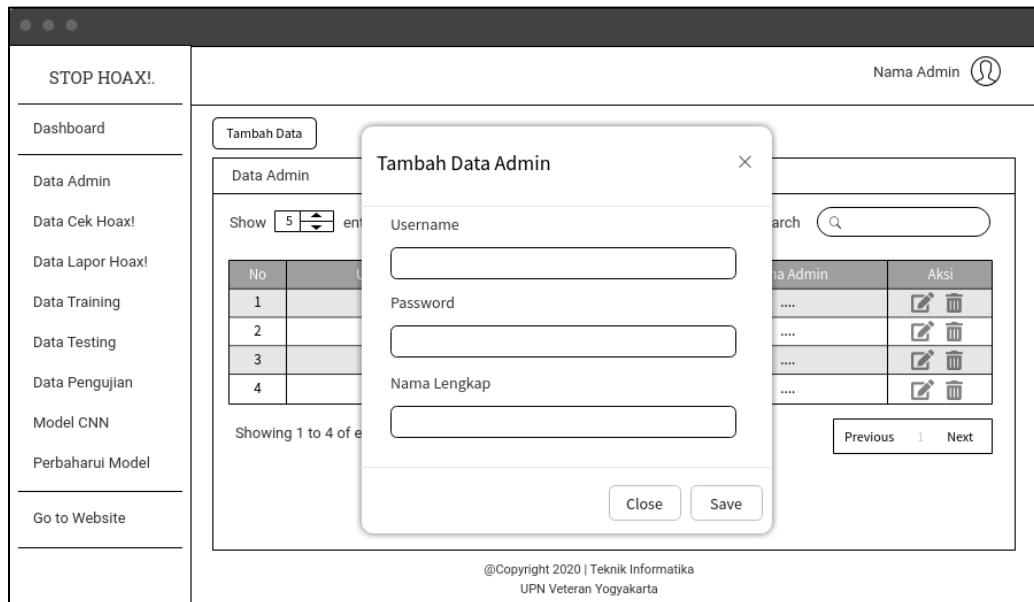
Showing 1 to 4 of entries

Previous 1 Next

@Copyright 2020 | Teknik Informatika
UPN Veteran Yogyakarta

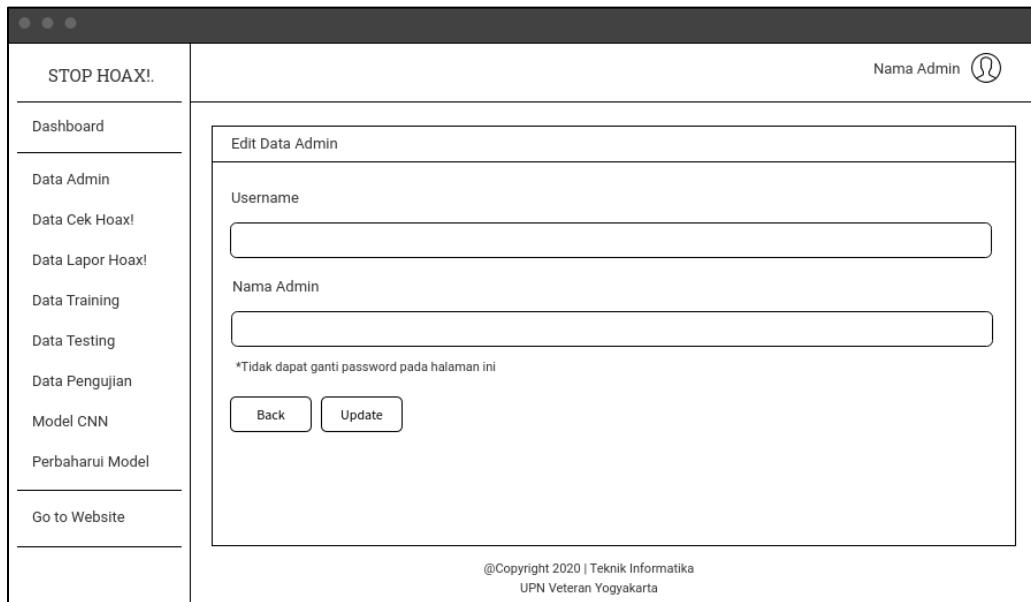
Gambar 3.38 Rancangan Halaman Data Admin

Apabila *admin* menekan tombol tambah data pada halaman data admin maka akan muncul *pop up* yang berisi *form* untuk menambah data admin. Isi dari *form* tersebut terdiri dari *username*, *password* dan nama lengkap. Jika admin ingin menyimpannya, tekan tombol *save*. Detail dari rancangan halaman tambah data admin ini seperti pada Gambar 3.39.



Gambar 3.39 Rancangan Halaman Tambah Data Admin

Fungsi lain yang ada di halaman data admin yaitu edit data. Tombol edit data direpresentasikan dengan gambar pensil pada kolom aksi di tabel. Pada halaman edit data admin, admin merubah sesuai dengan baris data yang dipilih. Data admin yang dapat diubah hanya *username* dan nama lengkap admin, sedangkan untuk *password* hanya dapat diubah oleh admin yang bersangkutan pada menu profil. Apabila sudah mengganti data admin yang dipilih dengan data admin yang baru pada form yang tersedia, selanjutnya tekan tombol *update* untuk menyimpannya. Rancangan halaman pada edit data admin seperti pada Gambar 3.40.

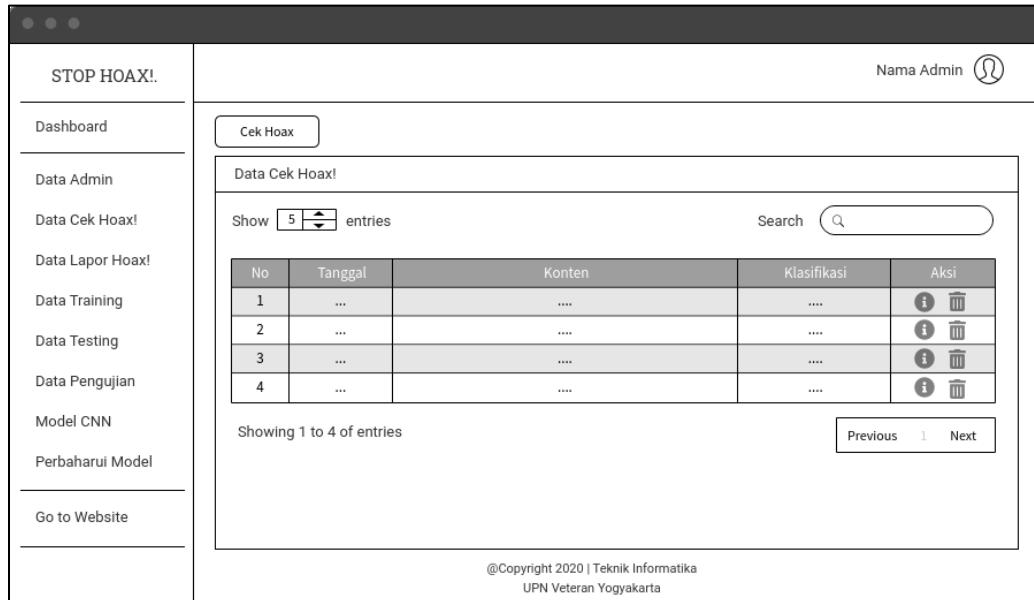


Gambar 3.40 Rancangan Halaman Edit Data Admin

Lalu, tombol yang berfungsi untuk menghapus data pada halaman data admin pada gambar tempat sampah di kolom aksi. Tekan *button* tersebut berdasarkan baris yang dipilih, maka data adminnya akan terhapus.

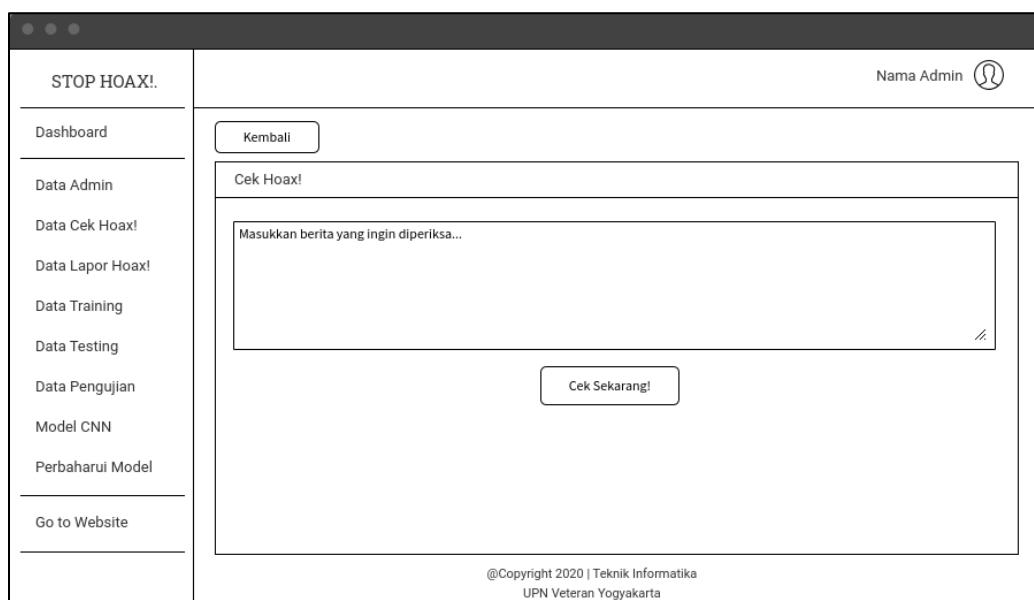
4. Rancangan Halaman Data Cek Hoax!

Halaman data cek hoax adalah halaman pada admin yang akan menampilkan data yang ada pada tabel cek hoax dari database. Data cek hoax yang ditampilkan yaitu no, tanggal, konten, klasifikasi dan kolom aksi untuk proses olah data. Halaman ini juga sama seperti halaman data admin yang mempunyai fitur untuk pencarian dan pengurutan data, fitur cek hoax, lihat data dan hapus data serta navigator tabel. Halaman ini tidak ada proses edit data bertujuan meminimalisir adanya manipulasi data hasil prediksi. Sehingga hasil yang ditampilkan merupakan hasil *real* dari model yang telah dibuat. Berikut rancangan dari halaman data cek hoax yang akan dibuat seperti pada Gambar 3.41.



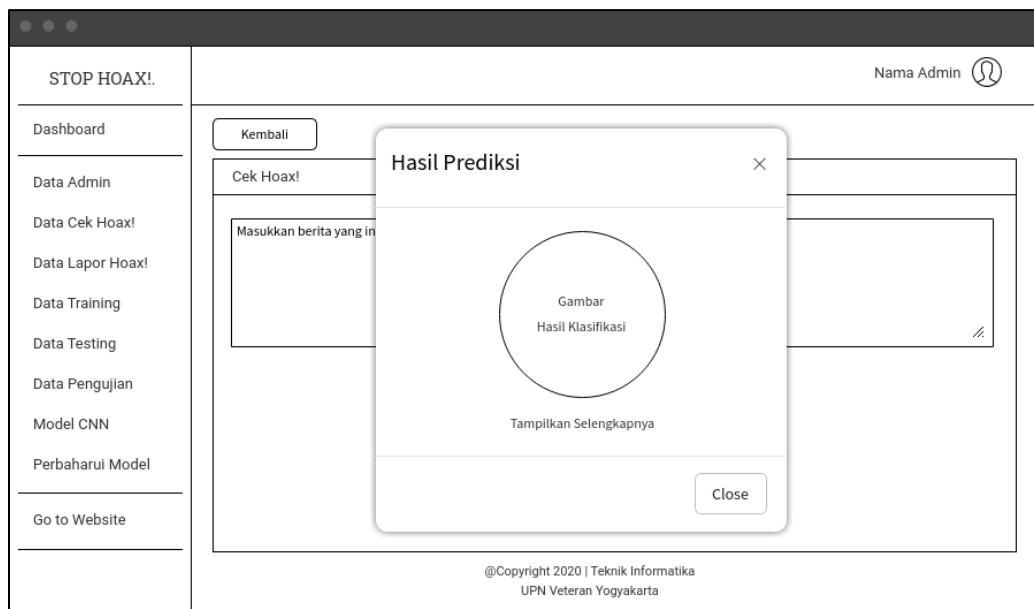
Gambar 3.41 Rancangan Halaman Data Cek Hoax

Selanjutnya yaitu fungsi pada tombol cek hoax pada halaman ini. Tombol tersebut untuk pindah ke halaman cek hoax. Halaman cek hoax berfungsi untuk melakukan proses pengecekan berita hoax seperti pada proses pengecekan hoax di halaman *user*. Dimana admin memasukkan konten berita yang ingin dicek lalu menekan *button* Cek Sekarang untuk melakukan proses pengecekan. Halaman cek hoax pada bagian admin ini akan dibuat seperti rancangan pada Gambar 3.42.



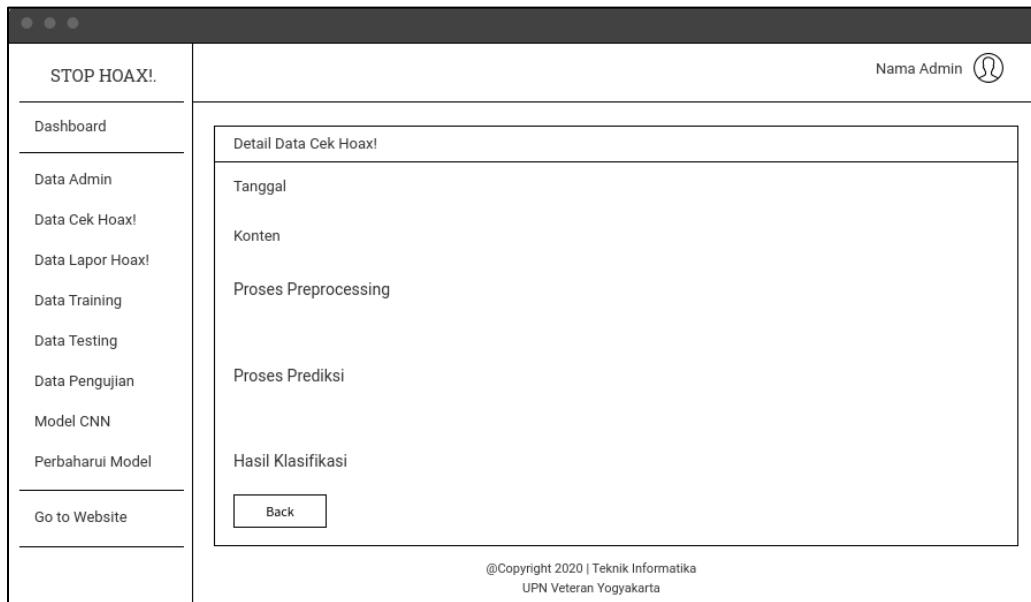
Gambar 3.42 Rancangan Halaman Cek Hoax Admin

Setelah menekan tombol Cek Sekarang, sistem akan melakukan proses klasifikasi dari konten yang dicek. Lalu, akan muncul *pop up* yang akan menampilkan hasil prediksinya dalam bentuk gambar *hoax* atau fakta. Rancangan *pop up* dari hasil prediksi tersebut dapat dilihat pada Gambar 3.43.



Gambar 3.43 Rancangan Halaman Hasil Prediksi Admin

Beda dari halaman cek hoax di *user*, pada halaman ini terdapat tombol *tampilkan selengkapnya* yang berfungsi untuk menampilkan hasil dari setiap tahapan proses klasifikasi secara detail. Halaman detail cek hoax tersebut akan seperti pada rancangan di Gambar 3.44. Halaman tersebut berisi tanggal pengecekan, konten, proses *preprocessing*, proses prediksi dan hasil klasifikasinya.

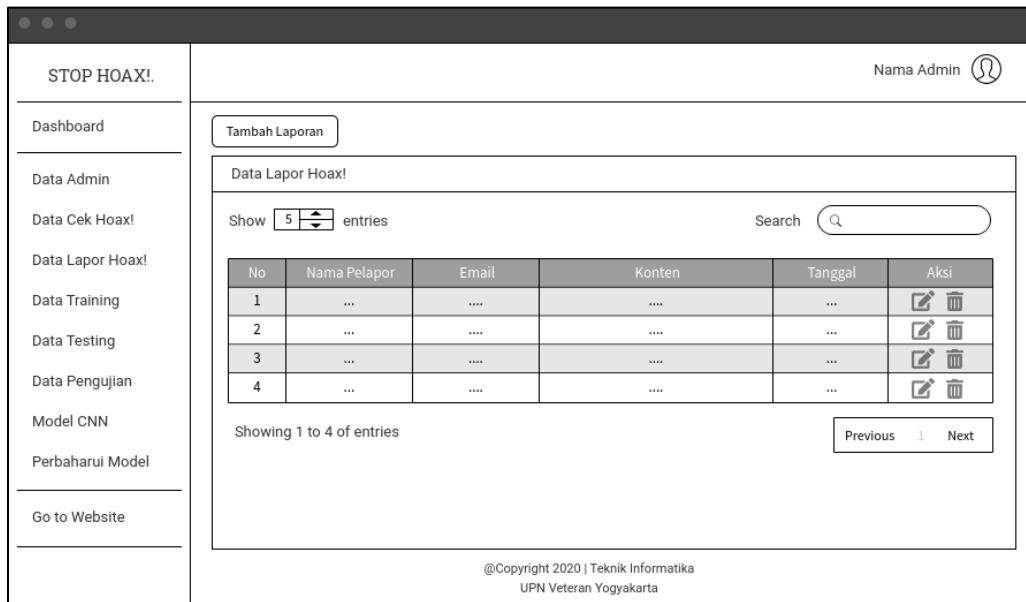


Gambar 3.44 Rancangan Halaman Detail Cek Hoax

Halaman detail cek hoax tersebut juga akan muncul saat menekan tombol yang bergambar i yang ada di kolom aksi pada halaman data cek hoax. Fitur olah data lain yang dapat dilakukan pada halaman data cek hoax yaitu hapus data cek hoax. Cara untuk menghapusnya dengan memilih baris yang dihapus dan menekan tombol yang bergambar tempat sampah di kolom aksi.

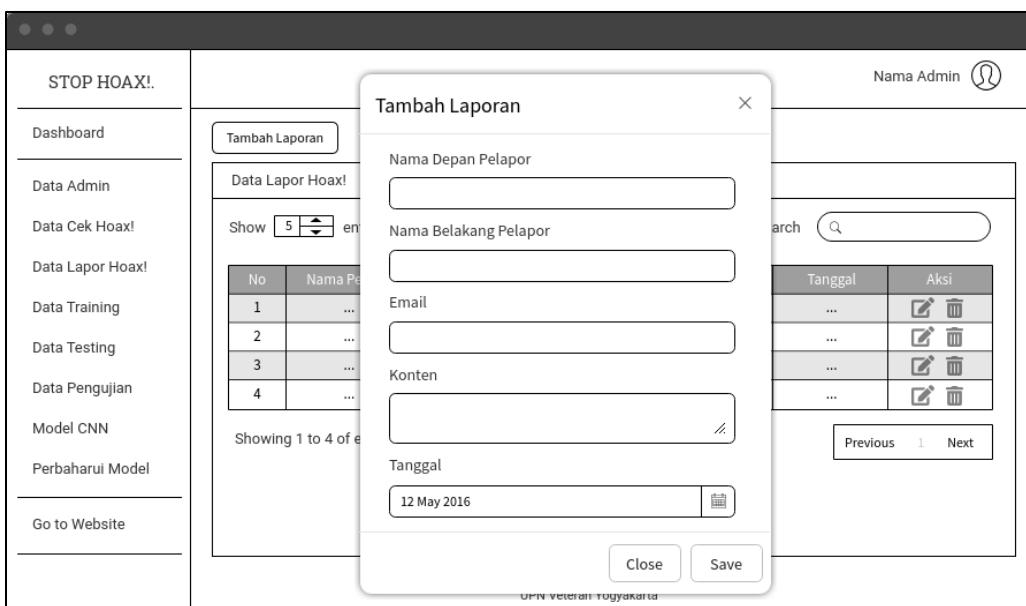
5. Rancangan Halaman Data Lapor Hoax!

Halaman data lapor hoax merupakan halaman yang berisi data-data laporan yang dikirim oleh *user* ataupun yang diinput oleh admin. Data yang ditampilkan yaitu nama pelapor, email, konten yang dilaporkan dan tanggal saat melakukan laporan. Admin dapat mengolah data seperti menambah data laporan, edit laporan dan hapus laporan. Rancangan dari halaman data lapor hoax dapat dilihat pada Gambar 3.45.



Gambar 3.45 Rancangan Halaman Data Lapor Hoax

Admin dapat menambah data lapor hoax dengan menekan tombol tambah laporan, kemudian mengisi *form* yang terdiri dari nama depan, nama belakang, email, konten dan tanggal pelaporan. Jika sudah, tekan *save* untuk menyimpan data tersebut. Rancangan dari Halaman tambah data lapor hoax dapat dilihat pada Gambar 3.46.



Gambar 3.46 Rancangan Halaman Tambah Data Lapor Hoax

Selain menambah data, admin juga dapat mengubah data lapor hoax. Untuk mengubah data dengan menekan tombol yang bergambar pensil. Pada halaman edit data

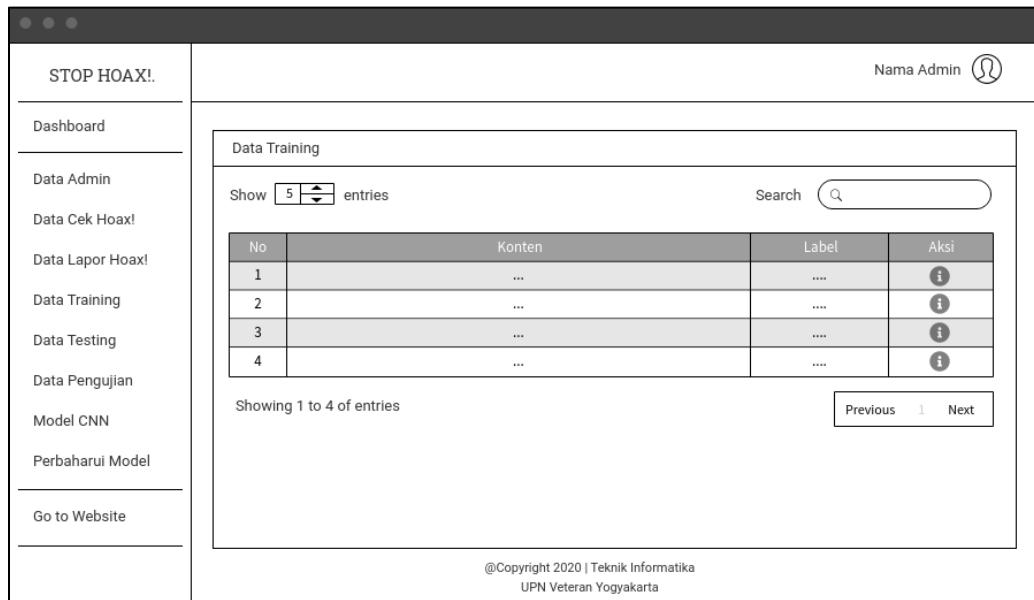
lapor hoax, akan muncul form yang telah terisi sesuai data yang dipilih. Kemudian admin dapat mengubah dan menyimpannya dengan menekan tombol *update*. Rancangan dari halaman edit data lapor hoax dapat dilihat pada Gambar 3.47.

Rancangan halaman edit data lapor hoax yang ditampilkan dalam tampilan desktop. Di bagian kiri ada sidebar dengan menu: STOP HOAX!, Dashboard, Data Admin, Data Cek Hoax!, Data Lapor Hoax!, Data Training, Data Testing, Data Pengujian, Model CNN, Perbaharui Model, dan Go to Website. Di bagian kanan ada judul "Edit Data Lapor Hoax!", form input untuk Nama Depan, Nama Belakang, Email, Konten (dengan area scroll), Tanggal (dengan ikon kalender), dan tombol Back dan Update. Di bagian bawah ada krediter: @Copyright 2020 | Teknik Informatika UPN Veteran Yogyakarta.

Gambar 3.47 Rancangan Halaman Edit Data Lapor Hoax

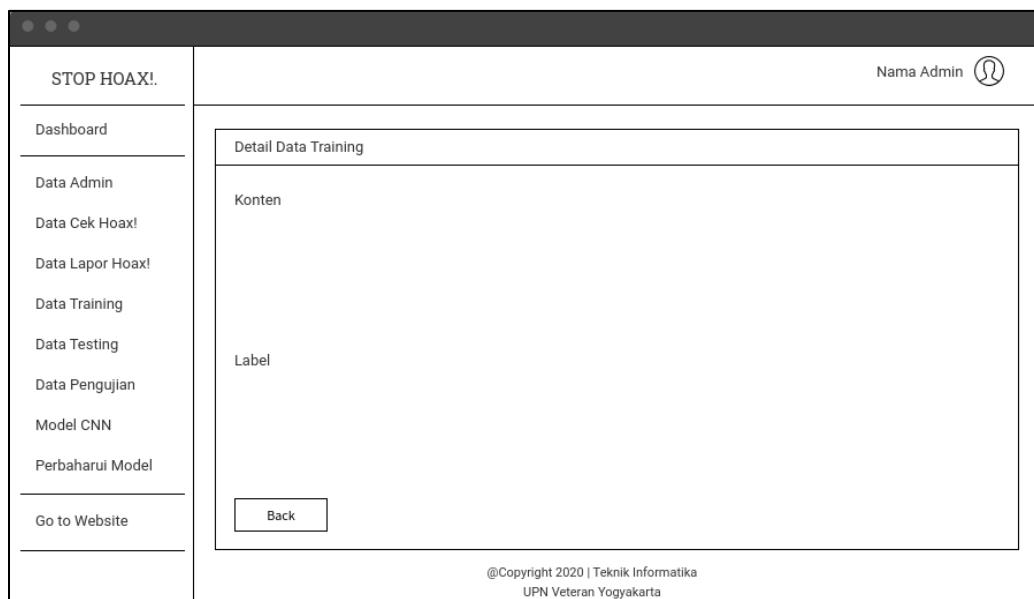
6. Rancangan Halaman Data Training

Halaman data training ini akan menampilkan data training yang berisi konten, label dan aksi untuk melihat data. Pada halaman data training ini tidak ada proses olah data seperti tambah dan hapus, hanya ada tombol bergambar i untuk melihat data secara lengkap. Rancangan dari halaman data training seperti pada Gambar 3.48.



Gambar 3.48 Rancangan Halaman Data Training

Ketika admin menekan *button i* pada halaman data training, akan muncul halaman detail data training yang menampilkan konten secara lengkap beserta labelnya. Berikut rancangan dari detail data training pada Gambar 3.49.



Gambar 3.49 Rancangan Halaman Detail Data Training

7. Rancangan Halaman Data Testing

Halaman data testing tidak jauh berbeda dengan data training, hanya berbeda di bagian data yang ditampilkan. Halaman data testing ini akan menampilkan tabel training yang berisi

data konten dan label serta ada kolom aksi untuk ke halaman detail data testing. Rancangan dari halaman detail data testing dapat dilihat pada Gambar 3.50.

Rancangan halaman Data Testing yang menampilkan daftar data dengan tiga kolom: Konten, Label, dan Aksi. Setiap baris memiliki tombol informasi (i) di kolom Aksi. Di bawah tabel terdapat tombol 'Previous' dan 'Next'.

No	Konten	Label	Aksi
1	i
2	i
3	i
4	i

Showing 1 to 4 of entries

Previous 1 Next

@Copyright 2020 | Teknik Informatika
UPN Veteran Yogyakarta

Gambar 3.50 Rancangan Halaman Data Testing

Ketika admin menekan *button i* pada kolom aksi tersebut maka akan ke halaman detail data testing untuk menampilkan data konten secara lengkap beserta labelnya sesuai dengan baris yang dipilih. Berikut rancangan dari halaman detail data testing pada Gambar 3.51.

Rancangan halaman Detail Data Testing yang menampilkan dua bagian utama: Konten dan Label. Di bawahnya terdapat tombol 'Back'.

Konten

Label

Back

@Copyright 2020 | Teknik Informatika
UPN Veteran Yogyakarta

Gambar 3.51 Rancangan Halaman Detail Data Testing

8. Rancangan Halaman Data Pengujian

Halaman data pengujian akan berisi data yang memuat tanggal, total dataset, total data traning dan data testing, presisi, recall dan akurasi yang berasal dari tabel pengujian di database. Pada halaman ini juga memiliki fitur yang biasa ada di tabel seperti pencarian dan pengurutan data, navigator tabel serta jumlah data yang ditampilkan per halaman. Rancangan dari halaman data pengujian seperti pada Gambar 3.52.

No	Tanggal	Total Dataset	Data Training	Data Testing	Presisi	Recall	Akurasi
1
2
3
4

Showing 1 to 4 of entries

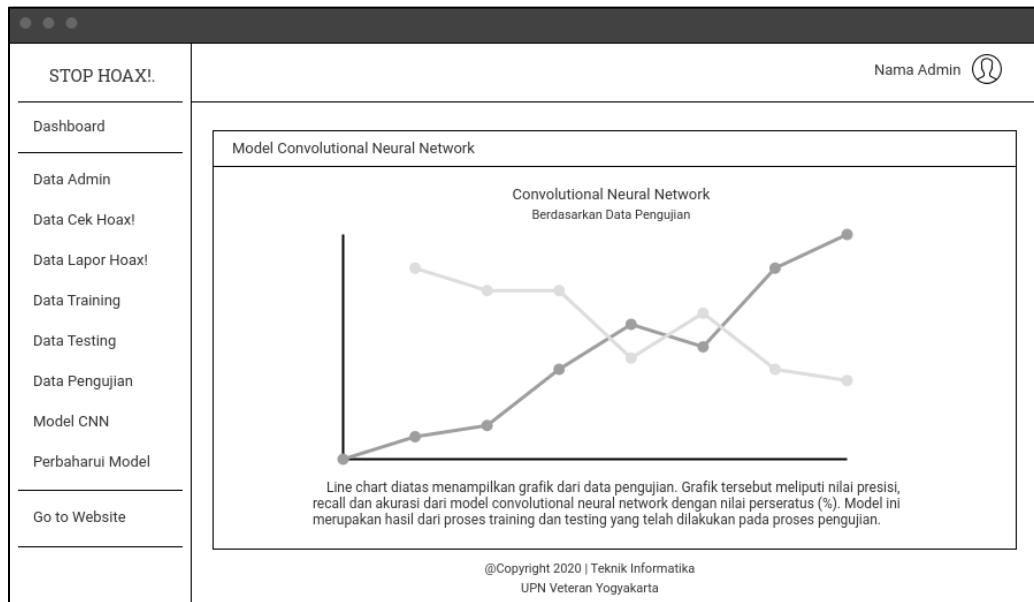
Previous 1 Next

@Copyright 2020 | Teknik Informatika
UPN Veteran Yogyakarta

Gambar 3.52 Rancangan Halaman Data Pengujian

9. Rancangan Halaman Model CNN

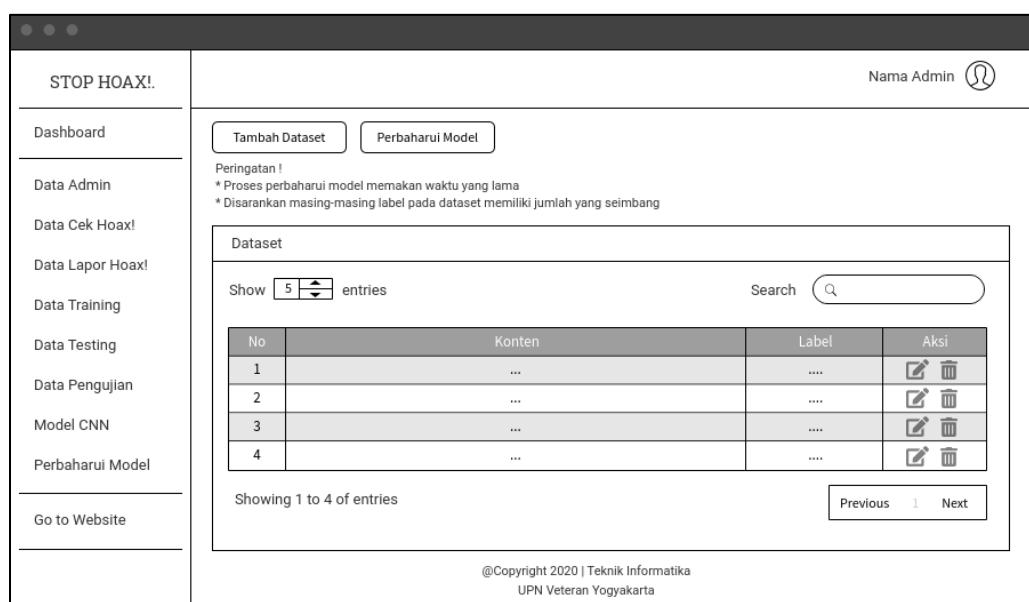
Halaman model CNN yaitu halaman yang akan menampilkan grafik dari prosentase nilai presisi, recall dan akurasi berdasarkan data pengujian. Data tersebut akan ditampilkan menggunakan *line chart*. Rancangan dari halaman model CNN akan dibuat seperti pada Gambar 3.53.



Gambar 3.53 Rancangan Halaman Model CNN

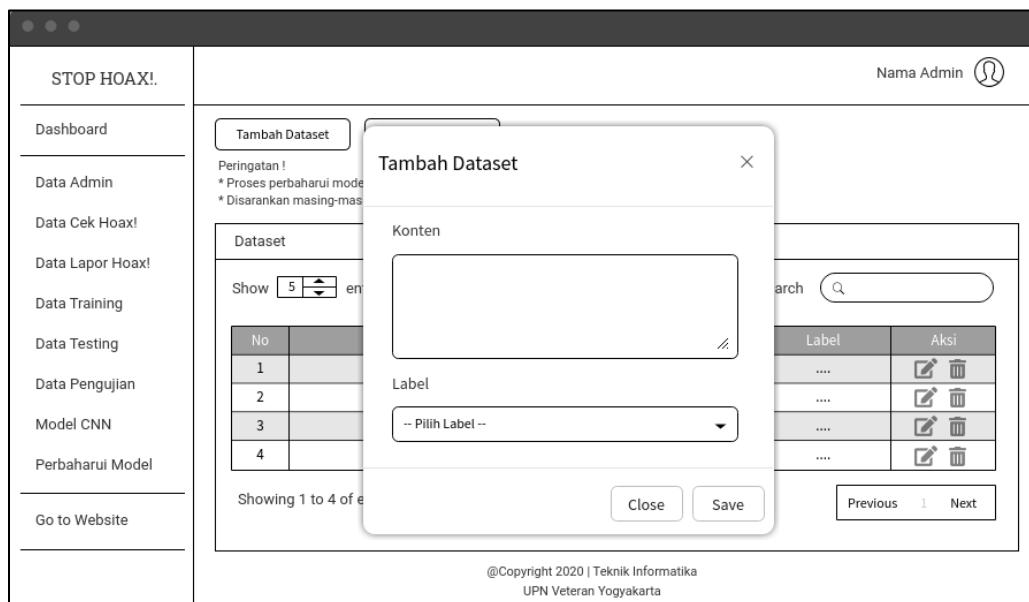
10. Rancangan Halaman Perbaharui Model

Halaman perbaharui model merupakan halaman untuk *update* model dengan data baru dengan melalui proses *training* dan *testing* menggunakan model yang sudah dibuat. Dengan adanya perbaharui model ini diharapkan dapat meningkatkan akurasi dari model sehingga hasil prediksinya menjadi lebih akurat. Halaman ini berisi dataset yang akan digunakan dan fitur olah data seperti tambah, edit dan hapus dataset. Rancangan dari halaman perbaharui model seperti pada Gambar 3.54.



Gambar 3.54 Rancangan Halaman Perbaharui Model

Jika menekan tombol tambah dataset, maka akan muncul *pop up* yang berisi form yang terdiri dari konten dan label. Untuk menyimpannya dengan menekan tombol *save*. Rancangan dari halaman tambah dataset akan seperti pada Gambar 3.55.



Gambar 3.55 Rancangan Halaman Tambah Dataset

Dari data yang telah disimpan, admin dapat melakukan edit dataset dengan mengklik *button* yang bergambar pensil di kolom aksi. Maka data yang telah di pilih akan muncul di halaman edit dataset. Terdapat tombol *update* untuk merubah data tersebut. Rancangan dari halaman edit dataset dapat dilihat pada Gambar 3.56.

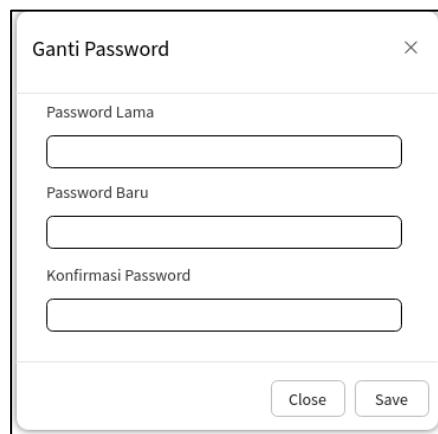


Gambar 3.56 Rancangan Halaman Edit Dataset

Selanjutnya fitur hapus dataset dengan menekan tombol yang bergambar tempat sampah di kolom aksi. Maka data yang dipilih akan terhapus.

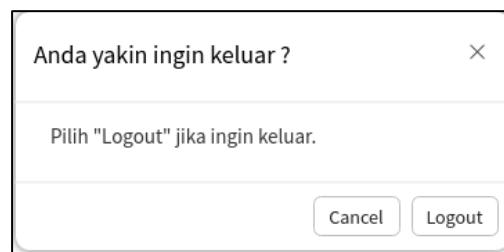
11. Rancangan Halaman Profil

Bagian profil admin terdapat pada bagian sebelah kanan atas di halaman *admin* dengan menekan *icon* akun. Maka akan muncul dua pilihan yaitu ganti password dan *logout*. Untuk mengganti password admin dengan mengisi form yang terdiri dari password lama, password baru dan konfirmasi password. Rancangan dari halaman ganti password akan dibuat seperti pada Gambar 3.57.



Gambar 3.57 Rancangan Halaman Ganti Password

Selain itu, pada menu profil terdapat sub menu *logout*. Menu tersebut berfungsi untuk keluar dari sistem. Rancangan dari tampilan *logout* seperti pada Gambar 3.58.



Gambar 3.58 Rancangan Halaman *Logout*

3.3 Perancangan Pengujian

Pada bagian ini akan membahas tentang perancangan pengujian. Penelitian ini akan melakukan dua pengujian yaitu pengujian terhadap model yang telah dibuat dan pengujian terhadap sistem.

3.3.1 Pengujian Model

Pengujian model merupakan proses validasi dan pengujian dari dataset pada algoritma CNN. Tahap ini bertujuan untuk mengukur kinerja dari suatu sistem yang telah dibangun. Proses ini biasa disebut juga dengan *training* dan *testing*. Bagian *training* yaitu untuk melatih sistem dengan dataset yang telah diberi label sebagai bahan belajar dan akan menghasilkan suatu model. Model yang dihasilkan ini akan digunakan untuk melakukan klasifikasi pada tahap *testing* dengan memberi data pada model tanpa adanya label. Pada *testing* ini akan mengukur seberapa baik model yang telah dihasilkan. Metode pengujian yang akan digunakan pada penelitian ini yaitu *k-fold cross validation*, *confusion matrix* dan kurva ROC.

Metode *k-fold cross validation* bertujuan untuk melakukan validasi dari suatu model dengan melakukan pengujian sebanyak K, dimana pada penelitian ini menggunakan K=5. Sedangkan, *confusion matrix* untuk mengukur tingkat akurasi, presisi dan recall dari suatu model berdasarkan prediksi yang dihasilkan oleh sistem. Parameter perhitungan *confusion matrix* terdiri 4 hal yaitu *True Positive* (TP), *True Negative* (TN), *False Positive* (FP), *False Negative* (FN). Selain itu, pengujian juga dilakukan menggunakan kurva ROC dengan mendapatkan nilai FPR dan TPR dari tabel *confusion matrix*. Berikut rancangan pengujian menggunakan *k-fold cross validation*, *confusion matrix* dan kurva ROC dapat dilihat pada Tabel 3.19, 3.20, 3.21, 3.22.

Tabel 3.19 Tabel *Confusion Matrix*

Fold	<i>Confusion Matrix</i>			
	TP	FN	FP	TN
1				
2				
3				
4				
5				
Rata-Rata				

Tabel 3.20 Tabel Pengujian *K-Fold Cross Validation*

Fold	Akurasi	Presisi	Recall
1			
2			
3			
4			
5			
Rata-Rata			

Tabel 3.21 Tabel FPR dan TPR Kurva ROC

Fold	FPR	TPR
1		
2		
3		
4		
5		
Rata-Rata		

Tabel 3.22 Tabel AUC Kurva ROC

Fold	AUC
1	
2	
3	
4	
5	
Rata-Rata	

3.3.2 Pengujian Sistem

Pada bagian ini akan dibuat suatu rancangan pengujian sistem. Pembuatan dari rancangan pengujian bertujuan untuk mempersiapkan konsep pengujian yang akan dilakukan untuk mengukur kinerja dari suatu sistem yang telah dibangun. Metode pengujian yang akan digunakan pada penelitian ini yaitu *black box testing*.

Pengujian menggunakan *black box* bertujuan untuk mengukur kinerja dari suatu perangkat lunak yang telah dibangun. Metode pengujian secara *black box* lebih menekankan pada bagian fungsionalitas dari suatu perangkat lunak sehingga dapat memastikan setiap fungsi yang ada berjalan dengan semestinya. Berikut detail rancangan pengujian menggunakan *black box* dapat dilihat pada Tabel 3.23.

Tabel 3.23 Rancangan Pengujian Black Box

Aktor	Halaman	Detail Pengujian	Pengujian	
			Berhasil	Gagal
User	Home	Menampilkan daftar menu		
		Menampilkan header		
		Fungsi tombol <i>Get Started</i>		
	Cek Hoax	Menampilkan daftar menu		
		Fungsi cek hoax		
		Menampilkan hasil klasifikasi		
	Lapor Hoax	Menampilkan daftar menu		
		Pengajuan laporan hoax		
	Disclaimer	Menampilkan daftar menu		
		Menampilkan disclaimer sistem		
Admin	Contact	Menampilkan daftar menu		
		Menampilkan informasi sistem		
		Berpindah ke menu lain via links		
	Login	Melakukan proses login		
		Fungsi session		
	Dashboard	Menampilkan jumlah setiap data		
	Data Admin	Menampilkan data admin		
		Menambahkan data admin		
		Mengedit data admin		
		Menghapus data admin		
		Fungsi pencarian dan pengurutan		
	Data Cek Hoax	Menampilkan data cek hoax		
		Melakukan pengecekan hoax		
		Menampilkan hasil klasifikasi		
		Menampilkan detail cek hoax		
		Menghapus data cek hoax		
		Fungsi pencarian dan pengurutan		
	Data Lapor Hoax	Menampilkan data laporan hoax		
		Menambahkan data laporan hoax		
		Mengedit data laporan hoax		
		Menghapus data laporan hoax		
		Fungsi pencarian dan pengurutan		
	Data Training	Menampilkan data training		
		Menampilkan detail data training		
		Fungsi pencarian dan pengurutan		
	Data Testing	Menampilkan data testing		
		Menampilkan detail data testing		
		Fungsi pencarian dan pengurutan		
	Data Pengujian	Menampilkan data pengujian		
	Fungsi pencarian dan pengurutan			
	Model CNN	Menampilkan grafik hasil pengujian		
		Mengolah data grafik		
	Pembaharui Model	Menampilkan data dataset		
		Menambahkan data dataset		
		Mengedit data dataset		
		Menghapus data dataset		
		Melakukan proses pembaharui model		
		Fungsi pencarian dan pengurutan		
	Profil	Fungsi ganti password		
	Fungsi logout			
	Go to Website	Berpindah ke halaman user		

BAB IV

HASIL, PENGUJIAN DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai implementasi dari hasil rancangan yang telah dibuat sebelumnya. Implementasi ini termasuk ke dalam tahap membangun *prototype* yang meliputi pembuatan aplikasi hingga siap dioperasikan pada keadaan sebenarnya. Setelah itu, akan dilakukan pengujian terhadap sistem yang telah dibuat untuk mengetahui kinerjanya baik dari sisi jalannya sistem maupun performa dari algoritma yang digunakan. Metode pengujian pada penelitian ini terdiri dari *black box testing*, *k-fold cross validation* dan *confusion matrix*. Tahap pengujian ini diharapkan dapat mengetahui kekurangan-kekurangan pada penelitian ini sehingga dapat dilakukan pengembangan selanjutnya.

4.1 Hasil Penelitian

Pada bagian ini akan membahas mengenai hasil implementasi dari rancangan yang telah dibuat sebelumnya. Hasil penelitian ini akan dibagi menjadi dua bagian yaitu implementasi klasifikasi berita *hoax* dan implementasi aplikasi.

4.1.1 Implementasi Klasifikasi Berita Hoax

Implementasi klasifikasi berita *hoax* yaitu hasil penelitian yang berkaitan dengan metode yang digunakan sesuai dengan tahapan dari klasifikasi berita *hoax*. Berdasarkan metodologi penelitian yang telah dibuat, proses klasifikasi berita *hoax*.

4.1.1.1 Modul Pengumpulan Data

Pengumpulan data menggunakan *web scraping* dari situs turnbackhoax.id. Proses pengumpulan data ini memanfaatkan *extension google* yang bernama *data miner*. Setelah melakukan instalasi, maka dapat melakukan *scraping* dengan membuka situs turnbackhoax.id dan masuk ke salah satu konten berita. Lalu, buka *data miner* dan pilih *list page*. Selanjutnya pilih *element selector* untuk bagian *rows*, *cols* dan *nav*. Berikut modul *element selector* pada Modul 4.1.

```
.mh-wrapper #Rows
.entry-content #Cols
.mh-post-nav-prev a #Nav
```

Modul Program 4.1 Pseudocode Element Selector Data Miner

4.1.1.2 Modul Labeling

Setelah mendapatkan data, selanjutnya diberikan label di masing-masing konten berita. Setiap data nantinya akan diberi label yaitu fakta atau *hoax*. Sebelum itu, data yang didapat dari hasil *web scraping* di *import* terlebih dahulu menggunakan *library csv* karena datanya berformat csv. Lalu, data tersebut dibuka menggunakan fungsi `open()` dengan parameter ‘r’ dan dibaca menggunakan `csv.reader()`. Setelah itu, menyiapkan file untuk hasil dari proses labeling dengan nama file `data_akhir2.csv` pada fungsi `open()` dengan parameter ‘w’ untuk menyimpan. Untuk menulis data pada file hasil labeling menggunakan `writer.writerows()`, misalnya data untuk nama kolom yaitu konten dan kelas. Berikut modul *import* data proses labeling pada Modul 4.2.

```
Algoritma Labeling

Deklarasi:
INPUT : open, csv.reader, writerows (Library csv)
OUTPUT : dataa_akhir2.csv

Deskripsi:
filename <- 'data_awal.csv'
f <- open(filename,'r', encoding='utf8')
reader <- csv.reader(f)
total_data <- 0
total_penjelasan <- 0
total_narasi <- 0
w <- open('data_akhir2.csv', 'w', encoding='utf8', newline='')
writer <- csv.writer(w)
head <- [['Konten','Kelas']]
writer.writerows(head)
```

Modul Program 4.2 Pseudocode Import Data Proses Labeling

Setelah *import* data, maka akan dilakukan proses labeling. Terdapat beberapa aturan untuk menentukan label atau kelas pada data dari turnbackhoax.id yaitu melihat kata pertama pada judul. Apabila kata pertama pada judul merupakan kata ‘salah’, ‘hoax’, ‘disinformasi’ atau ‘misinformasi’ maka narasi kontennya diberi label *hoax* dan bagian penjelasannya

diberi label fakta. Sedangkan apabila kata pertama pada judul adalah kata ‘klarifikasi’ atau ‘benar’ maka bagian penjelasannya diberi label fakta. Berikut modul labeling dapat dilihat pada Modul 4.3.

```

data = []
for row in reader:
    judul = row[0]
    tanggal = row[1]
    kelas = row[0].split(
') [0].translate(str.maketrans(''','' ,string.punctuation))
    narasi = re.findall(r'narasi(.*)?penjelasan',
row[2].replace('\n', ' ').lower())
    penjelasan = re.findall(r'penjelasan(.*)?referensi',
row[2].replace('\n', ' ').lower())
    if kelas == "SALAH" or kelas == "HOAX" or kelas
== "DISINFORMASI" or kelas == "MISINFORMASI" :
        total_data+=1
        if len(narasi) is not 0 :
            if len(narasi) == 1 :
                narasi = narasi[0]
            elif len(narasi) == 2 :
                narasi = narasi[0]+narasi[1]
            elif len(narasi) == 3 :
                narasi = narasi[0]+narasi[1]+narasi[2]
            if len(narasi) > 15 :
                kelas = "HOAX"
                data = [[narasi, kelas]]
                writer.writerows(data)
                total_narasi+=1
                #print("Narasi ", total_data, " : ",
len(narasi))
            if len(penjelasan) == 1 :
                penjelasan = penjelasan[0]
            elif len(penjelasan) == 2 :
                penjelasan = penjelasan[1]
            if len(penjelasan) > 10 :
                kelas = "FAKTA"
                data = [[penjelasan, kelas]]
                writer.writerows(data)
                total_penjelasan+=1
                #print("Penjelasan ", total_penjelasan, " : ",
len(penjelasan))
            elif kelas=="KLARIFIKASI" or kelas=="BENAR" :
                if len(penjelasan) == 1 :
                    penjelasan = penjelasan[0]
                elif len(penjelasan) == 2 :
                    penjelasan = penjelasan[1]

                if len(penjelasan) > 10 :
                    kelas = "FAKTA"
                    data = [[penjelasan, kelas]]
                    writer.writerows(data)
                    total_penjelasan+=1

```

4.1.1.3 Modul *Text Preprocessing*

```

Algoritma Text Preprocessing

Deklarasi:
Data : Array[1..N] of string

Algoritma:
HasilCaseFolding <- CaseFolding(data)
HasilRemovePunctuition <- RemovePunctuation(HasilCaseFolding)
HasilRemoveNumber <- RemovePunctuation(HasilRemovePunctuition)
HasilRemoveWhitespace <- RemovePunctuation(HasilRemoveNumber)
HasilTokenizing <- RemovePunctuation(HasilRemoveWhitespace)
HasilStopwordRemoval <- RemovePunctuation(HasilTokenizing)
HasilStemming <- RemovePunctuation(HasilStopwordRemoval)
HasilPreprocessing <- HasilStemming
OUTPUT(HasilPreprocessing)

```

Pada tahap ini, data konten akan dilakukan proses *text preprocessing* yang bertujuan untuk membersihkan data sehingga siap digunakan. Data yang digunakan yaitu data hasil dari proses labeling. Seperti pada proses sebelumnya, *import* datanya terlebih dahulu dan mengubah label menjadi bernilai 0 atau 1 sebelum dilakukan proses *text preprocessing*.

Berikut modul *import* data proses *text preprocessing*.

```

import pandas as pd

#Import Data
df = pd.read_csv('data_akhir.csv', encoding='utf-8')
df = df.dropna()
df = df.reset_index(drop=True)

#Merubah Label menjadi nilai 0 atau 1
macronum=sorted(set(df['Kelas']))
macro_to_id = dict((note, number) for number, note in
enumerate(macronum))
def fun(i):
    return macro_to_id[i]

df['Kelas']=df['Kelas'].apply(fun)

```

Modul Program 4.4 Pseudocode Proses Case Folding

Proses *import* data pada proses *text preprocessing* ini menggunakan *library* pandas untuk memudahkan proses pengolahan data supaya berbentuk *dataframe*. Membaca file menggunakan pandas menggunakan fungsi `read_csv()`. Tahap selanjutnya yaitu menghapus data kosong menggunakan `dropna()` dan menghapus index pada data menggunakan `reset_index(drop=True)`. Kemudian, label pada setiap data akan diubah

menjadi nilai 0 atau 1. Dimana label fakta diubah menjadi nilai 0 dan label *hoax* menjadi 1.

Hal tersebut karena algoritma hanya bisa menerima *inputan* numerik.

Setelah itu, data tersebut akan dilakukan proses *text preprocessing*. Berikut ini modul-modul yang digunakan pada proses *text preprocessing*.

a. **Modul Case Folding**

Modul *case folding* berfungsi untuk merubah semua karakter pada data konten menjadi huruf kecil (*lowercase*). Apabila pada data konten terdapat huruf kapital maka akan diubah menjadi huruf kecil. Implementasi dari proses *case folding* ini menggunakan fungsi `lower()` yang telah disediakan pada bahasa pemrograman python. Berikut ini adalah modul *case folding*.

```
Algoritma Case Folding
Deklarasi:
Data, HasilCaseFolding : array[1..N] of string
INPUT : data
OUTPUT : HasilCaseFolding

Deskripsi:
Function CaseFolding(data: array of string) : array of string
begin
    Algoritma
        data <- data.lower()
        Return(data)
end

read(data)
HasilCaseFolding <- CaseFolding(data)
```

Modul Program 4.5 Pseudocode Proses Case Folding

b. **Modul Remove Punctuation**

Modul *remove punctuation* merupakan proses untuk menghapus tanda baca pada suatu data karena dianggap sebagai *delimiter*. Data yang digunakan merupakan hasil dari proses *case folding*. Proses dari *remove puctuation* dapat menggunakan *syntax* `translate(str.maketrans(' ', ' ', string.punctuation))` dengan terlebih dahulu `import library string`. Modul *remove punctuation* dapat dilihat pada Modul Program 4.6

```

Algoritma Remove Punctuation

Deklarasi:
HasilRemovePunctuation : array[1..N] of string
INPUT   : HasilCaseFolding
          str.maketrans (Library string)
OUTPUT  : HasilRemovePunctuation

Deskripsi:
Function RemovePunctuation(data: array of string) : array of string
begin
    Algoritma
        data <- data.translate(str.maketrans(' ', ' ', string.punctuation))
        Return(data)
end
read(HasilCaseFolding)
HasilRemovePunctuation <- RemovePunctuation(HasilCaseFolding)

```

Modul Program 4.6 Pseudocode Proses Remove Punctuation

c. Modul Remove Number

Hasil dari proses *remove punctuation* akan dilanjutkan ke proses *remove number*.

Dimana pada proses ini akan menghapus karakter yang berupa angka. Implementasi dari penghapusan angka pada data menggunakan *library re* dan fungsi `sub(r'\d+', '', data)`.

Modul dari proses *remove number* dapat dilihat pada Modul Program 4.7.

```

Algoritma Remove Number

Deklarasi:
HasilRemoveNumber: array[1..N] of string
INPUT   : re.sub (Library re)
          HasilRemovePunctuation
OUTPUT  : HasilRemoveNumber

Deskripsi:
Function RemoveNumber(data: array of string) : array of string
begin
    Algoritma
        data <- re.sub(r'\d+', '', data)
        Return(data)
end
read(HasilRemovePunctuation)
HasilRemoveNumber<- RemoveNumber(HasilRemovePunctuation)

```

Modul Program 4.7 Pseudocode Proses Remove Number

d. Modul Remove Whitespace

Modul *remove whitespace* berfungsi untuk menghapus karakter *whitespace*. Apabila menemukan karakter *whitespace* akan dihapus sedangkan jika bukan karakter *whitespace*

maka karakter tersebut tetap disimpan. Proses ini dapat memanfaatkan fungsi `strip()`.

Berikut modul *remove whitespace* pada Modul Program 4.8.

```
Algoritma Remove Whitespace

Deklarasi:
HasilRemoveWhitespace : array[1..N] of string
INPUT : HasilRemoveNumber
OUTPUT : HasilRemoveWhitespace

Deskripsi:
Function RemoveWhitespace(data: array of string) : array of string
begin
    Algoritma
        data <- data.strip()
        Return(data)
end
read(HasilRemoveNumber)
HasilRemoveWhitespace <- RemoveWhitespace(HasilHasilNumber)
```

Modul Program 4.8 Pseudocode Proses Remove Whitespace

e. **Modul Tokenizing**

Modul *tokenizing* merupakan tahapan pada *text preprocessing* yang digunakan untuk memotong data menjadi per kata dengan spasi (“ “) sebagai pemisahnya. Data yang digunakan sebagai inputan merupakan hasil dari proses *remove whitespace*. Hasil dari proses *tokenizing* ini akan berbentuk token-token. Untuk melakukan proses *tokenizing* dapat memanfaatkan fungsi `word_tokenize()` pada *library nltk* maka akan menghasilkan token-token dari data yang menjadi *inputan*. Modul *tokenizing* ini pada Modul Program 4.9.

```
Algoritma Tokenizing

Deklarasi:
HasilTokenizing : array[1..N] of string
INPUT : HasilRemoveWhitespace
        Word_tokenize (Library nltk)
OUTPUT : HasilTokenizing

Deskripsi:
Function Tokenizing(data: array of string) : array of string
begin
    Algoritma
        data <- word_tokenize(data)
        Return(data)
end
read(HasilRemoveWhitespace)
HasilTokenizing <- Tokenizing(HasilRemoveWhitespace)
```

Modul Program 4.9 Pseudocode Proses Tokenizing

f. Modul Stopword Removal

Hasil dari proses *tokenizing* yang berupa per kata (token) akan dilakukan proses *stopword removal*. Proses *stopword removal* yaitu membuang kata-kata yang kurang penting (*stopword*) pada data seperti kata ganti orang, kata penghubung dan sebagainya. Kumpulan kamus data yang kurang penting (*stopword*) terdapat pada `nltk.corpus`. Prosesnya yaitu dengan melakukan pengecekan setiap token dari hasil *tokenizing*. Apabila token tersebut ada pada kamus *stopword* maka token tersebut akan dihapus, dan berlaku sebaliknya. Modul *stopword removal* dapat dilihat pada Modul Program 4.10.

```

Algoritma Stopword Removal

Deklarasi:
HasilStopwordRemoval, hasil_list : array[1..N] of string
INPUT : HasilTokenizing
        Stopwords (Library nltk)
OUTPUT : HasilStopwordRemoval

Deskripsi:
Function StopwordRemoval(data: array of string) : array of string
begin
    Algoritma
        List_stopword = set(stopwords.words('indonesian'))
        FOR counter = 1 TO length(data) DO
            IF data[counter] NOT IN list_stopword AND data[counter] NOT IN
            hasil_list THEN
                hasil_list.append(data[counter])
            ENDIF
        ENDFOR
        data <- hasil_list
        Return(data)
    end

    read(HasilTokenizing)
    HasilStopwordRemoval<- StopwordRemoval(HasilTokenizing)

```

Modul Program 4.10 Pseudocode Proses Stopword Removal

g. Modul Stemming

Modul *stemming* digunakan untuk mencari kata dasar dari setiap kata hasil *stopword removal* dengan menghapus imbuhan pada masing-masing kata. Hasil dari proses *stemming* ini juga merupakan hasil dari proses *text preprocessing*. Modul *stemming* ini menggunakan *library Sastrawi*. Pada *library* tersebut mengimplementasikan algoritma Nazief & Adriani. Berikut modul *stemming* dapat dilihat pada Modul Program 4.11.

```

Algoritma Stemming

Deklarasi:
HasilStemming : array[1..N] of string
INPUT   : HasilStopwordRemoval
          Stemmer.stem (Library sastrawi)
OUTPUT  : HasilStemming

Deskripsi:
Function Stemming(data: array of string) : array of string
begin
    Algoritma
        FOR counter = 1 TO length(data) DO
            kata = stemmer.stem(data[counter])
            data_akhir.append(kata)
        ENDFOR
        Return(data_akhir)
end

read(HasilStopwordRemoval)
HasilStemming<- Stemming(HasilStopwordRemoval)

```

Modul Program 4.11 Pseudocode Proses Stemming

4.1.1.4 Modul Word Embedding

Pada modul *word embedding* ini akan merubah kata dari hasil *preprocessing* menjadi vektor yang berisi kumpulan angka. Setiap kata akan direpresentasikan oleh vektor. Vektor yang dihasilkan pada proses ini memiliki dimensi 100. Maksud dari dimensi ini yaitu panjang vektor atau banyaknya vektor, jadi satu kata akan dibuat menjadi deretan angka sebanyak 100. Penggunaan modul ini karena algoritma hanya bisa memproses data berbentuk angka dan tidak bisa berbentuk string.

Implementasi dari proses *word embedding* menggunakan `gensim` dengan metode `word2vec`. Data yang menjadi *inputan* yaitu kumpulan token dari hasil proses *text preprocessing*. Parameter dari fungsi `word2vec` terdiri dari `size` dan `min_count`. `Size` untuk menentukan panjang dimensi yang diinginkan, sedangkan `min_count` yaitu minimal kemunculan dari token yang akan diproses. Berikut modul *word embedding* dapat dilihat pada Modul Program 4.12

```

Algoritma Word Embedding
Deklarasi:
INPUT  : data['tokens']
          Word2vec (Library gensim)
OUTPUT : Model, modelakhir.txt

Deskripsi:
Read(data['tokens'])
Model <- Word2Vec(data['tokens'], size=100, min_count=1)
Model.wv.save_word2vec_format('modelakhir.txt', binary=False)

```

Modul Program 4.12 Pseudocode Proses Word Embedding

4.1.1.5 Modul Convolutional Neural Network

Pada tahap ini akan membahas mengenai modul dari algoritma *convolutional neural network*. Proses ini menggunakan data dari hasil *word embedding* pada tahap sebelumnya. Penggunaan algoritma CNN pada penelitian ini terdiri dari 3 bagian yaitu *training*, *testing*, dan prediksi.

a. Modul Training

Tahap ini merupakan proses untuk pembuatan model algoritma CNN untuk klasifikasi berita *hoax*. Pembuatan model ini dengan cara melatih sistem dengan memberikan contoh berita beserta labelnya sebagai bahan pembelajaran bagi model. Modul *training* ini berisi arsitektur CNN yang digunakan mulai dari *input layer* hingga *fully connected layer*. Sebelum membuat arsitektur CNN, *import* terlebih dahulu *library* yang dibutuhkan. *Library* yang dibutuhkan pada proses *training* ini dapat dilihat pada Modul 4.13

```

from keras.layers import Embedding
from keras.layers import Dense, Input, Flatten
from keras.layers import Conv1D, MaxPooling1D, Embedding, Dropout
from keras.models import Model
from keras.callbacks import ModelCheckpoint
from keras.callbacks import EarlyStopping

```

Modul Program 4.13 Pseudocode Import Library Proses Training

Setelah itu, membuat arsitektur dari algoritma CNN. Berdasarkan rancangan yang telah dibuat, arsitektur CNN ini terdiri dari layer *input* dengan data yang bersumber dari hasil *word embedding*. Lalu, proses *convolutional layer* dengan parameter aktivasinya menggunakan ReLU. Banyaknya dimensi yang dihasilkan pada layer konvolusi ini adalah

128 dan ukuran kernel yang digunakan yaitu 5x5. Hasil dari proses konvolusi ini digunakan sebagai *input* pada proses *pooling* dengan ukuran filter 5x5 menggunakan *MaxPooling* yaitu mengambil nilai tertinggi. Kemudian, pada *fully connected layer* akan dilakukan proses *flatten* dan *dense* untuk menghasilkan *output* menjadi 2 kelas yaitu 0 untuk fakta dan 1 untuk *hoax*. Untuk menghasilkan probabilitas masing-masing kelas menggunakan aktivasi *softmax*. *Pseudocode* arsitektur CNN pada proses *tarining* dapat dilihat pada Modul 4.14

```

Algoritma Arsitektur Convolutional Neural Network

Deklarasi:
INPUT : train_word_index, train_embedding_weights
        Embedding, Input, Model, Conv1D (Library keras)
        MaxPooling1D, Flatten, Dense (Library keras)
OUTPUT : model

Deskripsi:
Read(train_word_index)
Read(train_embedding_weights)
EMBEDDING_DIM <- 100
MAX_SEQUENCE_LENGTH <- 1000
Embedding_layer <- Embedding(length(train_word_index)+1,
                                EMBEDDING_DIM,
                                weights=[train_embedding_weights],
                                input_length=MAX_SEQUENCE_LENGTH,
                                trainable=True)
Sequence_input <- Input(shape=(MAX_SEQUENCE_LENGTH,), dtype='int32')
embedded_sequences <- embedding_layer(sequence_input)
l_cov1 <- Conv1D(128, 5, activation='relu')(embedded_sequences)
l_pool1 <- MaxPooling1D(5)(l_cov1)
l_cov2 <- Conv1D(128, 5, activation='relu')(l_pool1)
l_pool2 <- MaxPooling1D(5)(l_cov2)
l_cov3 <- Conv1D(128, 5, activation='relu')(l_pool2)
l_pool3 <- MaxPooling1D(35)(l_cov3) # global max pooling
l_flat <- Flatten()(l_pool3)
l_dense <- Dense(128, activation='relu')(l_flat)
preds <- Dense(2, activation='softmax')(l_dense)

model <- Model(sequence_input, preds)
model.compile(loss='binary_crossentropy',
               optimizer='adam',
               metrics=['acc'])

OUTPUT("Simplified convolutional neural network")
model.summary()

```

Modul Program 4.14 Pseudocode Arsitektur CNN Proses Training

Berdasarkan Modul 4.14, fungsi `Input()` digunakan sebagai *input layer* dengan parameter `shape` untuk ukuran dimensi yang dihasilkan. Lalu, proses *convolutional layer* fungsi `Conv1D()` pada keras. Penggunaan fungsi tersebut karena data pada penelitian ini

berbentuk 1 dimensi. Pada fungsi tersebut juga dapat menyertakan fungsi aktivasi yang digunakan yaitu ReLU. Sedangkan untuk proses *pooling* menggunakan `MaxPooling1D()` dengan parameternya yaitu ukuran filter. Setelah itu, proses *fully connected layer* yang terdiri dari *flatten* dan *dense* dengan aktivasi *softmax*. Setelah semua layer telah dibuat, maka dapat dibentuk sebuah model CNN. Proses pembuatan modelnya dengan memanfaatkan fungsi `compile()` pada keras. Arsitektur dari model CNN yang telah dibuat dapat dilihat pada

Gambar 4.1.

Simplified convolutional neural network		
Model: "functional_17"		
Layer (type)	Output Shape	Param #
input_10 (InputLayer)	[None, 1000]	0
embedding_9 (Embedding)	(None, 1000, 100)	2547800
conv1d_27 (Conv1D)	(None, 996, 128)	64128
max_pooling1d_27 (MaxPooling (None, 199, 128))		0
conv1d_28 (Conv1D)	(None, 195, 128)	82048
max_pooling1d_28 (MaxPooling (None, 39, 128))		0
conv1d_29 (Conv1D)	(None, 35, 128)	82048
max_pooling1d_29 (MaxPooling (None, 1, 128))		0
flatten_9 (Flatten)	(None, 128)	0
dense_17 (Dense)	(None, 128)	16512
dense_18 (Dense)	(None, 2)	258

Total params: 2,792,794
 Trainable params: 2,792,794
 Non-trainable params: 0

Gambar 4.1 Arsitektur Model CNN Proses *Training*

Selanjutnya yaitu melakukan proses *training* pada model yang telah dibuat. Data yang digunakan yaitu data *train* yang telah dibagi sebelumnya menggunakan *k-fold cross validation*. Model akan dilatih secara berulang dengan memberikan contoh konten berita beserta labelnya. Model akan disimpan hanya nilai terbaik dengan format hdf5. Untuk menyimpan model tersebut menggunakan `ModelCheckpoint()`. *Training* ini menggunakan *epoch* atau perulangan sebanyak 100x dengan batch size 32. Selama proses *training* akan dimonitor nilai *validation loss* menggunakan `EarlyStopping()` dengan minimum

perulangan sebanyak 10x. Apabila metriknya tidak terjadi peningkatan maka perulangan akan berhenti. Proses *training* ini menggunakan fungsi *fit()* dengan menyertakan data train beserta labelnya dan *callback* yang digunakan. Berikut proses *training* dapat dilihat pada

Modul 4.15

```
Algoritma Training Convolutional Neural Network

Deklarasi:
INPUT : x_train, y_tr
        ModelCheckpoint, EarlyStopping (Library keras)
OUTPUT : model_cnn.hdf5, hist
cp=ModelCheckpoint('model_cnn.hdf5',monitor='val_acc',verbose=1,save_be
st_only=True)
num_epochs <- 100
batch_size <- 32
es_callback =EarlyStopping(monitor='val_loss', patience=10, verbose=1)
hist = model.fit(x_train, y_tr, epochs=num_epochs,
                  validation_split=0.2, shuffle=True ,
                  batch_size=batch_size,callbacks=[es_callback])
```

Modul Program 4.15 Pseudocode Proses Training

b. Modul Testing

Setelah melakukan proses *training*, model akan dilakukan proses *testing* dengan diberikan kumpulan data tanpa memberikan label. Lalu akan diprediksi oleh model dan dihitung total data yang diprediksi dengan benar dan total data yang diprediksi dengan jawaban yang salah. Kemudian, dihitung nilai akurasi, presisi, dan recall menggunakan *confusion matrix* dan kurva ROC pada setiap iterasi *cross validation*. Modul untuk proses *testing* dapat dilihat pada Modul 4.16.

```
Algoritma Testing Convolutional Neural Network

Deklarasi:
Prediction_labels, y_test, y_pred : array[1..N] of string
INPUT : model
        StratifiedKFold, auc, plot_roc_curve (Library sklearn)
        Argmax (Library numpy)
        DataFrame (Library pandas)
        Plot (Library matplotlib)
OUTPUT : prediction, akurasi, presisi, recall, auc

Deskripsi:
predictions <- model.predict(test_cnn_data, batch_size=1, verbose=1)
labels = [0,1]
FOR p = 1 TO length(predictions) DO
    prediction_labels.append(labels[np.argmax(predictions[p])])
ENDFOR
```

```

Akurasi <- sum(y[test]==prediction_labels)/length(prediction_labels)
y_test <- y[test].values
y_tes=[]
FOR i = 1 TO length(y_test) DO
    y_tes.append(y_test[i])
ENDFOR
y_pred=[]
FOR i = 1 TO length(predictions) DO
    y_pred.append(labels[np.argmax(predictions[i])])
ENDFOR

cf_klasifikasi <- pd.DataFrame(data=confusion_matrix(y_tes, y_pred,
labels=labels), columns=labels, index=labels)
OUTPUT(cf_klasifikasi)

tps_klasifikasi = {}
fps_klasifikasi = {}
fns_klasifikasi = {}
tns_klasifikasi = {}

FOR label = 1 TO length(labels) DO
    tps_klasifikasi[label] <- cf_klasifikasi.loc[label, label]
    fps_klasifikasi[label] <- cf_klasifikasi[label].sum() -
    tps_klasifikasi[label]
    fns_klasifikasi[label] <- cf_klasifikasi.loc[label].sum() -
    tps_klasifikasi[label]
ENDFOR

FOR label = 1 TO length(y_tes) DO
    tns_klasifikasi[label] <- length(y_tes) - (tps_klasifikasi[label] +
    fps_klasifikasi[label] + fns_klasifikasi[label])

accuracy_global_new_klasifikasi <- sum( tps_klasifikasi.values())/
len(y_tes)
acc.append(accuracy_global_new_klasifikasi)

tpfp_klasifikasi = [ai + bi for ai, bi in
zip(list(tps_klasifikasi.values()), list(fps_klasifikasi.values()))]

precision=[ai / bi if bi>0 else 0 for ai , bi in
zip(list(tps_klasifikasi.values()), tpfp_klasifikasi)]
precisionKlasifikasi=sum(precision)/2
presisi.append(precisionKlasifikasi)

tpfn_klasifikasi = [ai + bi for ai, bi in
zip(list(tps_klasifikasi.values()), list(fns_klasifikasi.values()))]
recall=[ai / bi if bi>0 else 0 for ai, bi in
zip(list(tps_klasifikasi.values()), tpfn_klasifikasi)]
recallKlasifikasi=sum(recall)/2
recal.append(recallKlasifikasi)

fpr, tpr, thresholds = roc_curve(y_test, y_pred)
nilai_auc = auc(fpr, tpr)
interp_tpr = interp(mean_fpr, fpr, tpr)
interp_tpr[0] = 0.0
tprs.append(interp_tpr)
aucs.append(nilai_auc)
ax.plot(mean_fpr, interp_tpr, label=r'ROC Fold %d (AUC = %f)' % (fold,
nilai_auc), lw=1, alpha=0.3)

```

c. Modul Prediksi

Model algoritma CNN telah dibuat apabila proses *training* dan *testing* telah dilakukan. Selanjutnya terdapat modul prediksi yang berfungsi untuk melakukan prediksi terhadap data yang tidak diketahui labelnya menggunakan model yang telah dibuat. Berbeda dengan proses *testing*, proses prediksi tidak diketahui labelnya sedangkan *testing* diketahui nilai labelnya namun labelnya sebagai pembanding. Proses prediksi ini dengan melakukan *load model* terlebih dahulu. Selanjutnya data dilakukan proses *preprocessing* hingga *word embedding*. Kemudian, data tersebut dilakukan proses prediksi menggunakan model yang telah dibuat. Modul prediksi dapat dilihat pada Modul 4.17.

```
Algoritma Prediksi Convolutional Neural Network

Deklarasi:
Preprocessing : Array[1..N] of string
INPUT : open, json_file.read (Library json)
        Pickle.load (Library pickle)
OUTPUT : klasifikasi

Deskripsi:
#Load Model
json_file =
open('C:/xampp/htdocs/hoax_classification/model/model_cnn.json', 'r')
model_json = json_file.read()
model = model_from_json(model_json)
model.load_weights("C:/xampp/htdocs/hoax_classification/model/model_cnn.h5")

with open('C:/xampp/htdocs/hoax_classification/model/tokenizer.pickle',
'rb') as handle:
    tokenizer = pickle.load(handle)

preprocessing <- [' '.join(sen) for sen in hasil_stemming]
sequences <- tokenizer.texts_to_sequences(hasil_stemming)
str_sequences <- str(sequences)

word_index <- tokenizer.word_index
x_test <- pad_sequences(sequences, maxlen=1000)

prediksi <- model.predict(x_test, batch_size=1, verbose=0)
probabilitas <- "FAKTA : " + str(prediksi[0][0]) + " | HOAX : " +
str(prediksi[0][1])
probabilitas_pembulatan <- "FAKTA : " + str(round(prediksi[0][0])) + " |
HOAX : " + str(round(prediksi[0][1]))

class_category = ['Fakta', 'Hoax']
for i in range(prediksi.shape[0]):
    klasifikasi = class_category[prediksi[i].argmax()]
```

Modul Program 4.17 Pseudocode Proses Prediksi

4.1.2 Implementasi Aplikasi

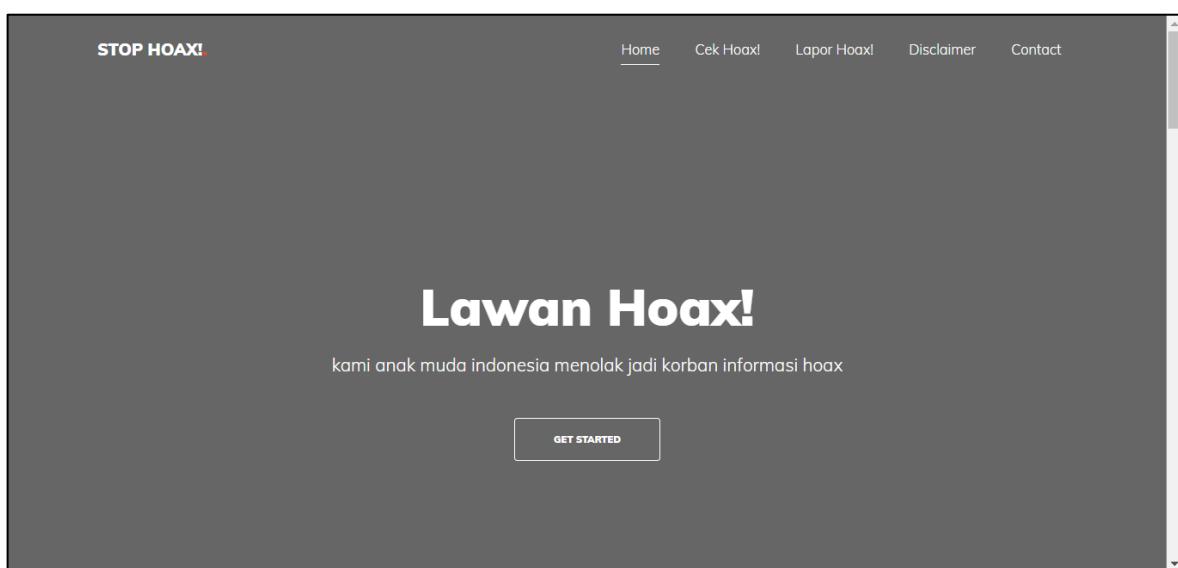
Implementasi aplikasi yaitu bagian yang menampilkan hasil dari rancangan aplikasi yang telah dibuat pada bab sebelumnya. Rancangan aplikasinya berisi tampilan *user interface* beserta fungsi-fungsinya. Secara garis besar, terdapat dua bagian yaitu halaman *user* dan halaman admin.

4.1.2.1 Halaman User

Pada bagian ini akan dibahas tentang halaman-halaman yang dapat diakses oleh *user*. Berikut halaman *user* yang terdiri dari beberapa bagian, yaitu:

1. Halaman *Home*

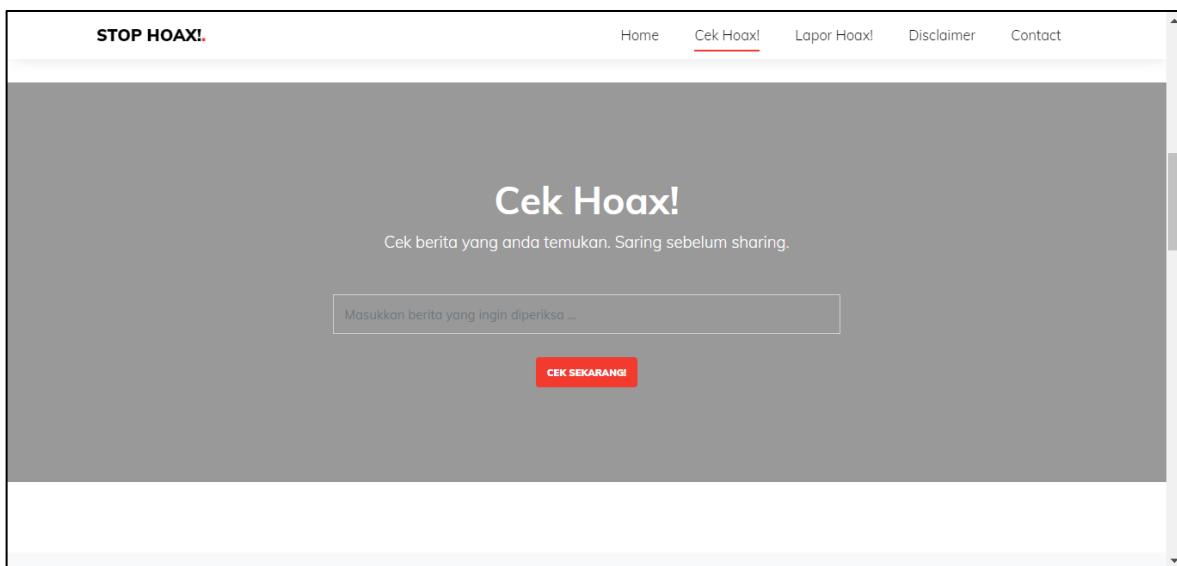
Halaman *home* merupakan halaman awal yang akan tampil saat *user* mengakses sistem. Terdapat judul dan slogan pada halaman ini. Selain itu, pada halaman ini juga ada button *Get Started* untuk menuju ke halaman *Cek Hoax*. Tampilan dari halaman *home* dapat dilihat pada Gambar 4.2.



Gambar 4.2 Halaman *Home*

2. Halaman Cek Hoax

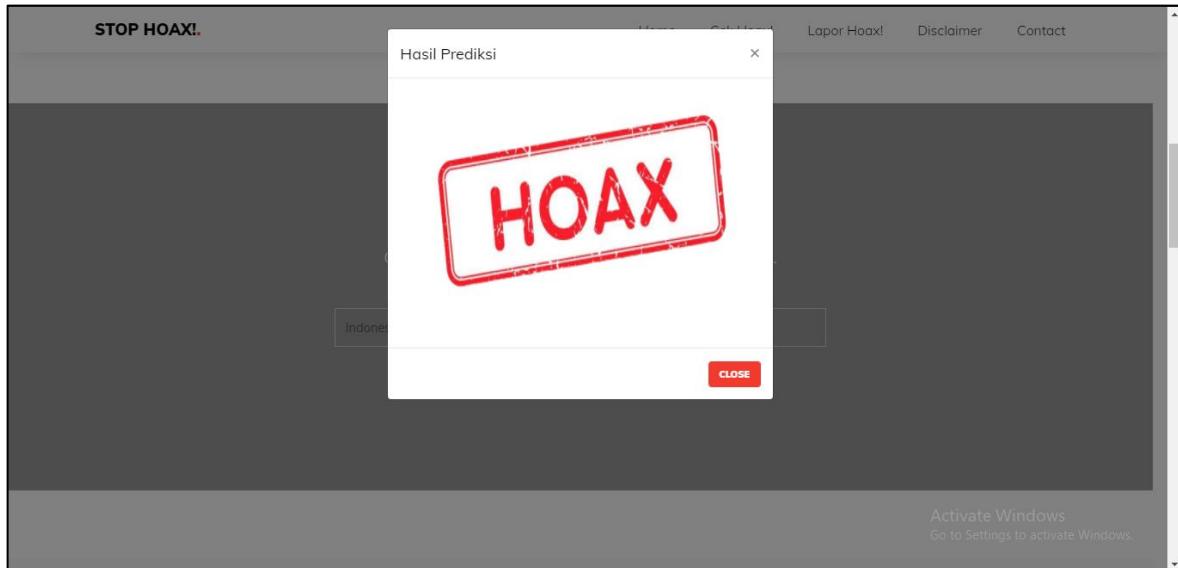
Halaman *cek hoax* merupakan halaman *user* yang berfungsi untuk melakukan pengecekan suatu konten berita berdasarkan hasil prediksi dari model CNN yang telah dibuat. Untuk melakukan proses prediksi suatu konten berita, *user* memasukkan berita tersebut ke dalam *field* yang telah tersedia. Lalu, menekan tombol *Cek Sekarang* dan berita tersebut akan diproses oleh sistem. Implementasi dari halaman *cek hoax* dapat dilihat pada Gambar 4.3.



Gambar 4.3 Halaman *Cek Hoax*

3. Halaman Hasil Prediksi

Setelah berita yang dimasukkan oleh *user* telah di proses oleh sistem maka akan muncul halaman hasil prediksi. Halaman ini berbentuk *pop up* dengan menampilkan hasil prediksi yang menunjukkan berita yang dicek termasuk ke dalam berita *hoax* atau fakta. Tampilan dari halaman hasil prediksi dapat dilihat pada Gambar 4.4.



Gambar 4.4 Halaman Hasil Prediksi

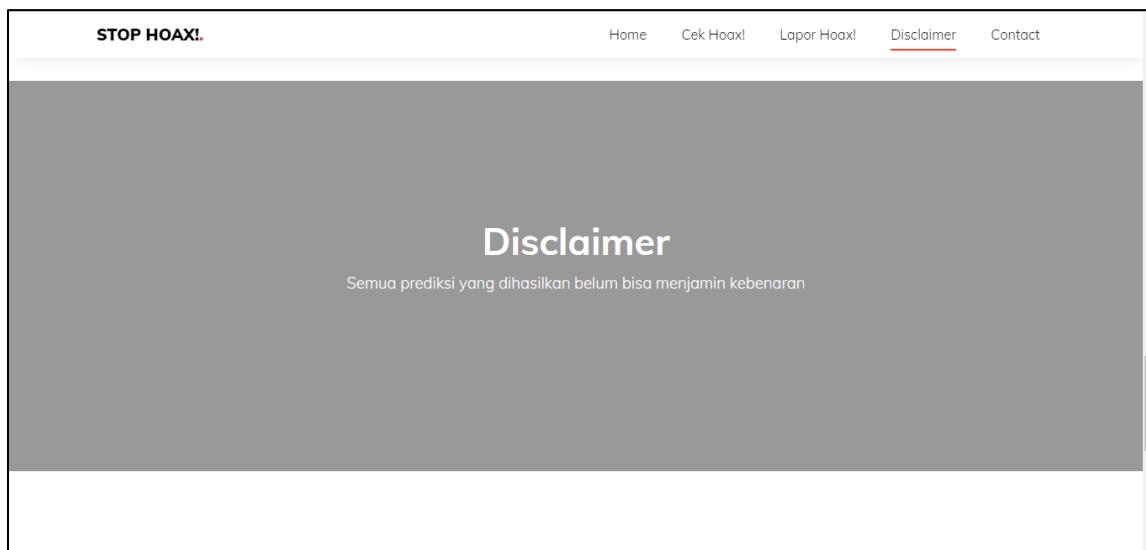
4. Halaman Lapor *Hoax*

Halaman lapor *hoax* merupakan halaman yang berfungsi sebagai tempat untuk melaporkan suatu berita apabila *user* menemukan berita *hoax* atau fakta di media sosial. Halaman ini terdapat *form* yang berisi identitas orang yang melapor, email dan konten berita yang ingin dilaporkan. Apabila telah dilaporkan, maka berita tersebut akan di validasi kebenarannya. Tampilan dari halaman lapor *hoax* dapat dilihat pada Gambar 4.5

Gambar 4.5 Halaman Lapor Hoax

5. Halaman *Disclaimer*

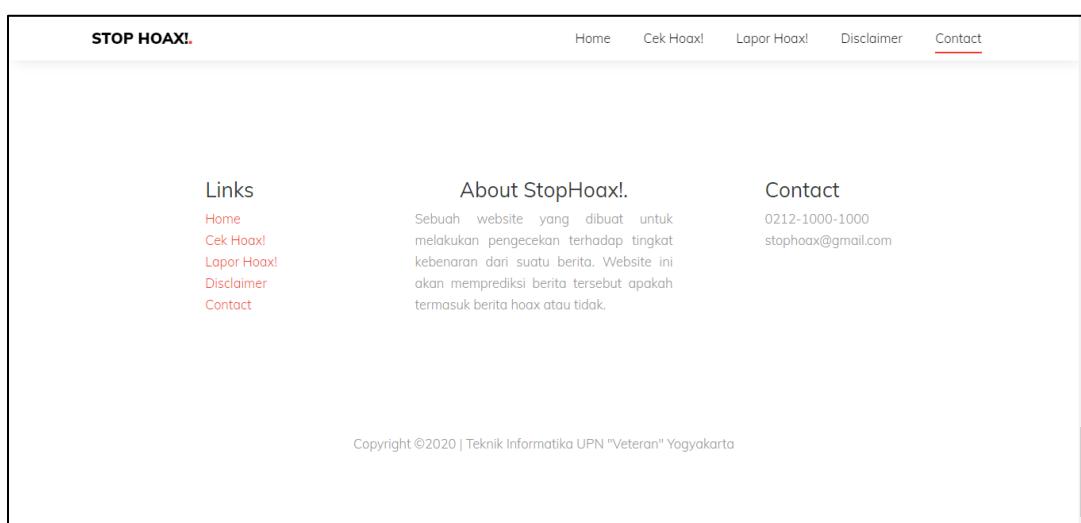
Halaman *disclaimer* ini berisi batasan atau ruang lingkup dari sistem ini yang hanya menampilkan hasil prediksi dan tidak menjamin kebenaran dari hasil prediksi tersebut. Halaman dari *disclaimer* ini dapat dilihat pada Gambar 4.6



Gambar 4.6 Halaman *Disclaimer*

6. Halaman *Contact*

Halaman *contact* adalah halaman memuat informasi yang berkaitan dengan sistem ini seperti link menuju ke halaman lain, penjelasan singkat tentang sistem yang telah dibuat dan kontak yang dapat dihubungi. Tampilan dari halaman *contact* ini dapat dilihat pada Gambar 4.7.



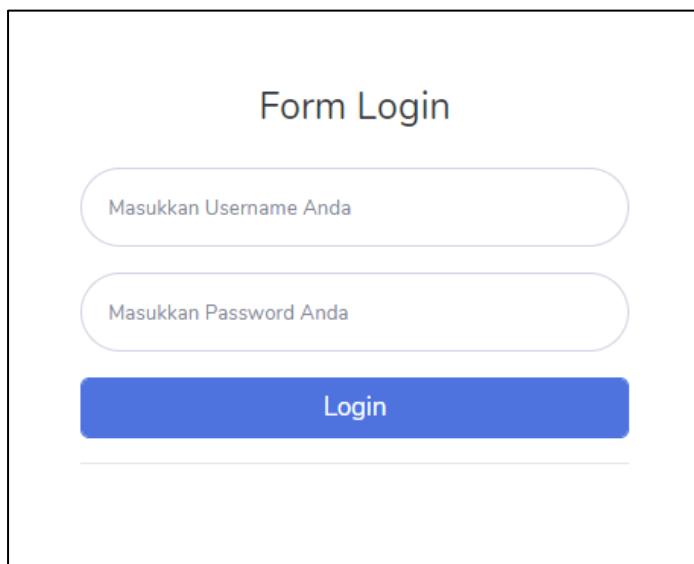
Gambar 4.7 Halaman *Contact*

4.1.2.2 Halaman Admin

Pada bagian ini berisi halaman-halaman yang dapat diakses oleh admin. Halaman untuk admin terdiri dari beberapa bagian, yaitu:

1. Halaman *Login*

Halaman *login* merupakan halaman awal yang akan tampil saat ingin masuk ke dalam sistem sebagai admin. Pada halaman ini harus mengisi *username* dan *password* dengan benar terlebih dahulu. Apabila berhasil maka akan masuk ke dalam sistem sebagai admin. Tampilan dari halaman *login* dapat dilihat pada Gambar 4.8.

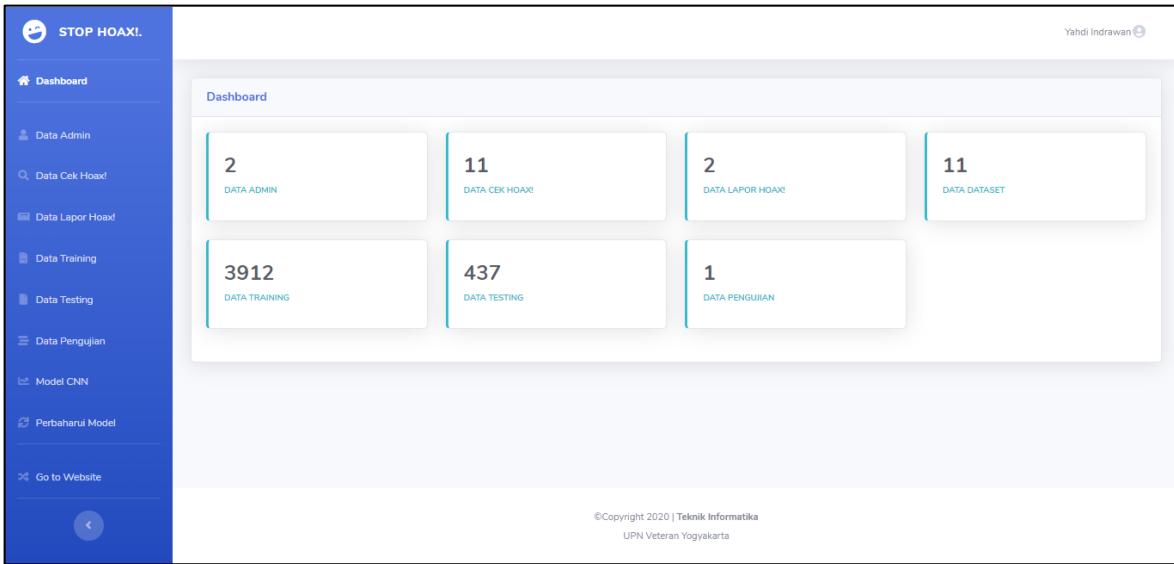


The diagram shows a wireframe of a login interface. At the top center, it says "Form Login". Below that is a horizontal input field with rounded ends, containing the placeholder text "Masukkan Username Anda". Below it is another similar input field with the placeholder text "Masukkan Password Anda". At the bottom is a wide, blue rectangular button with the word "Login" in white capital letters.

Gambar 4.8 Halaman *Login*

2. Halaman *Dashboard*

Apabila admin telah berhasil melakukan *login* maka akan tampil halaman *dashboard*. Halaman ini memuat ringkasan informasi dari data yang diolah oleh admin. Halaman *dashboard* akan menampilkan jumlah data dari masing-masing tabel yaitu jumlah data admn, data cek *hoax*, lapor *hoax*, dataset, data training dan data testing serta data pengujian. Tampilan dari halaman *dashboard* ini seperti pada Gambar 4.9.



Gambar 4.9 Halaman Dashboard

3. Halaman Data Admin

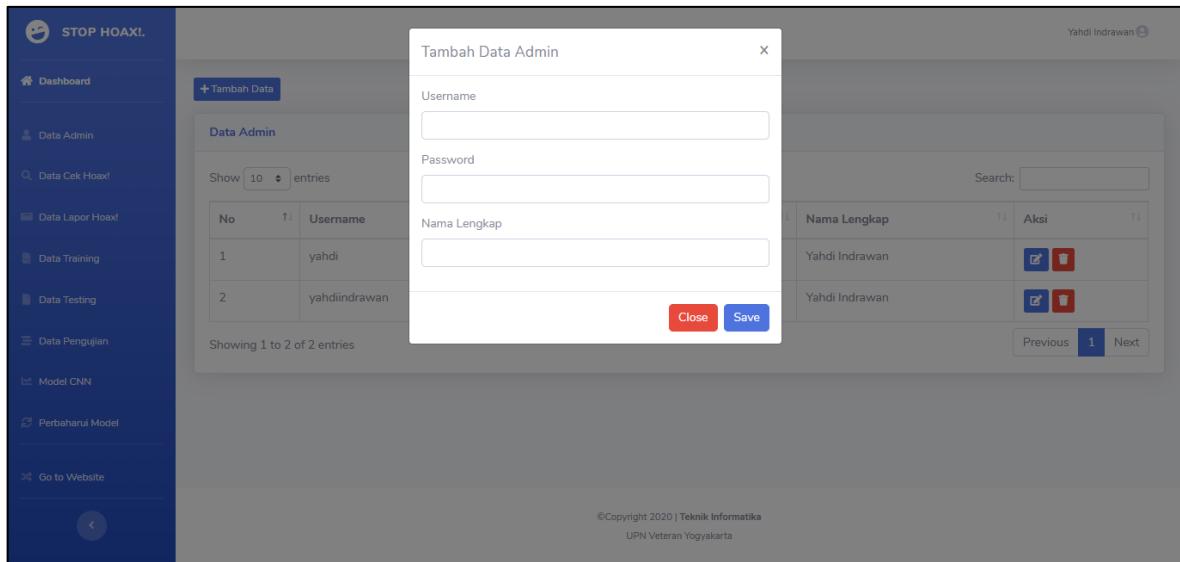
Halaman data admin akan menampilkan daftar admin yang tersedia. Data ini ditampilkan dalam bentuk tabel yang berisi username, *password* yang telah di enkripsi dan nama lengkap. Melalui halaman ini dapat menambah daftar dari data admin, mengedit data admin dan menghapus data admin. Tampilan dari halaman data admin ini seperti pada Gambar 4.10

No	Username	Password	Nama Lengkap	Aksi
1	yahdi	eWFoZGk=	Yahdi Indrawan	
2	yahdiindrawan	eWFoZGlpbmRyYXdhbg==	Yahdi Indrawan	

Gambar 4.10 Halaman Data Admin

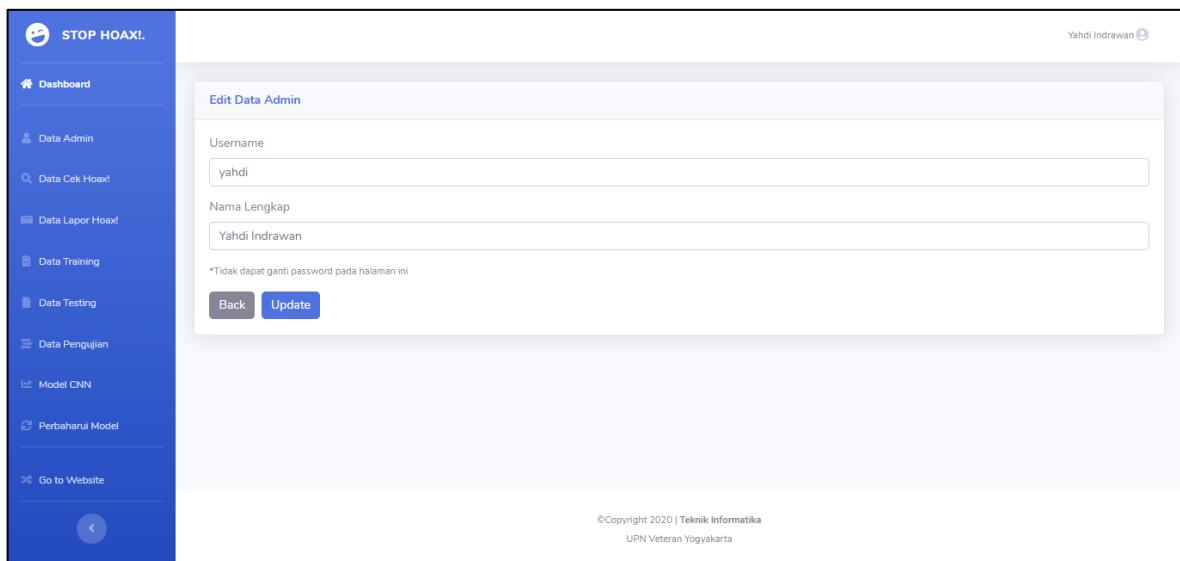
Pada fitur tambah data akan muncul *pop up* berisi form dengan *field username*, *password* dan nama lengkap dari identitas admin yang ingin dibuat. Jika *form* telah diisi

tekan tombol *save* untuk menyimpan data tersebut. *Pop up* dari proses tambah data admin dapat dilihat pada Gambar 4.11



Gambar 4.11 Halaman Tambah Data Admin

Selain itu, pada halaman data admin dapat melakukan edit data admin. Fitur ini untuk mengubah data admin sesuai dengan baris yang dipilih. Data admin yang dapat diubah hanya *username* dan nama lengkap, sedangkan *password* hanya dapat diubah oleh admin yang bersangkutan pada menu profil. Apabila data admin sudah diubah, tekan tombol *update* untuk menyimpan perubahan tersebut. Tampilan dari halaman edit data admin dapat dilihat pada Gambar 4.12



Gambar 4.12 Halaman Edit Data Admin

Lalu, ada fitur untuk menghapus data admin. Untuk menghapus data admin dengan memilih baris yang ingin di hapus dan tekan *button* bergambar tempat sampah maka data adminnya akan terhapus.

4. Halaman Data Cek Hoax

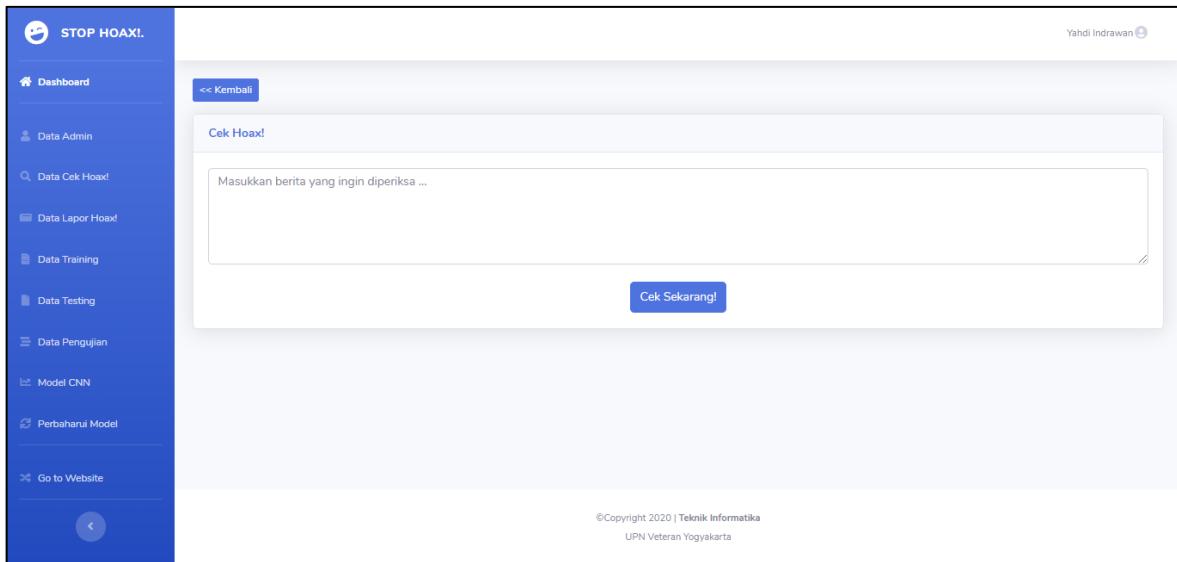
Pada bagian ini akan menampilkan data cek *hoax*. Data ini merupakan hasil pengecekan dari konten berita yang diprediksi oleh model. Isi dari data cek *hoax* yang ditampilkan pada halaman ini yaitu no, tanggal, konten, dan hasil klasifikasi dari model serta kolom untuk proses olah data. Fitur yang tersedia pada halaman data cek *hoax* seperti fitur cek *hoax* seperti pada halaman *user*, lihat detail dari data cek *hoax* dan fitur untuk menghapus data cek *hoax*. Halaman ini tidak memiliki fitur untuk edit data dengan tujuan untuk meminimalisir adanya manipulasi data hasil prediksi. Sehingga hasil yang ditampilkan merupakan hasil *real* dari model yang telah dibuat. Tampilan dari halaman data cek *hoax* dapat dilihat pada Gambar 4.13

No	Tanggal	Konten	Klasifikasi	Aksi
1	2020-08-10	Jokowi Pancasilais	Hoax	- -
2	2020-08-10	Jokowi PKI	Hoax	- -
3	2020-08-10	Dia lupa dengan dirinya sendiri, saat Pilkada DKI 2017 @AgusYudhoyono dipaksa ikut Pilgub oleh @SBYudhoyono AHY Aktif di TNI berpangkat Mayor dipaksa mundur gegara ikutkan Pilgub dan berakhir...	Hoax	- -
4	2020-08-07	Ass Wr Wb, Bpk/Ibu Ketua RT dan RW. Mohon diinformasikan ke Warga, Saudara, Keluarga dan kenalan Anda !!! Baru saja mendapat pesan. Sebuah peringatan !! Sekarang ada yang baru dan sedang terjadi....	Hoax	- -
5	2020-03-25	Wakil Ketua Komisi Pemberantasan Korupsi (KPK) Nurul Ghufron mengatakan, proses pengadaan barang dan jasa untuk penanganan virus Corona dapat dilakukan dengan penunjukan langsung. Ghufron...	Fakta	- -

Gambar 4.13 Halaman Data Cek Hoax

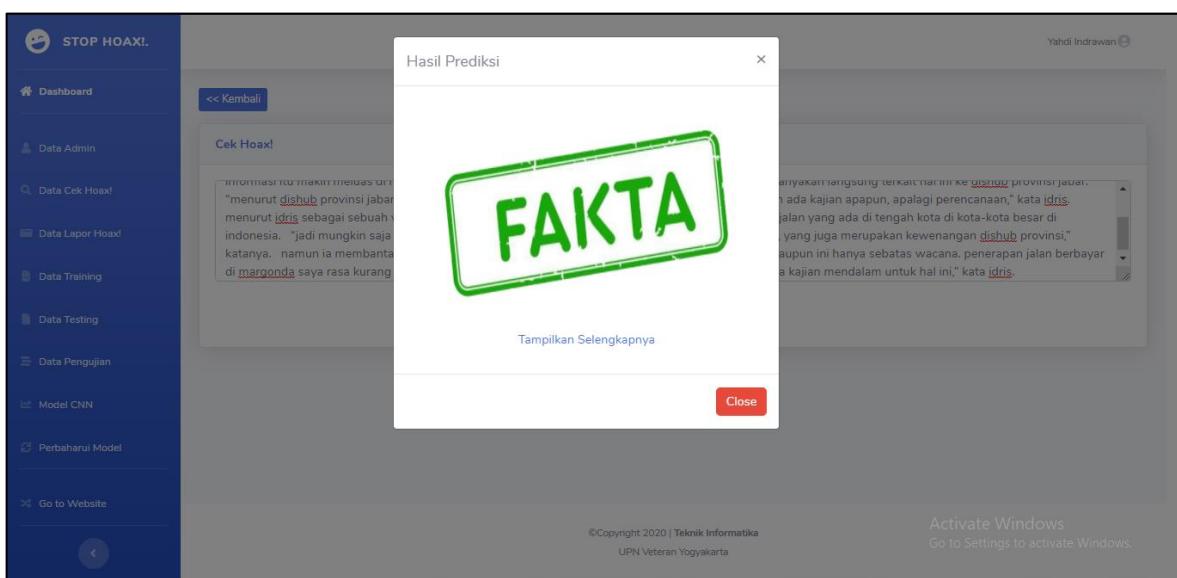
Selanjutnya yaitu fungsi cek *hoax*. Fitur ini untuk melakukan pengecekan suatu konten berita seperti pada halaman *user*. Untuk melakukan pengecekan berita ini dengan

menginputkan konten berita yang ingin diperiksa kebenarannya melalui proses prediksi menggunakan model yang telah dibuat. Halaman ini dapat dilihat pada Gambar 4.14



Gambar 4.14 Halaman Cek Hoax Admin

Setelah konten berita telah diproses maka akan muncul hasil prediksi dari model dalam bentuk *pop up*. Isi dari *pop up* yaitu konten tersebut termasuk ke dalam *hoax* atau fakta. Selain itu, ada *button* untuk melihat tampilan selengkapnya. Hasil prediksi ini memiliki tampilan seperti pada Gambar 4.15



Gambar 4.15 Halaman Hasil Prediksi Admin

Setelah itu, ada fitur untuk melihat detail dari data cek *hoax*. Pada bagian ini akan ditampilkan hasil dari semua tahapan saat proses prediksi. Halaman ini berisi tanggal pengecekan, konten yang diperiksa, hasil dari proses *preprocessing*, proses prediksi dan hasil klasifikasinya. Tampilan dari detail data cek *hoax* dapat dilihat pada Gambar 4.16.

The screenshot shows the STOP HOAXI application interface. On the left, there's a sidebar with a blue header 'STOP HOAXI.' and a list of menu items: Dashboard, Data Admin, Data Cek Hoax, Data Lapor Hoax, Data Training, Data Testing, Data Pengujian, Model CNN, Perbaikui Model, and Go to Website. The main content area has a title 'Detail Data Cek Hoax!' and a timestamp 'Tanggal 2020-08-10'. Below it is a section titled 'Konten' containing a text snippet. Underneath is a section titled 'Proses Preprocessing' with three sub-sections: 'Case Folding', 'Remove Punctuation', and 'Remove Number', each showing the original text followed by its processed version.

Gambar 4.16 Halaman Detail Cek Hoax Admin

5. Halaman Data Lapor *Hoax*

Halaman ini akan memuat data-data laporan yang dikirim oleh *user* ataupun yang diinput oleh admin. Data lapor *hoax* ini berisi nama pelapor, email, konten dan data saat melakukan pelaporan. Selain itu, halaman ini juga dapat melakukan olah data terhadap data lapor *hoax* seperti tambah laporan, edit laporan dan menghapus laporan. Halaman dari data lapor *hoax* ini seperti pada Gambar 4.17.

No	Nama Pelapor	Email	Konten	Tanggal	Aksi
1	Yahdi Indrawan	yahdiindrawan11@gmail.com	Ledakan di Lebanon terjadi karena serangan Israel	2020-08-07	
2	yahdi indrawan	yadhi@gmail.com	Beredar postingan video yang diklaim sebagai bencana alam puting beliung di Sukabumi. Postingan tersebut beredar di Twitter dan dibagikan pada tanggal 3 Maret 2020. Berdasarkan hasil penelusuran,...	2020-03-11	

Showing 1 to 2 of 2 entries

©Copyright 2020 | Teknik Informatika
UPN Veteran Yogyakarta

Gambar 4.17 Halaman Data Lapor Hoax

Berikutnya yaitu bagian tambah laporan pada data lapor *hoax*. Untuk membuat laporan dengan mengisi form yang tersedia seperti identitas pelapor, email, konten dan tanggal pelaporan apabila adminmenerima aduan dari platform lain. Tampilan dari tambah lapor *hoax* ini dapat dilihat pada Gambar 4.18.

Gambar 4.18 Halaman Tambah Lapor Hoax

Apabila terdapat data yang ingin diubah maka dapat memanfaatkan fitur edit data pada *button* bergambar pensil. Maka akan muncul form yang telah terisi sesuai dengan datanya, lalu admin dapat merubah data tersebut untuk melakukan *update* data. Halaman dari edit data lapor *hoax* dapat dilihat pada Gambar 4.19.

Gambar 4.19 Halaman Edit Lapor Hoax

6. Halaman Data Training

Halaman ini menampilkan semua data yang menjadi data training. Data training ini terdiri dari konten dan label. Pada bagian ini tidak memiliki operasi olah data seperti pada menu lainnya, hanya memiliki fitur informasi dengan icon i untuk melihat data secara lengkap. Tampilan dari halaman data training dapat dilihat pada Gambar 4.20

No	Konten	Label	Aksi
1	: pemimpin redaksi koran tempo budi setyarsa memberi klarifikasi terkait dengan beredarnya cover palsu surat kabar tersebut. sampul yang ilustrasinya seolah-olah halaman depan koran tempo tersebut...	Fakta	
2	"bukan main!", =====	Hoax	
3	: kematian siyono di tangan densus 88 telah mencatatkan dirinya sebagai korban ke 121 terduga teroris yang langsung dibunuh tanpa bukti. terdapat kisah menarik dalam jasad siyono, jasadnya tidak...	Hoax	
4	"anggota panwaslu dan kpu buka belang lembaganya dalam forum rapat", salinan narasi oleh sumber-sumber lainnya di (5) bagian referensi. =====	Hoax	
5	: "pki ketangkap basah...!! #breaking_news ; acara pki di klender jakarta timur di bubarkan oleh tni, fpi, fbr dan bang japar....!! makin biadab,, makin jadi,, makin kurang jajar komunis...!! gebug,, ...	Hoax	
6	"wali kota kuttinggi ramalan nurmatias menegaskan bahwa kebakaran pasar aur kuning murni musibah. untuk itu, dia mewanti-wanti dan meminta semua pihak untuk tidak memprovokasi pedagang. "belum...	Fakta	
7	: beredar isu yang menyebutkan bahwa dalam waktu dekat ini kebijakan electronic road pricing (erp) alias jalan berbayar akan diterapkan di ruas jalan margonda, depok. isu tersebut langsung dibantah..	Fakta	

Gambar 4.20 Halaman Data Training

Saat admin menekan tombol i maka akan muncul halaman detail data *training*. Halaman ini akan menampilkan data *training* secara lengkap karena pada halaman

sebelumnya data training yang ditampilkan hanya potongan data. Berikut tampilan detail data training pada Gambar 4.21

STOP HOAXI.

Detail Data Training

Konten

: pemimpin redaksi koran tempo budi setyarto memberi klarifikasi terkait dengan beredarnya cover palsu surat kabar tersebut. sampul yang ilustrasinya seolah-olah halaman depan koran tempo tersebut beredar di media sosial pada senin, 13 februari 2017. sampul depan koran tempo dengan headline berjudul "ahok: jika kami menang gereja dijakarta harus lebih besar dari istqlal", kata budi, dipastikan hasil rekayasa. "gambar yang disebar itu rekayasa dari halaman depan koran tempo edisi 27 juni 2016," ucapnya, senin, 13 februari 2017. sampul asli koran tempo edisi 27 juni 2016 berjudul "dki beli tanah rp 648 miliar milik sendiri" dengan gambar gubernur dki basuki tjahaja purnama. cover ini menampilkan berita tentang pemerintah provinsi dki membeli lahan seluas 4,6 hektare senilai rp 648 miliar di jalan lingkar luar cengkareng, jakarta barat, yang ternyata milik dinas kelaatan, pertanian, dan ketahanan pangan. sejumlah kejanggalan terlihat pada sampul palsu, antara lain judulnya yang panjang, terdiri atas lima baris. kalimat dalam judul pun tidak sesuai dengan kaidah bahasa indonesia, selain itu, tulisan "djakarta" salah, seharusnya "di jakarta", sedangkan pada cover asli, judulnya hanya tiga baris dan memenuhi kaidah bahasa yang benar. mengingat sampul tersebut rekayasa, budi berharap publik tidak mempercayai, budi menduga pembuat sampul koran tempo palsu itu punya kepentingan tertentu, terutama menyuguhkan pemilihan kepala daerah dki. "kami menyesalkan rekayasa gambar, yang bisa dibuat siapa saja, dan berpotensi memperkeruh suasana menjelang pemilihan gubernur ini," tutur budi.

Label

Fakta

Back

©Copyright 2020 | Teknik Informatika
UPN Veteran Yogyakarta

Gambar 4.21 Halaman Detail Data Training

7. Halaman Data Testing

Halaman data *testing* tidak jauh berbeda dengan halaman data *training*. Halaman data *testing* ini akan menampilkan semua data yang menjadi data *testing* saat proses pembuatan model. Halaman ini hanya terdapat fitur detail data *testing* untuk melihat data *testing* secara lengkap. Halaman data *testing* seperti pada Gambar 4.22

No	Konten	Label	Aksi
1	(1) http://bit.ly/2rhtadc / http://bit.ly/2mxvn7s, first draft news: "konten yang dimanipulasi ketika informasi atau gambar yang asli dimanipulasi untuk menipu". post sumber menggunakan video..	Fakta	View
2	"lolongan histeris siksa kubur alexander mendulang hikmah alexander warga non muslim jerman dinyatakan meninggal hingga dimakamkan. di dalam kuburnya alexander ternyata menjerit jerit kesakitan..."	Hoax	View
3	: akun youtube misteri malam dan akun facebook caption hits atau @captionhitsvideoofficial mengunggah video yang mengatakan bahwa thoriq riski ahmad maulidan yang merupakan korban hilang di bukit..	Fakta	View
4	: menyelakan ac saat pertama masuk mobil dengan jendela tertutup akan mengandung 400-800 mg benzene jika parkir di luar ruangan dibawah sinar matahari pada temperatur 15 derajat ke atas, tingkat..	Hoax	View
5	"kalo cucu sunan kaliaga harusnya saya kenal yai bukan sunan kaliaga kayaknya tapi #kalijodo", "makruf amin. jokowi pewaris indonesia krn beliau cucu sunan kali jaga". =====	Hoax	View
6	"ternyata ini alasan dibalik kenapa jokowi nggak mau kasus reklamasi diusut, mungkin uangnya untuk dana mantu anaknya.".	Hoax	View
7	: 1. "maaf sayang cuma ingin tahu, di mana, tapi selama ini pendukungnya tak mampu memberikan jawaban. tanggung jawab ijtimia ulama? prabowo shalat jumat di mana? kami sudah keliling masjid..."	Hoax	View

Gambar 4.22 Halaman Data Testing

Apabila admin melihat data *testing* secara detail maka tampilannya akan seperti pada Gambar 4.23. Detail data *testing* ini berisi konten yang ditampilkan secara lengkap beserta labelnya.

No	Konten	Label	Aksi
1	(1) http://bit.ly/2rhtadc / http://bit.ly/2mxvn7s, first draft news: "konten yang dimanipulasi ketika informasi atau gambar yang asli dimanipulasi untuk menipu". post sumber menggunakan video..	Fakta	
2	"lolongan histeris siksa kubur alexander mendulang hikmah alexander warga non muslim jerman dinyatakan meninggal hingga dimakamkan. di dalam kuburnya alexander ternyata menjerit jerit kesakitan,..	Hoax	
3	: akun youtube misteri malam dan akun facebook caption hits atau @captionhitsvideoofficial mengunggah video yang mengatakan bahwa thoriq riski ahmad Maulidan yang merupakan korban hilang di bukit..	Fakta	
4	: menyelakan ac saat pertama masuk mobil dengan jendela tertutup akan mengandung 400-800 mg benzene jika parkir di luar ruangan dibawah sinar matahari pada temperatur 15 derajat ke atas, tingkat...	Hoax	
5	"kalo cucu sunan kali juga harusnya ska kenal yai bukan sunan kali juga kayaknya tapi #kalijodo". "makruf amin. jokowi pewaris indonesia krn beliau cucu sunan kali juga". =====	Hoax	
6	"ternyata ini alasan dibalik kenapa jokowi nggak mau kasus reklamasi diusut, mungkin uangnya untuk dana mantu anaknya.".	Hoax	
7	: 1. "maaf saya cuma ingin tahu, di mana. tapi selama ini pendukungnya tak mampu memberikan jawaban. tanggung jawab ijtima ulama mana? prabowo shalat jumat di mana? kami sudah keliling masjid..	Hoax	

Gambar 4.23 Halaman Detail Data Testing

8. Halaman Data Pengujian

Halaman data pengujian merupakan halaman yang menampilkan data hasil pengujian. Data ini terdiri dari tanggal melakukan pengujian, total dataset, jumlah data training dan data testing, nilai presisi, recall serta nilai akurasi. Berikut tampilan dari halaman data pengujian pada Gambar 4.24.

No	Tanggal	Total Dataset	Data Training	Data Testing	Presisi	Recall	Akurasi
1	2020-03-09	4366	3493	873	0.8875	0.8925	0.885

Gambar 4.24 Halaman Data Pengujian

9. Halaman Data Model CNN

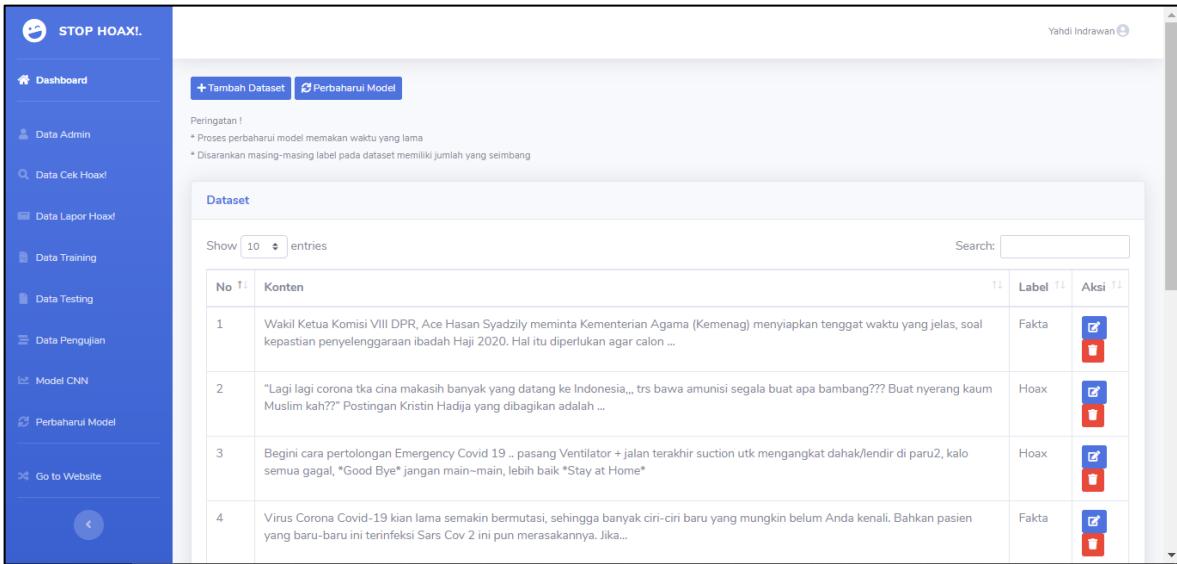
Halaman data moel CNN akan menampilkan grafik dari prosentase nilai presisi, recall dan akurasi. Data tersebut akan divisualisasikan dalam benuk *line chart*. Tampilan dari halaman ini dapat dilihat pada Gambar 4.25



Gambar 4.25 Halaman Data Model CNN

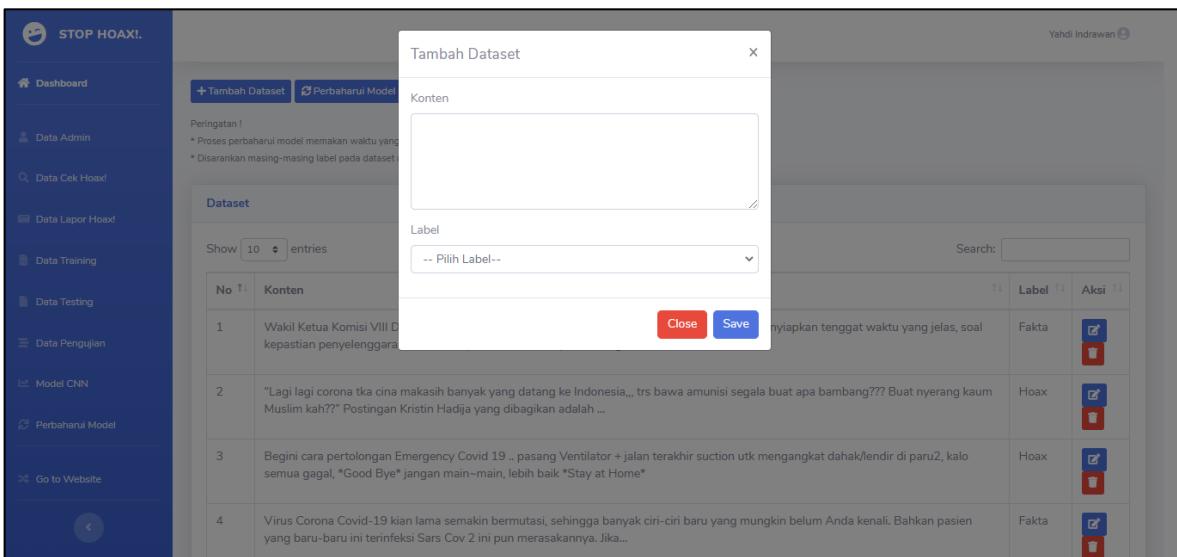
10. Halaman Data Perbaharui Model

Halaman data perbaharui model merupakan halaman yang berfungsi untuk melakukan proses *update model*. Prosesnya dengan melakukan proses *training* dengan model yang telah dibuat menggunakan dataset yang baru. Sehingga pengetahuan dari model menjadi lebih luas dan diharapkan hasil prediksinya menjadi lebih akurat. Halaman dari data perbaharui model dapat dilihat pada Gambar 4.26.



Gambar 4.26 Halaman Data Perbaharui Model

Sebelum melakukan proses perbaharui model, terlebih dahulu menambahkan dataset pada tombol tambah dataset. Dataset ini nantinya akan digunakan sebagai data *training* untuk memperbaharui model. Berikut tampilan tambah dataset pada bagian perbaharui model di Gambar 4.27.



Gambar 4.27 Halaman Tambah Dataset Pada Perbaharui Model

11. Halaman Data Profil

Bagian data profil berada dibagian sebelah kanan atas dengan menekan *icon* akun. Bagian ini terdapat dua pilihan yaitu menu ganti password dan *logout*. Ganti password dapat

dilakukan dengan mengisi form yang terdiri dari password lama, password baru dan konfirmasi password. Halaman ganti password ini seperti pada Gambar 4.28.

Ganti Password

Password Lama

Password Baru

Konfirmasi Password

Close Save

Gambar 4.28 Halaman Ganti Password

Menu lain di bagian profil yaitu *logout*. Menu ini berfungsi untuk keluar dari sistem dan akan kembali ke halaman *login*. Tampilan dari halaman *logout* seperti pada Gambar 4.29.



Gambar 4.29 Halaman *Logout*

4.2 Pengujian

Pada bagian ini akan menjelaskan tentang hasil pengujian yang dilakukan terhadap hasil penelitian yang telah dibuat. Pada penelitian ini melakukan dua pengujian yaitu pengujian terhadap model algoritma CNN dan pengujian terhadap sistem yang telah dibuat. Berikut penjelasan dari masing-masing hasil pengujian.

4.2.1 Pengujian Model

Pengujian model yaitu proses pengujian yang dilakukan untuk mengukur kinerja dari model algoritma CNN yang telah dibuat sebelumnya. Pengujian model yang dilakukan pada

penelitian menggunakan beberapa metode yaitu *k-fold cross validation*, *confusion matrix* dan kurva ROC.

4.2.1.1 K-Fold Cross Validation

K-fold cross validation merupakan salah satu metode validasi yang berfungsi untuk mengukur rata-rata tingkat keberhasilan dari model yang telah dibuat. Proses dari *cross validation* yaitu melakukan pembagian data menjadi n-kelompok. Pada penelitian ini menggunakan 5 fold yang berarti data akan dibagi menjadi 5 kelompok. Selanjutnya akan dilakukan proses *looping* sebanyak sebanyak jumlah kelompok. Lalu akan dilakukan proses *training*, *testing* dan pengujian terhadap model pada setiap iterasi. Untuk iterasi pertama yaitu kelompok pertama menjadi data *testing* dan kelompok lainnya menjadi data *training*. Hasil dari proses *training* dan *testing* akan dilakukan pengujian menggunakan *confusion matrix* untuk mendapatkan nilai TP, FP, TN dan FN yang nantinya akan digunakan untuk mencari nilai akurasi, presisi dan recall.

Setelah melalui proses *confusion matrix* maka akan dilakukan pengujian menggunakan kurva ROC. Metode ini menggunakan nilai-nilai dari *confusion matrix* untuk menghasilkan nilai FPR dan TPR serta nilai AUC. Jadi, setiap iterasi pada *cross validation* akan menghasilkan pengujian *confusion matrix* dan kurva ROC.

4.2.1.2 Confusion Matrix

Confusion matrix merupakan metode pengujian yang dilakukan pada penelitian ini untuk mengukur tingkat performa dari model algoritma CNN. *Confusion matrix* ini akan menghasilkan nilai akurasi, presisi dan recall berdasarkan nilai *True Positive* (TP), *False Positive* (FN), *True Negative* (TN) dan *False Negative* (FN). Karena menggunakan *cross validation* maka menghasilkan 5 tabel *confusion matrix* yang nantinya akan dicari nilai ratanya. Tabel *confusion matrix* yang dihasilkan dapat dilihat pada Tabel 4.1

Tabel 4.1 Tabel *Confusion Matrix*

Fold	Confusion Matrix			
	TP	FN	FP	TN
1	435	65	58	316
2	435	62	57	319
3	420	45	73	335
4	435	68	58	312
5	441	49	52	331

Berdasarkan tabel *confusion matrix* yang dihasilkan maka dapat dihitung nilai akurasi, presisi dan recall dari model algoritma CNN. Berikut hasil pengujian *confusion matrix* pada masing-masing iterasi beserta nilai rata-ratanya dapat dilihat pada Tabel 4.2

Tabel 4.2 Tabel Pengujian *K-Fold Cross Validation*

Fold	Akurasi	Presisi	Recall
1	85,93%	85,75%	85,59%
2	86,37%	86,18%	86,07%
3	86,48%	86,22%	86,68%
4	85,57%	85,40%	85,17%
5	88,43%	88,21%	88,28%
Rata-Rata	86,56%	86,35%	86,36%

4.2.1.3 Kurva ROC

Kurva ROC merupakan metode pengujian yang merepresentasikan hubungan antara nilai sensitivitas dan 1-spesifisitas. Kurva ROC ini akan menggambarkan keakuratan dari hasil prediksi model. Pada pengujian ini memanfaatkan nilai-nilai dari tabel *confusion matrix* untuk menghasilkan nilai FPR dan TPR. Setelah itu, menghitung nilai AUC model dari masing-masing iterasi. Berikut hasil FPR dan TPR model CNN dapat dilihat pada Tabel 4.3

Tabel 4.3 Tabel FPR dan TPR Kurva ROC

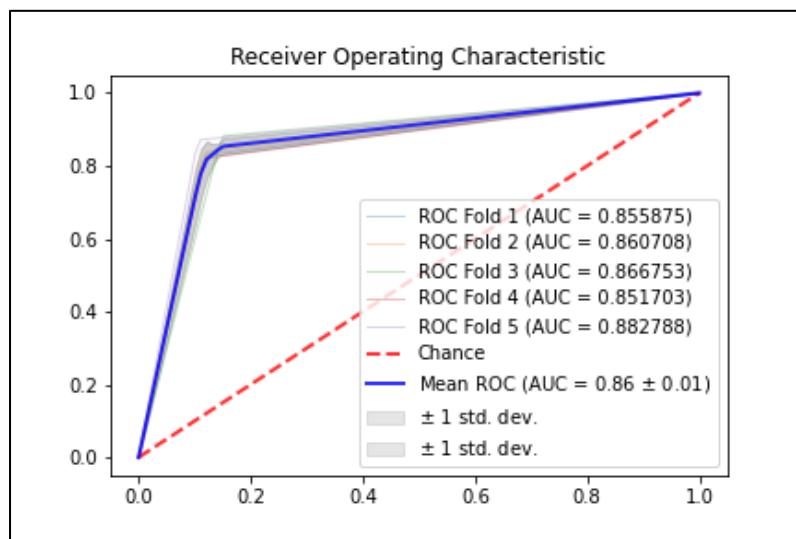
Fold	FPR	TPR
1	0.155	0.870
2	0.152	0.875
3	0.179	0.903
4	0.157	0.865
5	0.136	0.900
Rata-Rata	0.156	0.883

Selanjutnya menghitung nilai AUC yaitu luas daerah dibawah kurva ROC. Nilai AUC dari kurva ROC pada model CNN dapat dilihat pada Tabel 4.4

Tabel 4.4 Tabel AUC Kurva ROC

Fold	AUC
1	0.856
2	0.861
3	0.867
4	0.852
5	0.883
Rata-Rata	0.864

Setelah mendapatkan nilai FPR dan TPR serta nilai AUC maka dapat dilakukan visualisasi dalam bentuk grafik. Grafik dari kurva ROC dapat dilihat pada Gambar 4.30.

**Gambar 4.30** Kurva ROC

4.2.2 Pengujian Sistem

Pada bagian ini akan menjelaskan hasil pengujian terhadap sistem yang telah dibangun. Pengujian sistem ini bertujuan untuk mengukur kinerja dari perangkat lunak yang telah dibangun sehingga *bug* dan sebagainya dapat diketahui dan selanjutnya dapat diperbaiki. Metode pengujian sistem yang akan digunakan yaitu metode *black box*. Metode ini menekankan pada bagian fungsionalitas untuk memastikan setiap fungsi berjalan dengan semestinya. Berikut hasil pengujian sistem menggunakan metode *black box* pada Tabel 4.5

Tabel 4.5 Hasil Pengujian Black Box

Aktor	Halaman	Detail Pengujian	Pengujian	
			Berhasil	Gagal
User	Home	Menampilkan daftar menu	10	
		Menampilkan header	10	
		Fungsi tombol <i>Get Started</i>	10	
	Cek Hoax	Menampilkan daftar menu	10	
		Fungsi cek hoax	10	
		Menampilkan hasil klasifikasi	10	
	Lapor Hoax	Menampilkan daftar menu	10	
		Pengajuan laporan hoax	10	
	Disclaimer	Menampilkan daftar menu	10	
		Menampilkan disclaimer sistem	10	
Admin	Contact	Menampilkan daftar menu	10	
		Menampilkan informasi sistem	10	
		Berpindah ke menu lain via links	10	
	Login	Melakukan proses login	10	
		Fungsi session	10	
	Dashboard	Menampilkan jumlah setiap data	10	
	Data Admin	Menampilkan data admin	10	
		Menambahkan data admin	10	
		Mengedit data admin	10	
		Menghapus data admin	10	
		Fungsi pencarian dan pengurutan	10	
	Data Cek Hoax	Menampilkan data cek hoax	10	
		Melakukan pengecekan hoax	10	
		Menampilkan hasil klasifikasi	10	
		Menampilkan detail cek hoax	10	
		Menghapus data cek hoax	10	
		Fungsi pencarian dan pengurutan	10	
	Data Lapor Hoax	Menampilkan data laporan hoax	10	
		Menambahkan data laporan hoax	10	
		Mengedit data laporan hoax	10	
		Menghapus data laporan hoax	10	
		Fungsi pencarian dan pengurutan	10	
	Data Training	Menampilkan data training	10	
		Menampilkan detail data training	10	
		Fungsi pencarian dan pengurutan	10	
	Data Testing	Menampilkan data testing	10	
		Menampilkan detail data testing	10	
		Fungsi pencarian dan pengurutan	10	
	Data Pengujian	Menampilkan data pengujian	10	
		Fungsi pencarian dan pengurutan	10	
	Model CNN	Menampilkan grafik hasil pengujian	10	
		Mengolah data grafik	10	
	Pembaharui Model	Menampilkan data dataset	10	
		Menambahkan data dataset	10	
		Mengedit data dataset	10	
		Menghapus data dataset	10	
		Melakukan proses pembaharui model	10	
		Fungsi pencarian dan pengurutan	10	
	Profil	Fungsi ganti password	10	
		Fungsi logout	10	
	Go to Website	Berpindah ke halaman user	10	

4.3 Pembahasan

Berdasarkan implementasi dan hasil pengujian yang dilakukan didapatkan hasil bahwa model dari algoritma *convolutional neural network* dapat melakukan klasifikasi berita *hoax* dengan baik. Tingkat performa dari model CNN yang telah dibuat memiliki nilai rata-rata akurasi sebesar 86,56%, presisi sebesar 86,35% dan recall sebesar 86,36%. Hasil tersebut berdasarkan pengujian yang dilakukan menggunakan metode *confusion matrix*. Sedangkan dari proses pengujian menggunakan kurva ROC menghasilkan rata-rata nilai FPR, TPR dan AUC secara urut sebesar 0.156, 0.883, dan 0.864. Hasil tersebut merupakan nilai rata-rata dari 5 iterasi pada proses *cross validation*.

Tingkat performa yang dihasilkan menunjukkan bahwa algoritma *Convolutional Neural Network* dapat melakukan klasifikasi berita *hoax* menggunakan data yang berbentuk teks dengan hasil yang baik. Meskipun biasanya algoritma ini digunakan untuk mengolah data yang berbentuk citra.

BAB V

PENUTUP

5.1 Kesimpulan

Berdasarkan hasil analisis, perancangan dan pembahasan yang dilakukan, dapat diperoleh kesimpulan sebagai berikut:

1. Telah dibangun suatu sistem untuk melakukan klasifikasi berita *hoax* dengan menggunakan metode *Convolutional Neural Network* (CNN).
2. Metode *Convolutional Neural Network* (CNN) dapat mengenali pola-pola berita *hoax* dan melakukan klasifikasi dengan hasil pengujian menunjukkan rata-rata tingkat akurasi sebesar sebesar 86,56%, presisi sebesar 86,35% dan recall sebesar 86,36%. Penelitian ini menggunakan 4349 data yang dibagi menjadi data *training* dan *testing* menggunakan metode *k-fold cross validation* dengan $k=5$.
3. Meskipun tujuan dibuatnya metode *Convolutional Neural Network* (CNN) untuk mengolah data berbentuk dua dimensi (citra) namun metode CNN juga dapat digunakan untuk mengklasifikasikan data berbentuk teks dengan hasil yang baik juga. Hal tersebut ditunjukkan pada pengujian kurva ROC yang menghasilkan nilai AUC sebesar 0.864. Hasil AUC ini berada pada nilai antara 0.80 – 0.90 sehingga masuk ke dalam kategori *good classification*.

5.2 Saran

Adapun saran yang dapat digunakan untuk pengembangan penelitian lebih lanjut adalah sebagai berikut:

1. Membuat pengambilan dataset melalui *web scraping* menjadi otomatis sehingga dapat melakukan proses *upgrade* model secara berkala. Hal ini karena masalah berita *hoax* sangat bergantung pada konten berita yang *up to date*.

2. Dapat menambahkan klasifikasi pada kategori berita. Misalnya berita dibagi menjadi kategori ekonomi, politik, sosial dan sebagainya. Jadi, hasilnya dapat diketahui berita tersebut termasuk berita *hoax* atau fakta serta termasuk pada kategori ekonomi, politik atau sosial.
3. Dapat membandingkan dengan metode *deep learning* lainnya sehingga dapat diketahui metode *deep learning* mana yang lebih optimal dalam menangani masalah klasifikasi berita *hoax*.

DAFTAR PUSTAKA

- Abdullah, A. H. (2018). Word Embedding. Retrieved February 29, 2020, from <https://rpubs.com/> website: <https://rpubs.com/ahmadhusain/wordembedding>
- 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*.
- Berry, M. W., & Kogan, J. (2010). *Text Mining : Applications and Theory*. United Kingdom: WILEY.
- 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>
- Goller. (2000). Automatic Document Classification: A Thorough Evaluation of Various Methods. *Proceedings of International Symposium on Information Theory and Its Application*, 145–162.
- 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 Klasifikasit Analysis. *Procedia Computer Science*, (17), 26–32.
- Hashimi, H., Hafez, A., & Mathkour, H. (2015). Selection criteria for text mining approaches. *College of Computer and Information Sciences*.
- Hughes, M., Li, I., Kotoulas, S., & Suzumura, T. (2017). *Medical Text Classification using Convolutional Neural Networks*.
- Kotler, P., & Keller, K. L. (2012). *Marketing Management* (14th ed.). United State of America: Pearson.
- 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.
- Lina, Q. (2019). Apa itu Convolutional Neural Network ? Retrieved June 25, 2020, from <https://medium.com/> website: <https://medium.com/@16611110/apa-itu-convolutional-neural-network-836f70b193a4>
- MacDougall, C. D. (1958). *Hoaxes*. New York: Dover Publication.
- 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.
- Nasution, M. A. (2017). HOAX SEBAGAI BENTUK HUDUD MENURUT HUKUM ISLAM. *Yurisprudentia*, 3(1).
- Nazief, B., & Adriani, M. (1996). *Confix Stripping: Approach to Stemming Algorithm for Bahasa Indonesia*.
- Nikhit, K. (2015). A Web Scraping Approach in Node.js. *International Journal of Science, Engineering and Technology Research*, 909–912.
- Pereira, R. C., & Vanitha, T. (2015). Web Scraping of Social Networks. *International Journal of Innovative Research in Computer and Communication Engineering*, 3(7), 237–240.
- Pressman, R. S. (2010). *Software Engineering : A Practitioner's Approach* (7th ed.). McGraw-Hill Education.
- Primas, N. (2017). *Deteksi Cacat Permukaan Buah Manggis Berbasis Pengolahan Citra Menggunakan Metode Klasifikasi Support Vector Machine (SVM)*. Universitas Muhammadiyah Yogyakarta.
- Purnomo, D. (2017). Model Prototyping Pada Pengembangan Sistem Informasi. *JIMP - Jurnal Informatika Merdeka Pasuruan*, 2(2), 54–61.
- 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.muda.h.percaya.berita>
- Rini, D. P. (2012). *Metodologi Pengembangan Sistem*. Yogyakarta: Graha Ilmu.
- 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 Pendekripsi 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.
- 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/](http://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). Klasifikasi 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>

LAMPIRAN

