



**ITS**

Institut  
Teknologi  
Sepuluh Nopember

TUGAS AKHIR - IS184853

**ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA  
INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (STUDI KASUS : POLITIK)**

***ASPECT LEVEL SENTIMENTS ANALYSIS OF  
INDONESIAN TEXT ON SOCIAL MEDIA USING  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (CASE STUDY: POLITICS)***

MUHAMMAD WILDAN MAULIDANI  
NRP 0521164000029

Dosen Pembimbing :  
Renny Pradina K., S.T., M.T., SCJP

DEPARTEMEN SISTEM INFORMASI  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**TUGAS AKHIR - IS184853**

**ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA  
INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (STUDI KASUS : POLITIK)**

**MUHAMMAD WILDAN MAULIDANI**  
NRP 0521164000029

**Dosen Pembimbing :**  
Renny Pradina K., S.T., M.T., SCJP

**DEPARTEMEN SISTEM INFORMASI**  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020



**ITS**  
Institut  
Teknologi  
Sepuluh Nopember

**FINAL PROJECT - IS184853**

**ASPECT LEVEL SENTIMENTS ANALYSIS OF  
INDONESIAN TEXT ON SOCIAL MEDIA USING  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (CASE STUDY: POLITICS)**

**MUHAMMAD WILDAN MAULIDANI**  
NRP 0521164000029

Supervisor :  
Renny Pradina K., S.T., M.T., SCJP

DEPARTMENT OF INFORMATION SYSTEMS  
Faculty of Intelligent Electrical and Informatics Technology  
Institut Teknologi Sepuluh Nopember  
Surabaya 2020



**LEMBAR PENGESAHAN**

**ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA  
INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (STUDI KASUS : POLITIK)**

**TUGAS AKHIR**

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh:

**MUHAMMAD WILDAN MAULIDANI**  
**NRP. 0521164000029**

Surabaya, 21 Januari 2020

**KEPALA  
DEPARTEMEN SISTEM INFORMASI**



**Dr. Mudjahidin, S.T., M.T.**  
**NIP. 19701010 200312 1 001**

## LEMBAR PERSETUJUAN

### ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN HIERARCHICAL ATTENTION POSITION-AWARE NETWORK (STUDI KASUS : POLITIK)

#### TUGAS AKHIR

Disusun Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada

Departemen Sistem Informasi  
Fakultas Teknologi Elektro dan Informatika Cerdas  
Institut Teknologi Sepuluh Nopember

Oleh :

**MUHAMMAD WILDAN MAULIDANI**

**NRP. 05211640000029**

Disetujui Tim Penguji : Tanggal Ujian : 08 Januari 2020  
Periode Wisuda : Maret 2020

**Renny Pradina K., S.T., M.T., SCJP**

(Pembimbing I)

**Nur Aini Rakhmawati, S.Kom., M.Sc.Eng,  
Ph.D**

(Penguji I)

**Faizal Johan Atletiko, S.Kom., M.T**

(Penguji II)



**ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA  
INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (STUDI KASUS : POLITIK)**

**Nama Mahasiswa : Muhammad Wildan Maulidani**  
**NRP : 0521164000029**  
**Jurusan : Sistem Informasi FTEIC-ITS**  
**Pembimbing 1 : Renny Pradina K., S.T., M.T., SCJP**

**ABSTRAK**

*Jumlah pengguna internet penduduk indonesia dari tahun ke tahun semakin meningkat. Penggunaan internet ini sebagian besar digunakan untuk berinteraksi di media sosial. Pengguna media sosial seringkali mengungkapkan pikiran atau pandangannya di media sosial salah satunya dibidang politik. Khususnya pada tahun 2019 ini dimana terjadi pesta demokrasi di Indonesia yaitu diadakannya pemilu . Hal ini membuat banyak bermunculan opini-opini tentang objek politik. Untuk mengetahui pandangan atau penilaian masyarakat terhadap suatu objek politik dapat digunakan analisis sentimen. Umumnya terdapat dua macam sentimen atau opini yang ditulis yaitu sentimen positif dan sentimen negatif. Tetapi perlu diperhatikan juga bahwa dalam satu kalimat opini dapat mengandung berbagai sentimen. Hal ini dikarenakan suatu objek politik dapat memiliki berbagai aspek yang juga diberikan pandangan sendiri oleh penulis. Banyaknya jumlah opini yang tidak sesuai dengan kaidah standar akan membuat penarikan kesimpulan sentimen level aspek dari objek menjadi sulit. Sehingga diperlukan sebuah sistem yang mampu melakukan klasifikasi sentimen level aspek terhadap objek. Klasifikasi sentimen level aspek dilakukan dengan menggunakan metode deep-learning yaitu Recurrent Neural Network dan juga berdasarkan attention. Penelitian sebelumnya melakukan percobaan dengan menggunakan metode Hierarchical Attention*

*Position-aware network (HAPN) yang berdasarkan pada Bi-GRU mencapai hasil yang sangat baik. Teks yang digunakan sebagai input pelatihan model dan diubah menjadi representasi vector menggunakan metode Word Embedding. Kemudian dilakukan evaluasi performa dari model HAPN. Hasil dari penelitian tugas akhir ini adalah didapatkannya hasil klasifikasi sentimen level aspek pada objek politik yang lebih baik.*

***Kata kunci: Analisis Sentimen Level Aspek, Politik, Media Sosial, Hierarchical Attention Position-aware Network***



**ASPECT LEVEL SENTIMENTS ANALYSIS OF  
INDONESIAN TEXT ON SOCIAL MEDIA USING  
HIERARCHICAL ATTENTION POSITION-AWARE  
NETWORK (CASE STUDY: POLITICS)**

**Student Name** : Muhammad Wildan Maulidani  
**NRP** : 0521164000029  
**Department** : Information Systems ELECTICS-ITS  
**Supervisor 1** : Renny Pradina K., S.T., M.T., SCJP

**ABSTRACT**

*The number of Indonesian internet users has increased from year to year. Internet is mostly used to interact on social media. Social media users often express their thoughts or views in social media where one of them is political field. Especially in 2019 where a democratic party takes place in Indonesia, namely the holding of elections. This makes a lot of opinions on political objects emerge. To find out the views or assessments of the public about a political object, sentiment analysis can be used. Generally there are two kinds of sentiments or opinions written, namely positive sentiment and negative sentiment. But it should also be noted that in one sentence opinion can contain various sentiments. This is because a political object can have various aspects which are also given their own views by the author. The large number of opinions that are not in accordance with the standard rules will make the conclusion of the sentiment level aspect of the object becomes difficult. So we need a system that capable of classifying sentiment level aspects of objects. Aspect level sentiment classification is done by using deep-learning methods namely Recurrent Neural Network and also based on attention. Previous studies conducted experiments using the Hierarchical Attention Position-aware network (HAPN) method based on Bi-GRU achieved very good results. The text is used as an input training model and is converted to a vector representation using the Word Embedding method. Then evaluate*

*the performance of the HAPN model. The results of this thesis research is to obtain better results from the aspect level sentiment classification on political objects.*

***Keywords: Aspect Level Sentiment Analysis, Politics, Social Media, Hierarchical Attention Position-aware Network***

## SURAT PERNYATAAN BEBAS PLAGIARISME

Saya yang bertandatangan di bawah ini:

Nama : Muhammad Wildan Maulidani  
NRP : 0521164000029  
Tempat/Tanggal lahir : Gresik / 08 Juli 1998  
Fakultas/Departemen : FTEIC / Sistem Informasi  
Nomor Telp/Hp/email : 083832355442 / maulidaniw@gmail.com

Dengan ini menyatakan dengan sesungguhnya bahwa penelitian/makalah/tugas akhir saya yang berjudul

ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA INDONESIA  
PADA MEDIA SOSIAL MENGGUNAKAN HIERARCHICAL  
ATTENTION POSITION-AWARE NETWORK (STUDI KASUS :  
POLITIK)

### Bebas Dari Plagiarisme Dan Bukan Hasil Karya Orang Lain.

Apabila dikemudian hari ditemukan seluruh atau sebagian penelitian/makalah/tugas akhir tersebut terdapat indikasi plagiarisme, maka saya bersedia menerima sanksi sesuai peraturan dan ketentuan yang berlaku.

Demikian surat pernyataan ini saya buat dengan sesungguhnya dan untuk dipergunakan sebagaimana mestinya.

Surabaya, 14 Januari 2020



Muhammad Wildan Maulidani

*Halaman ini sengaja dikosongkan*

## KATA PENGANTAR

Puji dan syukur penulis tuturkan ke hadirat Allah SWT, Tuhan Semesta Alam yang telah memberikan kekuatan dan hidayah-Nya kepada penulis sehingga penulis mendapatkan kelancaran dalam menyelesaikan tugas akhir ini dengan judul ANALISIS SENTIMEN LEVEL ASPEK TEKS BAHASA INDONESIA PADA MEDIA SOSIAL MENGGUNAKAN HIERARCHICAL ATTENTION POSITION-AWARE NETWORK (STUDI KASUS : POLITIK) yang merupakan salah satu syarat kelulusan pada Departemen Sistem Informasi, Fakultas Teknologi Informasi, Institut Teknologi Sepuluh Nopember Surabaya.

Terima kasih penulis sampaikan kepada pihak-pihak yang telah mendukung, memberikan saran, motivasi, semangat, dan bantuan baik berupa materiil maupun moril demi tercapainya tujuan pembuatan tugas akhir ini. Tugas akhir ini tidak akan pernah terwujud tanpa bantuan dan dukungan dari berbagai pihak yang sudah meluangkan waktu, tenaga dan pikirannya. Secara khusus penulis akan menyampaikan ucapan terima kasih yang sebanyak-banyaknya kepada:

1. Bapak Asfa Harir dan Ibu Nadhifah selaku kedua orang tua dari penulis serta Mas Yayak, Mbak Pipit, Mbak Leli, Mbak Wiwin, Mbak Feni dan Mbak Maya selaku saudara kandung dari penulis yang tiada henti memberikan dukungan dan semangat
2. Bapak Dr. Mudjahidin, S.T., M.T. selaku Ketua Departemen Sistem Informasi ITS Surabaya.
3. Ibu Renny Pradina K., S.T., M.T., SCJP, selaku dosen pembimbing dan sebagai narasumber yang senantiasa meluangkan waktu, memberikan ilmu dan petunjuk, serta memotivasi untuk kelancaran tugas akhir.
4. Ibu Nur Aini Rakhmawati, S.Kom, M.Sc.Eng, Ph.D, Ibu Irmasari Hafidz, S.Kom., M.Sc. dan Bapak Faizal Johan

Atletiko, S.Kom., M.T selaku dosen penguji yang telah memberikan saran dan kritik untuk perbaikan tugas akhir.

5. Bapak Edwin Riksakomara, S.Kom., M.T selaku dosen wali penulis yang telah membimbing penulis mulai dari awal perkuliahan.
6. Seluruh dosen dan karyawan Departemen Sistem Informasi ITS yang telah memberikan ilmu yang bermanfaat kepada penulis.
7. Evan selaku teman se-per dosen pembimbing-an yang telah membantu penulis dalam mengerjakan tugas akhir ini.
8. Teman-teman seperjuangan dalam Data Science yaitu Muslikh, Krisna, Yoga, Fafa dan Berta yang membantu penulis dalam mempelajari Data Science yang sangat berguna dalam pengerjaan tugas akhir ini.
9. Teman-teman Departemen RTA HMSI terutama Ubai, Ayun, Pep dan Reza serta rekan-rekan ARTEMIS yang telah memberikan banyak kenangan saat kuliah.
10. Berbagai pihak yang tidak bisa disebutkan satu persatu yang telah turut serta menyukseskan penulis dalam menyelesaikan tugas akhir.

Penyusunan laporan ini masih jauh dari kata sempurna sehingga penulis menerima adanya kritik maupun saran yang membangun untuk perbaikan di masa yang akan datang. Semoga buku tugas akhir ini dapat memberikan manfaat bagi pembaca.

Surabaya, 08 Januari 2020

Penulis

## DAFTAR ISI

LEMBAR PENGESAHAN.....	v
LEMBAR PERSETUJUAN.....	vi
ABSTRAK .....	vii
ABSTRACT .....	ix
SURAT PERNYATAAN BEBAS PLAGIARISME .....	xi
KATA PENGANTAR.....	xiii
DAFTAR ISI.....	xv
DAFTAR GAMBAR .....	xxi
DAFTAR TABEL .....	xxiii
DAFTAR KODE.....	xxvii
BAB I PENDAHULUAN .....	1
1.1. Latar Belakang .....	1
1.2. Rumusan Masalah .....	4
1.3. Batasan Permasalahan .....	4
1.4. Tujuan Penelitian.....	5
1.5. Manfaat Penelitian.....	5
1.6. Relevansi .....	5
BAB II TINJAUAN PUSTAKA .....	7
2.1. Penelitian Sebelumnya .....	7
2.2. Landasan Teori.....	10
2.2.1. <i>Natural Language Processing</i> .....	10
2.2.2. Analisis Sentimen.....	11
2.2.3. Analisis Sentimen Level Aspek.....	12
2.2.4. <i>Deep Learning</i> .....	12
2.2.5. <i>Recurrent Neural Network (CNN)</i> .....	14
2.2.6. <i>Gated Recurrent Unit</i> .....	15
2.2.7. <i>Attention Mechanisme</i> .....	17
2.2.8. <i>Word Embedding</i> .....	18

2.2.9.	<i>Word2Vec</i> .....	19
2.2.10.	<i>Hierarchical Attention Position-aware Network</i> .....	19
2.2.11.	<i>Character Embedding</i> .....	24
2.2.12.	Ukuran Evaluasi Klasifikasi Sentimen .....	24
2.2.13.	Media Sosial <i>Twitter</i> .....	25
2.2.14.	Politik .....	26
BAB III METODOLOGI PENELITIAN .....		27
3.1.	Tahapan Pelaksanaan Penelitian Tugas Akhir.....	27
3.1.1.	Studi Literatur.....	28
3.1.2.	Pengumpulan Data.....	29
3.1.3.	Pra-Pemrosesan Data .....	30
3.1.4.	Pelatihan Data.....	36
3.1.4.1	Implementasi Model Word Embedding.....	36
3.1.4.2	Pembuatan dan Pelatihan Model HAPN.....	37
3.1.5.	Analisis dan Evaluasi Model .....	38
3.1.6.	Dokumentasi.....	38
BAB IV PERANCANGAN.....		39
4.1.	Akuisisi Data Media Sosial .....	39
4.2.	Perancangan <i>Crawler</i> .....	40
4.3.	Desain <i>Database</i> .....	40
4.4.	Perancangan Pra-Pemrosesan Data.....	42
4.4.1.	Perancangan Penggabungan Dataset .....	42
4.4.2.	Perancangan Penghapusan Data yang Terduplikasi .....	42
4.4.3.	Perancangan <i>Filtering</i> Bahasa Indonesia.....	42
4.4.4.	Perancangan Pembersihan dataset .....	42



4.5.	Perancangan Pemberian Label Data.....	43
4.6.	Perancangan Dataset Model HAPN .....	45
4.7.	Perancangan Implentasi Model <i>Word Embedding</i> .....	45
4.8.	Perancangan Model <i>Hierararchical Attention Position-aware Network</i> .....	45
4.9.	Perancangan Evaluasi Pengukuran.....	47
BAB V IMPLEMENTASI .....		51
5.1.	Persiapan Implementasi.....	51
5.2.	Pembuatan <i>Crawler Twitter</i> .....	52
5.3.	Pra-Pemrosesan Dataset .....	55
5.3.1.	Penggabungan Dataset .....	55
5.3.2.	Penghapusan Data yang Terduplikasi .....	56
5.3.3.	<i>Filtering</i> Bahasa .....	56
5.3.4.	Pembuatan Pembersihan Dataset.....	58
5.4.	Pembuatan Pelabelan Data.....	59
5.5.	Pembuatan Model <i>Hierarchical Attention Position Aware Network</i> .....	60
5.6.	Proses Pelatihan Model HAPN .....	68
5.7.	Pengujian Performa Model HAPN.....	72
BAB VI HASIL DAN PEMBAHASAN.....		77
6.1.	Hasil Data <i>Crawling</i> .....	77
6.2.	Hasil Penghapusan Duplikasi Dataset.....	78
6.3.	<i>Filtering</i> Bahasa .....	79
6.4.	Hasil Pembersihan Dataset.....	80
6.5.	Hasil Pelabelan Data .....	81
6.6.	Hasil Implementasi Model <i>Word Embedding</i> .....	83
6.7.	Hasil Pengujian Data.....	84

6.7.1	Konfigurasi Parameter Awal .....	84
6.7.2	Hasil Percobaan Subtask A.....	84
6.7.2.1	Pengaruh Model <i>Neural Network</i> .....	85
6.7.2.2	Pengaruh <i>Optimizer</i> .....	88
6.7.2.3	Pengaruh <i>Learning Rate</i> .....	91
6.7.2.4	Pengaruh <i>Dropout</i> .....	93
6.7.2.5	Pengaruh <i>Character Embedding</i> .....	95
6.7.2.6	Pembahasan Hasil Subtask A .....	96
6.7.3	Hasil Percobaan Subtask B.....	99
6.7.3.1	Pengaruh Model <i>Neural Network</i> .....	99
6.7.3.2	Pengaruh <i>Optimizer</i> .....	102
6.7.3.3	Pengaruh <i>Learning Rate</i> .....	104
6.7.3.4	Pengaruh <i>Dropout</i> .....	106
6.7.3.5	Pengaruh <i>Character Embedding</i> .....	109
6.7.3.6	Pembahasan Hasil Subtask B .....	111
6.7.4	Analisis Antar Subtask .....	113
6.7.5	Uji Signifikansi.....	114
6.7.5.1	Uji Signifikansi Subtask A .....	114
6.7.5.2	Uji Signifikansi Subtask B .....	115
BAB VII	KESIMPULAN DAN SARAN .....	117
7.1.	Kesimpulan.....	117
7.2.	Saran.....	118
DAFTAR PUSTAKA.....		121
BIODATA PENULIS.....		125
LAMPIRAN A DATA BERLABEL.....		127
LAMPIRAN B HASIL EVALUASI MODEL TERBAIK SUBTASK A .....		131

LAMPIRAN C HASIL EVALUASI MODEL TERBAIK  
SUBTASK B.....137

*Halaman ini sengaja dikosongkan*

## DAFTAR GAMBAR

Gambar 2.1 <i>Feedforward Neural Network</i> .....	13
Gambar 2.2 <i>Recurrent Neural Network</i> .....	14
Gambar 2.3 <i>Bidirectional Recurrent Neural Network</i> .....	15
Gambar 2.4 Struktur model berbasis GRU.....	16
Gambar 2.5 Struktur memory GRU .....	16
Gambar 2.6 <i>Attention Mechanisme dalam bidirectional rnn.</i> .....	18
Gambar 2.7 <i>Arsitektur Hierarchical Attention Position-aware Network.</i> .....	20
Gambar 3.1 Alur Pengerjaan Tugas Akhir .....	27
Gambar 3.2 <i>Arsitektur Penelitian Tugas Akhir</i> .....	28
Gambar 3.3 <i>Proses Pra-Pemrosesan Data</i> .....	31
Gambar 3.4 <i>Filtering Bahasa dan Topik Politik</i> .....	32
Gambar 3.5 Alur Proses Model HAPN .....	37
Gambar 5.1 Aplikasi Pelabelan Data .....	60
Gambar 6.1 Performa Model Terbaik <i>Neural Network</i> Subtask A .....	87
Gambar 6.2 Perbandingan Akurasi pada Percobaan Model <i>Neural Network</i> Subtask A.....	88
Gambar 6.3 Perbandingan Akurasi pada Percobaan <i>Optimizer</i> Subtask A .....	90
Gambar 6.4 Perbandingan Akurasi pada Percobaan <i>Learning Rate</i> Subtask A .....	92
Gambar 6.5 Perbandingan Akurasi pada Percobaan <i>Dropout</i> Subtask A .....	94
Gambar 6.6 Perbandingan Akurasi pada Percobaan <i>Character Embedding</i> Subtask A .....	96
Gambar 6.7 <i>Confusion Matrix</i> Model Terbaik Subtask A.....	98
Gambar 6.8 Performa Model Terbaik <i>Neural Network</i> Subtask B .....	101
Gambar 6.9 Perbandingan Akurasi pada Percobaan Model <i>Neural Network</i> Subtask B.....	101
Gambar 6.10 Perbandingan Akurasi pada Percobaan <i>Optimizer</i> Subtask B.....	103

Gambar 6.11 Perbandingan Akurasi pada Percobaan <i>Learning Rate</i> Subtask B.....	105
Gambar 6.12 Performa <i>Dropout</i> Terbaik Subtask B .....	107
Gambar 6.13 Perbandingan Akurasi pada Percobaan <i>Dropout</i> Subtask B.....	108
Gambar 6.14 Perbandingan Akurasi pada Percobaan <i>Character Embedding</i> Subtask B.....	110
Gambar 6.15 <i>Confusion Matrix</i> Model Terbaik Subtask B.....	112

## DAFTAR TABEL

Tabel 2.1 Studi Sebelumnya.....	7
Tabel 3.1 Daftar Entitas untuk kata kunci dengan topik politik ..	29
Tabel 3.2 Contoh Penghapusan Tanda Baca, Link dan Simbol ..	33
Tabel 3.3 Contoh Proses Casefolding .....	34
Tabel 3.4 Contoh Pelabelan Data .....	35
Tabel 4.1 Kata Kunci untuk <i>Crawling</i> .....	39
Tabel 4.2 Desain Database Crawler Post Twitter.....	41
Tabel 4.3 Desain Database Sentimen .....	41
Tabel 4.4 Label Aspek.....	44
Tabel 4.5 Label Sentimen.....	44
Tabel 4.6 Perlakuan Label.....	45
Tabel 4.7 Parameter yang digunakan dalam Model HAPN .....	46
Tabel 5.1 Spesifikasi Perangkat Keras Laptop.....	51
Tabel 5.2. Daftar Library.....	51
Tabel 6.1 Hasil <i>Crawling Tweet</i> .....	77
Tabel 6.2 Hasil Duplikasi Dataset .....	78
Tabel 6.3 Hasil Seleksi Bahasa Indonesia .....	79
Tabel 6.4 Hasil Pembersihan Dataset.....	80
Tabel 6.5 Pembagian Data Berdasarkan Topik.....	81
Tabel 6.6 Penyebaran Data Hasil Final Pelabelan.....	82
Tabel 6.7 Hasil Pelabelan Aspek dan Sentimen.....	83
Tabel 6.8 Jumlah Kata Terdeteksi dari Model <i>Word Embedding</i> <i>Word2vec</i> .....	84
Tabel 6.9 Konfigurasi Awal Model HAPN .....	84
Tabel 6.10 Penyebaran Data pada Subtask A.....	85
Tabel 6.11 Penyebaran Label Pada tiap Data Subtask A .....	85
Tabel 6.12 Hasil Akurasi Percobaan Model <i>Neural Network</i> Pada Data <i>Train</i> Subtask A.....	86
Tabel 6.13 Hasil Evaluasi Pengukuran Percobaan Model <i>Neural</i> <i>Network</i> Pada Data <i>Test</i> Subtask A.....	86
Tabel 6.14 Hasil Akurasi Percobaan <i>Optimizer</i> Pada Data <i>Train</i> Subtask A .....	89
Tabel 6.15 Hasil Evaluasi Pengukuran Percobaan <i>Optimizer</i> Pada Data <i>Test</i> Subtask A .....	90

Tabel 6.16 Hasil Akurasi Percobaan <i>Learning Rate</i> Pada Data <i>Train</i> Subtask A.....	91
Tabel 6.17 Hasil Evaluasi Pengukuran Percobaan <i>Learning Rate</i> Pada Data <i>Test</i> Subtask A.....	92
Tabel 6.18 Hasil Akurasi Percobaan <i>Dropout</i> Pada Data <i>Train</i> Subtask A.....	93
Tabel 6.19 Hasil Evaluasi Pengukuran Percobaan <i>Dropout</i> Pada Data <i>Test</i> Subtask A .....	94
Tabel 6.20 Hasil Akurasi Percobaan <i>Character Embedding</i> Pada Data <i>Train</i> Subtask A .....	95
Tabel 6.21 Hasil Evaluasi Pengukuran Percobaan <i>Character Embedding Test</i> Subtask A.....	96
Tabel 6.22 Pengukuran Evaluasi Model Terbaik dari Subtask A	97
Tabel 6.23 Perhitungan <i>Recall</i> dan <i>F1-Score</i> per Label pada Model Terbaik Subtask A .....	98
Tabel 6.24 Penyebaran Data pada Subtask B .....	99
Tabel 6.25 Penyebaran Label Pada tiap Data Subtask B.....	99
Tabel 6.26 Hasil Akurasi Percobaan <i>Model Neural Network</i> Pada Data <i>Train</i> Subtask B .....	100
Tabel 6.27 Hasil Evaluasi Pengukuran Percobaan Model <i>Neural Network</i> Pada Data <i>Test</i> Subtask B .....	100
Tabel 6.28 Hasil Akurasi Percobaan <i>Optimizer</i> Pada Data <i>Train</i> Subtask B.....	102
Tabel 6.29 Hasil Evaluasi Pengukuran Percobaan <i>Optimizer Pada Data Test</i> Subtask B .....	103
Tabel 6.30 Hasil Akurasi Percobaan <i>Learning Rate</i> Pada Data <i>Train</i> Subtask A.....	104
Tabel 6.31 Hasil Evaluasi Pengukuran Percobaan <i>Learning Rate</i> Pada Data <i>Test</i> Subtask B.....	105
Tabel 6.32 Hasil Akurasi Percobaan <i>Dropout</i> Pada Data <i>Train</i> Subtask B.....	106
Tabel 6.33 Hasil Evaluasi Pengukuran Percobaan <i>Dropout</i> Pada Data <i>Test</i> Subtask B.....	107
Tabel 6.34 Hasil Akurasi Percobaan <i>Character Embedding</i> Pada Data <i>Train</i> Subtask B .....	109



Tabel 6.35 Hasil Evaluasi Pengukuran Percobaan <i>Character Embedding Test Subtask B</i> .....	110
Tabel 6.36 Pengukuran Evaluasi Model Terbaik dari Subtask B .....	111
Tabel 5.37 Perhitungan Recall dan F1-Score per Label pada Model Terbaik Subtask B .....	112
Tabel 6.38 Perbandingan Konfigurasi Model Terbaik .....	113
Tabel 6.39 Hasil Evaluasi Pengukuran.....	113
Tabel 6.40 Data Percobaan Uji Signifikansi Subtask A.....	114
Tabel 6.41 Data Percobaan Uji Signifikansi Subtask B .....	115

*Halaman ini sengaja dikosongkan*

## DAFTAR KODE

Kode 5.1 Potongan Kode Implementasi API untuk Crawling.....	52
Kode 5.2 Potongan Kode Parameter Proses Crawling .....	53
Kode 5.3 Potongan Kode Perulangan Crawling.....	54
Kode 5.4 Potongan Kode untuk Pnggabungan Dataset.....	55
Kode 5.5 Potongan Kode untuk Menghapus Kata Duplikat .....	56
Kode 5.6 Potongan Kode Untuk Mendeteksi Bahasa .....	57
Kode 5.7 Potongan Kode Proses Pembersihan Dataset.....	58
Kode 5.8 Potongan Kode Pembuatan <i>Position Embedding</i> .....	61
Kode 5.9 Potongan Kode Implementasi <i>Word Embedding</i> .....	62
Kode 5.10 Potongan Kode Pembuatan <i>Character Embedding</i> ...	65
Kode 5.11 Potongan Kode Pembuatan Model HAPN.....	67
Kode 5.12 Potongan Kode Inisiasi Skenario dan Pembagian Data .....	69
Kode 5.13 Potongan Kode Pembuatan Fungsi <i>Training Model</i> .	70
Kode 5.14 Potongan Kode Penjalanan Skenario Pelatihan Model .....	71
Kode 5.15 Potongan Kode Pengambilan Hasil Prediksi .....	72
Kode 5.16 Potongan Kode Class Untuk Menghitung Metriks Evaluasi .....	73
Kode 5.17 Potongan Kode Prediksi Data dan Penghitungan Metriks Evaluasi.....	74

*Halaman ini sengaja dikosongkan*

# **BAB I**

## **PENDAHULUAN**

Pada bab pendahuluan ini akan diuraikan proses identifikasi masalah penelitian yang meliputi latar belakang masalah, perumusan masalah, batasan masalah, tujuan tugas akhir, manfaat kegiatan tugas akhir, manfaat kegiatan tugas akhir dan relevansi terhadap pengerjaan tugas akhir. Berdasarkan uraian pada bab ini, harapannya gambaran umum permasalahan dan pemecahan masalah pada tugas akhir dapat dipahami.

### **1.1. Latar Belakang**

Jumlah pengguna internet dari tahun ke tahun meningkat dengan cepat [1]. Hal ini berdasarkan pada hasil survei dari Asosiasi Penyelenggara Jasa Internet Indonesia (APJII), dimana hingga pada tahun 2017 terdapat 143 juta dari 262 juta orang penduduk Indonesia yang menggunakan internet [1]. Jumlah tersebut merupakan 54,68% dari total populasi penduduk Indonesia. Hal ini menunjukkan kenaikan angka yang cukup besar dibandingkan pada tahun 2015 yaitu sebesar 110,2 juta[1]. Layanan sosial media menjadi salah satu layanan yang sering diakses oleh pengguna internet Indonesia, yaitu hingga mencapai 87,13% [1].

Banyaknya pengguna internet di Indonesia ini dapat memberikan pengaruh dalam berbagai aspek. Salah satunya yaitu media sosial yang digunakan sebagai sarana komunikasi memiliki peran membawa penggunaannya untuk membagikan dan memberi feedback secara terbuka terhadap berbagai hal[2]. Media sosial seperti *facebook* dan *twitter* pada awalnya cenderung berkaitan dengan pertemanan, namun saat ini mulai banyak digunakan untuk menyinggung ranah politik dan pemerintahan [2]. Media sosial dijadikan sebagai strategi komunikasi politik merupakan hal relatif baru dan menjadi fenomena hangat hingga kini. Banyak tokoh politik dan partai politik memanfaatkan media sosial untuk melakukan interaksi dengan publik sekaligus memasarkan gagasan yang dimiliki. Berdasarkan media berita suara.com sebanyak 93.75% atau 15

dari 16 partai politik peserta pemilu 2019 menggunakan media sosial sebagai alat untuk menyebarluaskan materi kampanye mereka kepada publik [3]. Dimana mayoritas partai menggunakan empat platform: *Facebook*, *Twitter*, *Instagram*, dan *YouTube* [3]. *Twitter* merupakan media sosial terbanyak yang digunakan oleh 15 partai politik diikuti Instagram dengan 13 partai politik. Media sosial diyakini mampu meningkatkan elektabilitas calon pemimpin atau pejabat dari partai tertentu. Hal ini dapat dilihat dengan maraknya penggunaan media sosial dalam pemilihan umum presiden dan legislatif pada tahun 2019. Semakin banyaknya penggunaan media sosial terutama dalam pemberian opini publik terhadap objek politik, membuka peluang bagi pemerintah atau objek politik untuk mendapatkan akses informasi yang diperlukan. Organisasi dan partai politik mungkin tidak perlu lagi melakukan survei, jajak pendapat untuk mengumpulkan opini publik karena ada banyak informasi yang tersedia untuk umum melalui sosial media.

Opini publik dalam media sosial mendefinisikan penilaian publik secara keseluruhan terhadap objek politik. Untuk mengekstraksi tingkat penelitian publik tersebut dapat digunakan Analisis Sentimen. Analisis sentimen merupakan studi berbasis komputasi dari pendapat, emosi, penilaian, dan sikap dari manusia terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atributnya [4]. Proses ini akan dilakukan dengan menggunakan metode *Natural Language Processing* (NLP) dan metode analisis teks[5]. Suatu opini diolah agar mengetahui secara akurat tentang emosi yang telah tertulis pada opini tersebut, sehingga didapatkan informasi penilaian terhadap sebuah objek yang dapat digolongkan menjadi positif, netral atau negatif.

Opini publik yang ditulis oleh pengguna media sosial dapat mengandung berbagai aspek terhadap suatu objek. Contohnya dalam kata “Pembangunan infrastruktur sekarang sangat maju dan baik, tetapi sayangnya keuangan negara terancam jatuh karena hutang yang terus dilakukan”, sentimen terhadap aspek “pembangunan” adalah positif sedangkan sentimen untuk aspek

“keuangan negara” adalah negatif. Hal ini membuat analisis sentimen level aspek perlu diperlukan dibandingkan hanya dengan analisis sentimen secara keseluruhan kata. Penelitian sebelumnya telah menggunakan telah membuktikan bahwa interaksi antara aspek dengan teks secara keseluruhan penting dalam analisis sentimen [6].

Namun, menemukan dan menyaring informasi yang terkandung dalam opini publik dalam sosial media merupakan tugas yang berat. Banyaknya jumlah opini yang tidak sesuai dengan kaidah standar akan membuat penarikan kesimpulan sentimen level aspek dari objek menjadi sulit. Sehingga diperlukan sebuah sistem yang mampu melakukan klasifikasi sentimen level aspek terhadap objek. Penelitian yang ada telah menunjukkan berbagai teknik untuk berbagai tugas analisis sentimen. Salah satunya yaitu penggunaan deep learning yang sejak satu dekade yang lalu telah muncul sebagai teknik machine learning yang menjadi *state-of-the-art* dalam berbagai bidang NLP dan analisis sentimen [7].

Metode *Deep Learning* yang baik dalam melakukan analisis sentimen level aspek yang menggunakan data berurutan atau sekuensial adalah *Recurrent Neural Network (RNN)*[8]. *Gated Recurrent unit (GRU)* merupakan salah satu metode yang populer dalam RNN. GRU digunakan dalam metode *Hierarchical Attention Position-aware Network* yang mendapat nilai tertinggi dalam Dataset SemEval 2014 untuk tugas analisis sentimen level aspek [6]. Dimana metode ini menggunakan *bidirectional-GRU*, *attention*, informasi posisi aspek dalam sebuah kalimat serta penggunaan *word embedding*. Representasi posisi dari aspek dalam pembuatan model penting untuk melihat posisi kata tambahan dari target aspek. Contohnya dalam kata sebelumnya “maju dan baik” merupakan kata tambahan dari “pembangunan infrastruktur” dan memiliki sentimen positif. Sehingga dari permasalahan tersebut penelitian ini menawarkan solusi dalam melakukan analisis sentimen level aspek terhadap opini publik yang berkaitan

dengan objek politik menggunakan metode *Hierarchical Attention Position-aware Network*.

## **1.2. Rumusan Masalah**

Berdasarkan uraian pada latar belakang yang telah dikemukakan sebelumnya, maka rumusan masalah pada penelitian tugas akhir ini adalah sebagai berikut :

1. Bagaimana mendapatkan data teks bertopik politik yang tepat pada media sosial?
2. Bagaimana menentukan aspek dan sentimen yang sesuai untuk data teks bertopik politik?
3. Bagaimana menerapkan arsitektur *Hierarchical Attention Position-aware Network* untuk tugas klasifikasi sentimen level aspek?
4. Bagaimana performa klasifikasi sentimen level aspek dengan model *Hierarchical Attention Position-aware Network*?

## **1.3. Batasan Permasalahan**

Berdasarkan permasalahan yang telah diuraikan sebelumnya, adapun batasan masalah dalam penelitian tugas akhir ini adalah sebagai berikut :

1. Studi kasus yang digunakan pada penelitian ini merupakan data yang diperoleh dari media sosial *twitter* dengan topik politik.
2. Data yang digunakan merupakan data teks berbahasa Indonesia.
3. Anotasi aspek dilakukan berdasarkan kata benda dan merupakan subjek atau objek dalam kalimat dari data yang diperoleh.
4. Pengujian yang dilakukan menggunakan sampel data dari data yang telah didapatkan.



#### 1.4. Tujuan Penelitian

Tujuan besar dari penelitian tugas akhir ini adalah sebagai berikut:

1. Menerapkan metode *crawling* dan *filtering* data bertopik politik pada media sosial *twitter*.
2. Menentukan aspek dan sentimen dari aspek yang tepat untuk data teks bertopik politik.
3. Menerapkan arsitektur *Hierarchical Attention Position-aware Network* untuk mengklasifikasi sentimen level aspek dari data teks bertopik politik.
4. Melakukan analisis performa klasifikasi sentimen level aspek dengan model *Hierarchical Attention Position-aware Network*.

#### 1.5. Manfaat Penelitian

Berikut merupakan manfaat yang diharapkan dari penelitian tugas akhir ini:

1. Bagi penulis, untuk mengetahui dan memahami metode *word embedding* pada data teks, serta model dan arsitektur *Hierarchical Attention Position-aware Network* dan menerapkannya pada tugas klasifikasi sentimen level aspek.
2. Bagi masyarakat, sebagai bentuk penelitian dengan menggunakan metode *deep learning* dengan data teks bahasa Indonesia, sehingga memungkinkan penelitian selanjutnya ke dalam permasalahan-permasalahan lain pada NLP Bahasa Indonesia.

#### 1.6. Relevansi

Topik pada penelitian tugas akhir ini adalah analisis sentimen level aspek, sehingga relevan terhadap bidang keilmuan laboratorium Akuisisi Data dan Diseminasi Informasi (ADDI). Mata kuliah yang berkaitan dengan penelitian tugas akhir ini antara lain Sistem Cerdas, Penggalan Data dan Visualisasi Informasi.

*Halaman ini sengaja dikosongkan*

## BAB II TINJAUAN PUSTAKA

Pada bab ini akan membahas penelitian sebelumnya yang berhubungan dengan tugas akhir dan teori-teori yang berkaitan dengan permasalahan tugas akhir.

### 2.1. Penelitian Sebelumnya

Tabel 2.1 menampilkan daftar penelitian sebelumnya yang mendasari tugas akhir ini.

**Tabel 11.1 Studi Sebelumnya**

<b>1. Hierarchical Attention Based Position-aware Network for Aspect-level Sentiment Analysis</b>
<b>Penulis/Tahun/Sumber :</b> Li, Lishuang. Dkk; 2018 [6]
<b>Metode :</b> <ul style="list-style-type: none"><li>• Hierarchical Attention Based Position-aware Network</li><li>• GloVe word embedding</li><li>• Bi-GRU Based Sentence Encoder</li></ul>
<b>Pembahasan :</b> <p>Penelitian ini membahas tentang kinerja dari metode <i>Hierarchical Attention Based Position-aware Network</i> dalam melakukan klasifikasi sentimen level aspek pada data laptop dan restoran dari dataset SemEval 2014. Dan didapatkan hasil akurasi terbaik dibandingkan dengan metode-metode lain. Hasil dari penelitian ini membuktikan bahwa penggunaan <i>position embedding</i> untuk menghasilkan representasi posisi dan mekanisme berbasis <i>hierarchical attention</i> untuk memadukan informasi aspek dan kata keseluruhan dalam arsitektur model sangat baik.</p>

<p><b>Keterkaitan :</b></p> <p>Penelitian pada paper berkaitan dengan penggunaan metode <i>Hierarchical Attention Based Position-aware Network</i> pada data teks</p>
<p><b>2. Recurrent Attention Network on Memory for Aspect Sentiment Analysis</b></p>
<p><b>Penulis/Tahun/Sumber :</b></p> <p>Cheng P dkk; 2017 [9]</p>
<p><b>Metode :</b></p> <ul style="list-style-type: none"> <li>• <i>Recurrent Attention Network</i></li> <li>• <i>Bi-LSTM</i></li> <li>• <i>Position-Weighted Memory</i></li> </ul>
<p><b>Pembahasan :</b></p> <p>Penelitian ini menerapkan metode <i>Recurrent Attention Network</i> untuk mengidentifikasi sentimen terhadap target opini dalam data komentar atau review. Penelitian ini mengadopsi mekanisme <i>multiple-attention</i> untuk menangkap sentimen dari fitur yang terletak jauh dari target.</p>
<p><b>Keterkaitan :</b></p> <p>Penelitian pada paper ini berkaitan dengan jenis data tugas akhir yang akan di klasifikasi yaitu data teks dan penggunaan <i>position</i> dan <i>attention</i> untuk membangun model.</p>

### 3. Analisis Sentimen Teks Bahasa Indonesia Pada Media Sosial Menggunakan Algoritma Convolutional Neural Network (Studi Kasus : Politik)

**Penulis/Tahun/Sumber :**

Wianto, Prasetyo Wahyu Adi; 2018 [10]

**Metode :**

- *Convolutional Neural Network*
- *Word2Vec*

**Pembahasan :**

Penelitian ini menerapkan metode *Convolutional Neural Network* (CNN) dalam melakukan analisis sentimen terhadap data teks. Terdapat beberapa tahapan dalam penggunaan Algoritma CNN. Tahapan tersebut diawali dengan melihat hasil pengujian menggunakan output model *word embedding* yang paling baik. Pada penelitian didapatkan model *Word2Vec* menghasilkan nilai akurasi lebih baik dibandingkan model yang lainnya. Selanjutnya dilakukan pengujian algoritma *Convolutional Neural Network* dan dibandingkan dengan model lain yaitu SVM dan *Naïve Baiyes*. Hasil evaluasi pengukuran didapatkan bahwa model CNN lebih baik dari dua model lainnya.

**Keterkaitan :**

Penelitian pada tugas akhir ini berkaitan dengan jenis data tugas akhir yang akan di klasifikasi yaitu data teks untuk studi kasus politik

## 2.2. Landasan Teori

Landasan teori berisi teori-teori yang digunakan dalam pengerjaan penelitian tugas akhir. Dalam landasan teori, acuan yang digunakan adalah berdasarkan penelitian dan buku.

### 2.2.1. *Natural Language Processing*

*Natural Language Processing* (NLP) adalah bidang ilmu yang melakukan pengolahan bahasa alami, seperti ucapan dan teks dengan menggunakan perangkat lunak sebagai pengolahannya. Metode matematis dapat digunakan dalam memahami bahasa alami, hal ini dikarenakan terdapat suatu pola dan aturan dalam tiap bahasa [11]. Namun pengolahan bahasa bukan hal yang mudah dikarenakan kombinasi dari berbagai kosakata maupun aturan yang kompleks dalam suatu bahasa. Maka dari itu, penggunaan perangkat lunak dapat dilakukan untuk melakukan banyak perhitungan matematis dengan cepat. NLP memodelkan pengetahuan terhadap bahasa, baik dari segi kata, kalimat dan konteks kata dalam kalimat dengan menggunakan perangkat lunak.

Berikut ini adalah berbagai penerapan NLP [12]:

- *Information Retrieval*
- *Spell and Grammar Checking*
- *Machine Translation*
- *Summarization*
- *Question Answering*
- *Information Extraction*
- *Sentiment Analysis*
- *Word Prediction*
- *Text Categorization*
- *Speech Recognition*
- *Spoken Dialog Systems*

### 2.2.2. Analisis Sentimen

Analisis sentimen atau disebut juga *opinion mining* mengacu pada bidang yang luas dari pengolahan bahasa alami, komputasi linguistik dan *Text Mining*. Analisis sentimen adalah bidang studi yang menganalisis opini, sentimen, penilaian, penilaian, sikap, dan emosi orang terhadap entitas seperti produk, layanan, organisasi, individu, masalah, peristiwa, topik, dan atributnya [4]. Tugas dasar dalam analisis sentimen adalah mengelompokkan teks yang ada dalam sebuah kalimat atau dokumen kemudian menentukan sentimen yang ada dalam kalimat atau dokumen tersebut apakah bersifat positif atau negatif. Analisis sentimen juga dapat menyatakan perasaan emosional sedih, gembira, atau marah. Ekspresi atau sentimen mengacu pada fokus topik tertentu, pernyataan pada satu topik mungkin akan berbeda makna dengan pernyataan yang sama pada subject yang berbeda.

Para peneliti telah meneliti analisis sentimen pada tiga level yaitu level dokumen, level kalimat dan level aspek. Analisis sentimen tingkat dokumen mengklasifikasikan dokumen yang dikemukakan oleh seseorang (misal ulasan produk) sebagai pernyataan opini positif atau negatif secara keseluruhan [13]. Analisis sentimen level kalimat mengklasifikasikan kalimat individu dalam suatu dokumen. Secara tradisional, orang sering kali mengklasifikasikan kalimat sebagai kalimat opini atau bukan, yang disebut klasifikasi subjektivitas. Lalu hasilnya kalimat beropini diklasifikasikan sebagai opini positif, negative atau netral [13]. Analisis sentimen level aspek memiliki tugas untuk mengekstraksi dan meringkas opini orang yang diungkapkan pada entitas dan aspek/fitur entitas, yang juga disebut target. Misalnya, dalam ulasan produk, ini bertujuan untuk meringkas opini positif dan negatif masing-masing aspek produk, walaupun sentimen umum pada produk bisa positif atau negatif. Seluruh tugas analisis sentimen berbasis aspek terdiri dari beberapa subtask seperti ekstraksi aspek, ekstraksi entitas, dan klasifikasi aspek sentimen.

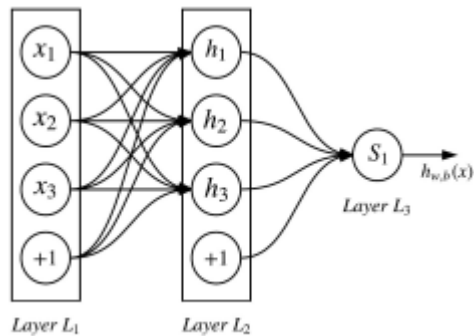
### 2.2.3. Analisis Sentimen Level Aspek

Analisis sentimen level aspek merupakan tugas penting dalam analisis sentimen [14]. Berbeda dengan Analisis sentimen level dokumen dan level kalimat, Analisis sentimen level aspek lebih mempertimbangkan baik sentimen dan informasi target, karena sentimen selalu memiliki target. Seperti disebutkan sebelumnya, target biasanya adalah entitas atau aspek entitas. Untuk kesederhanaan, baik entitas maupun aspek biasanya hanya disebut aspek. Diberikan kalimat dan target opini yang terdapat dalam kalimat, tugas analisis sentimen level aspek tersebut bertujuan untuk menentukan polaritas sentimen kalimat terhadap target opini. Target opini mengacu pada kata atau frasa (urutan kata) yang menggambarkan aspek sebuah entitas. Misalnya, dalam kalimat "Tempat ini memiliki dekorasi interior yang lucu dan harga terjangkau", target opininya adalah dekorasi interior dan harga. Dibandingkan dengan klasifikasi sentimen tingkat dokumen atau level kalimat, tantangan utama klasifikasi sentimen tingkat aspek adalah untuk membedakan sentimen terhadap target yang berbeda ketika ada beberapa target dalam sebuah kalimat. Misalnya, kalimat "Makanan sangat enak, tetapi layanannya lambat." Mengungkapkan sentimen positif pada target atau aspek makanan dan sentimen negatif pada target atau aspek layanan. Tugas tersebut baru-baru ini ditangani oleh *attention mechanisme*. Meskipun banyak teknik *deep learning* telah diusulkan untuk menangani klasifikasi sentimen, level aspek, masih belum ada teknik yang mendominasi dalam literatur [13].

### 2.2.4. Deep Learning

*Deep Learning* adalah penerapan dari *neural network* untuk tugas-tugas pembelajaran menggunakan jaringan dengan banyak lapisan [13]. *Deep Learning* dapat mengeksploitasi lebih banyak kekuatan pembelajaran (representasi) dari *neural network*, yang dulunya dianggap praktis hanya dengan satu atau dua lapisan dan sejumlah data kecil.





**Gambar 11.1** *Feedforward Neural Network*

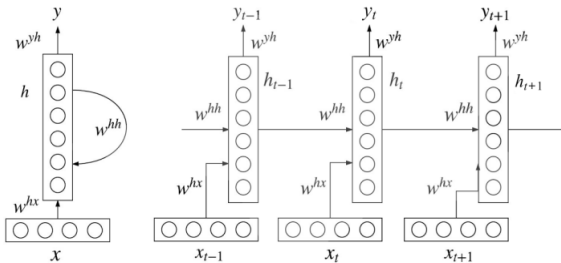
Gambar 2.1 merupakan contoh sederhana dari neural network, yang terdiri dari tiga lapisan  $L_1$ ,  $L_2$  dan  $L_3$ .  $L_1$  adalah lapisan input, yang sesuai dengan vektor input  $(x_1, x_2, x_3)$  dan *intercept*  $+1$ .  $L_3$  adalah lapisan output, yang sesuai dengan vektor keluaran  $(S_1)$ .  $L_2$  adalah lapisan tersembunyi, yang hasilnya tidak terlihat sebagai output jaringan. Lingkaran di  $L_1$  mewakili elemen dalam vektor input, sementara lingkaran di  $L_2$  atau  $L_3$  mewakili neuron, elemen perhitungan dasar dari *neural network* atau disebut fungsi aktivasi. Garis antara dua neuron mewakili koneksi untuk aliran informasi. Setiap koneksi dikaitkan dengan bobot, atau nilai yang mengendalikan sinyal antara dua neuron. Pembelajaran *neural network* dicapai dengan menyesuaikan bobot antara neuron dengan informasi yang mengalir melaluinya hal ini menyerupai proses pembelajaran otak pada manusia.

Para peneliti kehilangan minat dalam *neural network* pada akhir 1990-an terutama karena mereka dianggap hanya praktis untuk *neural network* dan pada *deep learning* dianggap rumit dan menggunakan komputasi sangat mahal. Namun, dalam 10 tahun terakhir, *deep learning* membuat terobosan dan menghasilkan hasil yang canggih di banyak domain aplikasi, mulai dari *computer vision*, *speech recognition*, dan yang terbaru, NLP [7]. Kebangkitan kembali neural network dapat dikaitkan dengan banyak faktor. Yaitu meliputi: (1) ketersediaan daya

komputasi karena kemajuan perangkat keras (misalnya, GPU),  
 (2) ketersediaan data pelatihan dalam jumlah besar [15].

### 2.2.5. Recurrent Neural Network (CNN)

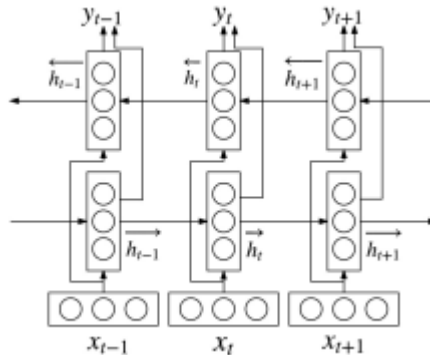
*Recurrent neural network* (RNN) adalah *neural network* yang memiliki koneksi antar neuron dan membentuk siklus terarah. Tidak seperti *feedforward neural network*, RNN dapat menggunakan “*memory*” internalnya untuk memproses urutan input, yang membuatnya populer untuk memproses informasi sekuensial. *Memory* berarti bahwa RNN melakukan tugas yang sama untuk setiap elemen dari urutan dengan setiap keluaran bergantung pada semua perhitungan sebelumnya. Hal ini seperti mengingat informasi tentang apa yang telah diproses sejauh ini [13].



**Gambar 11.2 Recurrent Neural Network**

Gambar 2.2 menunjukkan contoh RNN. Grafik kiri adalah jaringan tanpa lipatan dengan siklus, sedangkan grafik kanan adalah jaringan urutan terlipat dengan tiga langkah waktu. Panjang langkah waktu ditentukan oleh panjang input. Sebagai contoh, jika urutan kata yang akan diproses adalah kalimat enam kata, RNN akan dibuka menjadi *neural network* dengan enam langkah atau lapisan waktu. Satu lapisan sesuai dengan sebuah kata.

Secara teoritis, RNN dapat menggunakan informasi dalam urutan panjang yang berubah-ubah, tetapi dalam praktiknya, RNN terbatas untuk melihat ke belakang hanya beberapa langkah karena terdapat masalah yaitu hilangnya atau meledaknya gradien [16].



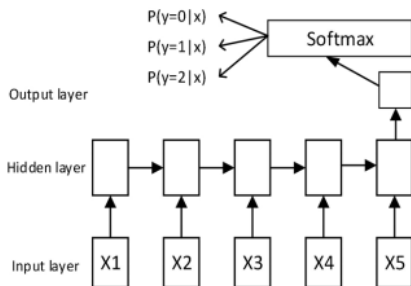
**Gambar 11.3** *Bidirectional Recurrent Neural Network*

Para peneliti telah mengembangkan jenis RNN yang lebih canggih untuk mengatasi kekurangan model RNN standar: *Bidirectional RNN*, *Long Short Term Memory*, *Gated Recurrent Unit*. *Bidirectional RNN* didasarkan pada gagasan bahwa output pada setiap waktu mungkin tidak hanya bergantung pada elemen sebelumnya dalam urutan, tetapi juga tergantung pada elemen berikutnya dalam urutan. Misalnya, untuk memprediksi kata yang hilang dalam urutan, mungkin diperlukan melihat konteks kiri dan kanan. *Bidirectional RNN* terdiri dari dua RNN, yang ditumpuk satu sama lain [17]. Yang memproses input dalam urutan aslinya dan yang memproses urutan input terbalik. Output kemudian dihitung berdasarkan keadaan tersembunyi dari kedua RNN.

### 2.2.6. *Gated Recurrent Unit*

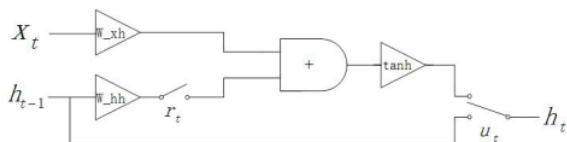
*Gated Recurrent Unit* (GRU) pertama kali diperkenalkan oleh Chung dkk., pada tahun 2014 [18]. Tujuan utama dari pembuatan GRU adalah untuk membuat setiap *recurrent unit*

untuk dapat menangkap *dependencies* dalam skala waktu yang berbeda-beda secara adaptif.



**Gambar 11.4 Struktur model berbasis GRU**

Seperti diilustrasikan pada Gambar 2.4, ada lapisan input yang terdiri dari beberapa neuron, jumlah neuron ditentukan oleh ukuran ruang fitur. Demikian pula, jumlah neuron di lapisan output sesuai dengan ruang output. Lapisan tersembunyi yang berisi sel memori mencakup fungsi utama jaringan GRU. Perubahan dan pemeliharaan status sel tergantung pada dua gerbang dalam sel: *reset gate*  $r_t$  dan *update gate*  $u_t$ . Struktur sel memori diilustrasikan sebagai diagram sirkuit pada Gambar 2.4.



**Gambar 11.5 Struktur memory GRU**

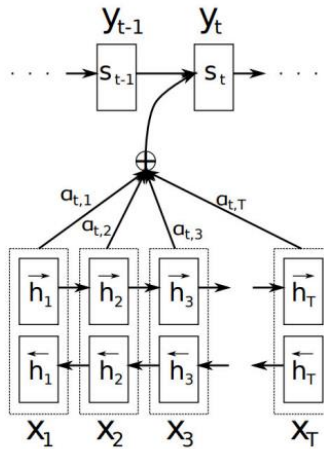
Kedua gerbang terkait dengan dua elemen:  $x_t$  dan  $h_{t-1}$ . Yang pertama berasal dari urutan input, yang terakhir adalah nilai output sel memori dari titik waktu sebelumnya. Dengan demikian, setiap gerbang memenuhi tugas yang berbeda untuk mencapai tujuan penyingkapan:

- *Reset gate* digunakan untuk mengontrol pengaruh  $h_{t-1}$  (informasi unit pada catatan waktu sebelumnya) pada informasi terkini  $x_t$ . Jika  $h_{t-1}$  tidak penting untuk  $x_t$ , maka  $r_t$  dapat dibuka, membuat  $h_{t-1}$  tidak mempengaruhi  $x_t$ .
- *Update gate* menentukan apakah informasi terkini  $x_t$  akan diabaikan. Ketika  $u_t$  diaktifkan pada cabang bawah, informasi terkini  $x_t$  akan diabaikan, sembari membentuk "*short-circuit connection*" dari  $h_{t-1}$  ke  $h_t$ . Ini membuat gradien menyebar secara terbalik, dan secara efektif menyelesaikan masalah hilangnya gradien.

Jaringan GRU memproses input yang sesuai pada setiap langkah waktu  $t$ . Jaringan akan mengembalikan output final hanya jika selesai memproses elemen terakhir dalam urutan. Metode penyesuaian parameter (bobot dan bias) mirip dengan yang ada pada jaringan neural feedforward tradisional. Selama pelatihan, hilangnya fungsi tujuan dari set pelatihan diminimalkan [19].

### **2.2.7. Attention Mechanisme**

*Attention Mechanisme* dalam neural network terinspirasi oleh mekanisme perhatian visual yang ditemukan di manusia. Dimana, perhatian visual manusia dapat fokus pada wilayah tertentu dari suatu gambar dengan "resolusi tinggi" sambil melihat gambar di sekitarnya dalam "resolusi rendah" lalu menyesuaikan titik fokus dari waktu ke waktu. Dalam NLP, *attention mekanisme* memungkinkan model untuk mempelajari apa yang harus diperhatikan berdasarkan pada teks input dan apa yang telah dihasilkan sejauh ini. Daripada meng-*encode* teks secara lengkap menjadi vektor dengan panjang yang tetap seperti standar dari RNN [13].



**Gambar 11.6** *Attention Mechanisme dalam bidirectional rnn.*

*Attention mechanism* ini pertama kali dimanfaatkan untuk *machine translation* dalam NLP [20]. Dimana *encoder-decoder framework* diusulkan dengan penggunaan *attention mechanism* untuk memilih referensi target kata dalam bahasa asli untuk kata-kata dalam bahasa target sebelum diterjemahkan. Gambar 2.5 menggambarkan penggunaan *attention mechanism* dalam *bidirectional RNN*. Dapat dilihat bahwa setiap kata keluaran dari *decoder*  $y_t$  tergantung pada kombinasi berbobot dari semua input *states*. Tidak hanya *states* terakhir seperti dalam kasus normal.  $a_{t,T}$  adalah bobot yang menentukan seberapa banyak dari masing-masing kondisi input harus dikalikan sesuai bobot untuk tiap output. Contohnya, jika  $a_{2,2}$  memiliki nilai yang besar, artinya *decode* harus mengambil banyak perhatian pada *states* kedua dalam kalimat sembari menghasilkan kata kedua dari kalimat target.

### 2.2.8. *Word Embedding*

Banyak model *deep learning* dalam NLP membutuhkan hasil *word embedding* sebagai input fitur [21]. *Word embedding* adalah teknik untuk *language modelling* dan *transfer learning*,

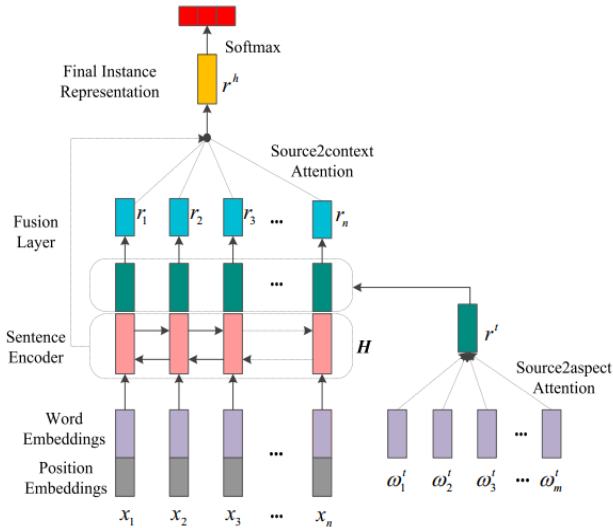
yang mengubah kata-kata dalam kosakata menjadi vektor bilangan real kontinu (contoh., kata "topi"  $\rightarrow$  (... , 0,15, ... , 0,23, ... , 0,41, ...). Teknik ini biasanya melibatkan *mathematic embedding* dari vektor berdimensi tinggi menjadi vektor berdimensi rendah. Setiap dimensi dari *word embedding* mewakili fitur dari sebuah kata. Setiap vektor dapat meng-*encode* pola linguistik dalam sebuah kalimat [21]. Ketika proses pelatihan dilakukan pada *word embedding*, maka hasil yang didapatkan salah satunya berupa kemiripan atau kedekatan antara kata maupun juga relasi yang lainnya.

### 2.2.9. *Word2Vec*

*Word2Vec* adalah salah satu implementasi metode *word embedding* yang dibuat oleh Tomas Mikolov, dkk [22]. *Word2Vec* bertujuan untuk merepresentasikan kata-kata kedalam vektor yang rapat dan memiliki dimensi yang rendah. Tomas Mikolov menggunakan sebuah metode yang merupakan hasil memodifikasi NNLM (*Neural Networks Language Model*) dimana *neural networks* dapat bekerja dengan baik dalam menjaga keteraturan linear antara kata [22]. Selain itu *Word2Vec* juga lebih cepat dalam proses pelatihan dan dapat menangani dataset dengan skala yang besar. Arsitektur model yang dapat digunakan dalam *Word2Vec* ada 2 yaitu *Continuous Bag-of-Words* dan *Continuous Skip-gram*. Berdasarkan eksperimen, keputusan krusial yang mampu mempengaruhi kinerja dari *Word2Vec* adalah pemilihan model yang digunakan, ukuran dari vektor, *subsampling rate*, dan *training window* [23]

### 2.2.10. *Hierarchical Attention Position-aware Network*

*Hierarchical Attention Position-aware Network* merupakan sebuah model *deep learning* berbasis hirarki yang menggunakan *position embedding* saat memodelkan kalimat untuk menghasilkan representasi posisi dari aspek selain itu digunakan mekanisme *attention* untuk mendapatkan sentimen dari aspek dalam kalimat [6]. Berikut merupakan arsitektur dari model HAPN:



**Gambar 11.7** *Arsitektur Hierarchical Attention Position-aware Network.*

Dalam pendekatan ini, setiap target bersama dengan kalimat dari target akan menjadi sebuah masukan. Terdapat kalimat terdiri atas  $w = \{w_1, w_2, \dots, w_n\}$  dan sebuah target yang memiliki  $m$  kata  $w^t = \{w_1^t, w_2^t, \dots, w_m^t\}$ . Dimana  $w^t$  merupakan bagian dari kalimat  $w$ . tujuan dari model ini adalah untuk memprediksi sentemen dari target dalam kalimat. Dapat dilihat dalam gambar, model ini memiliki 4 bagian yaitu: *input embedding*, *Bi-GRU based encoding layer*, *hierarchical attention based fusion layer* dan *output layer*.

### 1. Input Embedding

*Layer embedding* memiliki dua bagian yaitu *word embedding* dan *position embedding*.  $w_w \in \mathbb{R}^{d_w \times v_w}$  merupakan *word embedding*, dimana  $d_w$  adalah dimensi dari *word embedding* dan  $v_w$  adalah ukuran dari kamus kata. Selain itu juga terdapat *position embedding*  $w_p \in \mathbb{R}^{d_p \times v_p}$  dimana  $d_p$  adalah dimensi dari *position*



*embedding* dan  $v_p$  adalah ukuran dari kemungkinan posisi antara tiap kata dengan target. Dimana posisi ini akan di inialisasi secara random, dan diubah saat fase pelatihan.

## 2. *Bi-GRU Based Sentence Encoder*

Bi-GRU digunakan untuk mempelajari lebih tentang representasi abstrak dari sebuah kalimat [24]. Berikut ini dijelaskan *layer encoding* dalam detail.

Pada fase *encoding*, pertama tiap token  $w_i$  dalam kalimat diubah menjadi vektor  $x_i$  menggunakan penggabungan dari beberapa vektor berikut:

- a. *Word embedding*  $w_i$  dari  $w_t$ .
- b. *Position embedding* dari  $w_i$ , dimana posisi relevan antara kata ke- $i$  dan target didefinisikan dengan kalkulasi berikut:

$$\begin{cases} i - k & i < k \\ i - k - m & n \geq i > k + m \\ 0 & k + m \geq i > k \end{cases} \quad (1)$$

Dimana  $k$  merupakan indeks dari kata pertama dalam target,  $m$  merupakan panjang dari target dan  $n$  merupakan panjang dari kalimat. Vektor *position embedding* diperoleh dengan inialisasi secara acak pada tabel *embeddings* sesuai dengan posisi yang relevan.

Dengan begitu maka kalimat dapat direpresentasikan menjadi  $x = \{x_1, x_2, \dots, x_n\}$ . Kemudian dijalankan dua parallel GRU *layer*, yaitu GRU maju dan GRU mundur. GRU maju dijalankan untuk membuat representasi *hidden* ( $\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n$ ) dan GRU mundur dijalankan untuk mendapatkan representasi *hidden* ( $\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_n$ ). Sehingga didapatkan sebuah representasi baru  $H = (h_1, h_2, \dots, h_n)$  dengan menggabungkan *hidden* vektor pada ( $\vec{h}_1, \vec{h}_2, \dots, \vec{h}_n$ ) dan ( $\overleftarrow{h}_1, \overleftarrow{h}_2, \dots, \overleftarrow{h}_n$ ):  $h_i = [\vec{h}_i, \overleftarrow{h}_i]$ . Catat bahwa  $h_i \in \mathbb{R}^{2d_h}$  mengenkapsulasi informasi

konteks pada keseluruhan kalimat ( dari 1 ke n) dengan fokus yang lebih besar pada posisi I, dimana  $d_h$  merupakan dimensi dari *hidden state*. Berdasarkan pengenalan terhadap *position embedding*,  $h_i$  dapat disebut sebagai *position aware*.

### 3. Hierarchical Attention Based Fussion Layer

Pada bagian ini, digambarkan mekanisme ringkas yang digunakan untuk menggabungkan informasi target dan kalimat utuh. Secara rinci *Source2aspect attention* pertama kali digunakan untuk menangkap petunjuk paling penting dalam kata-kata target dan representasi aspek diperoleh. Selanjutnya, representasi aspek-spesifik dari setiap kata dihasilkan berdasarkan representasi aspek dan representasi posisi yang di-*encode*. *Source2context attention* kemudian digunakan untuk menangkap sentimen yang paling indikatif kata-kata dalam konteks dan menghasilkan jumlah *weighted embeddings* sebagai kalimat final.

*Source2aspect attention*: Berdasarkan fakta bahwa aspek setidaknya lebih dari 2 kata [25]. mekanisme *Sorce2aspect* digunakan untuk membuat representasi dari aspek. *Source2aspect attention* terinspirasi dari penelitian tentang *self-attention network* [26]. Pertama akan digunakan fungsi penilaian dengan mengambil *word embedding* dari tiap kata pada target sebagai masukan.

$$f_t(\omega_i^t) = \tan h(W_t \cdot \omega_i^t) \quad (2)$$

Dimana  $W_t \in \mathbb{R}^{d_w}$  adalah *weight vector* dan  $\tan h$  adalah fungsi non linear. Nilai dari  $f_t$  kemudian digunakan untuk memberi *weight* pada kata penting dalam target. Pada basis ini, *weight* terpenting yang telah dinormalisasi dari kata ke I dalam target  $a_i^t$  dihitung dengan cara berikut:

$$a_i^t = \frac{\exp(f_t(\omega_i^t))}{\sum_{j=1}^m \exp(f_t(\omega_j^t))} \quad (3)$$

Di akhir, kombinasi *weight* dari *word embedding* ditentukan sebagai representasi dari target.

$$r^t = \sum_{t=1}^m a_i^t \cdot \omega_i^t \quad (4)$$

*Information Fusion*: seteah mendapatkan representasi dari target, kemudian representasi digunakan untuk mengkonstruksi representasi dari spesifik target dari tiap kata dalam kalimat dengan rumus berikut:

$$r_i = W_s \cdot [h_i, r^t] \quad (5)$$

Dimana  $W_s \in \mathbb{R}^{(2d_h + d_w) \times d_w}$  adalah weight matriks.  $r_i$  merupakan representasi spesifik target dari kata  $w_i$  ke  $i$  dalam kalimat.

*Source2context Attention*: representasi spesifik target dari tiap kata kemudian digunakan untuk mempelajari *attention* dan kemudian dibuat representasi akhir dari kalimat. *Attention* didefinisikan dalam rumus berikut:

$$f_s([r_i, h_i]) = \tan h(W_c \cdot [r_i, h_i]) \quad (6)$$

$$a_i = \frac{\exp(f_s([r_i, h_i]))}{\sum_{j=1}^m \exp(f_s([r_j, h_j]))} \quad (7)$$

Dimana  $W_c \in \mathbb{R}^{2d_h + d_w}$  merupakan weight vector dan  $a_i$  menggambarkan kepentingan dari kata  $i$  dalam kalimat.

Terakhir, kombinasi dari weight dalam hidden state position-aware dituliskan sebagai berikut:

$$r^h = \sum_{i=1}^n a_i \cdot h_i \quad (8)$$

Yang merupakan representasi final dari instant ini.

#### 4. Output and Model training

Setelah mendapat representasi akhir  $r^h$  dari instans setelah tiga bagian dijalankan. Representasi akhir kemudian dimasukkan kedalam *softmax layer* untuk memprediksi target sentimen.

Diberikan semua data pelatihan  $(x^{(i)}, y^{(i)})$ , dapat didefinisikan *loss function* sebagai *negative log-likelihood*:

$$\mathcal{L}(\theta) = - \sum_{j=1}^n \log p(y^{(i)} | x^{(j)}, \theta) \quad (9)$$

### 2.2.11. Character Embedding

Xiang dan Yann pada tahun 2016 memperkenalkan *Character CNN*, atau penggunaan karakter dalam *word embedding* yang dapat meningkatkan kinerja model [27]. Dalam penelitiannya, daftar karakter didefinisikan dengan 70 karakter yang termasuk 26 huruf Inggris, 10 digit, 33 karakter khusus dan karakter baris baru.

Dalam bahasa Inggris, semua kata dibentuk oleh 26 huruf (atau 52 jika menyertakan karakter huruf besar dan kecil). Dengan penyisipan karakter, vektor setiap kata dapat dibentuk meskipun kata-kata tersebut tidak memiliki kosakata. Di sisi lain, *word embedding* hanya dapat menangani kata-kata yang terlihat atau ada dalam kamus kata. Manfaat lain dari *character embedding* adalah cocok untuk kata-kata yang salah dalam pengejaan, emotikon dan kata-kata baru. *Character embedding* dapat menangani kata-kata yang jarang lebih baik daripada *word2vec*. Alasan lain adalah dengan hanya ada sedikit vektor, dapat mengurangi kompleksitas model dan meningkatkan kinerja atau kecepatan.

*Character embedding* merupakan desain yang bagus untuk memecahkan banyak masalah dalam klasifikasi teks. *Character Embedding* menyelesaikan beberapa masalah *word embedding*. Perbedaan antara *character embedding* dan *word embedding* adalah bahwa *character embedding* dapat membangun kata apa saja selama karakter tersebut dimasukkan [28].

### 2.2.12. Ukuran Evaluasi Klasifikasi Sentimen

Ukuran evaluasi yang berbeda tersedia ketika menghitung nilai evaluasi untuk klasifikasi. Van Rijsbergen menetapkan *recall* (R) dan *precision* (P) untuk keperluan *information retrieval*. Selain itu juga diperkenalkan efektivitas, yang dapat ditulis ulang dalam *F-score* atau *F-Measure* [29]. Dalam klasifikasi

sentimen terdapat ukuran baru yang dikembangkan dari ukuran yang dikenalkan oleh van Risjbergen. Ukuran utama untuk klasifikasi ini adalah *AvgRec*, atau rata-rata *recall*, dimana merupakan rata-rata nilai recall di seluruh kelas positif (P), negatif (N), dan netral (U). Ukuran ini dihitung sebagai berikut:

$$AvgRec = \frac{1}{3}(RP + RN + RU) \quad (10)$$

Dimana *RP*, *RN* dan *RU* merujuk ke *recall* sehubungan dengan kelas positif, negatif, dan netral, masing-masing.

*AvgRec* berkisar dalam [0, 1], di mana nilai 1 hanya dicapai oleh pengklasifikasi sempurna (yaitu, pengklasifikasi yang mengklasifikasikan semua item dengan benar), nilai 0 hanya dicapai oleh pengklasifikasi yang gagal mengelompokkan semua item dengan benar. Keuntungan dari *AvgRec* daripada akurasi "standar" adalah bahwa ukuran ini lebih kuat untuk kelas yang tidak seimbang. Akurasi klasifikasi untuk kelas mayoritas adalah frekuensi relatif dari kelas mayoritas, yang mungkin jauh lebih tinggi dari 0,5 jika data tes tidak seimbang. Standar F1 juga peka terhadap ketidakseimbangan kelas karena alasan yang sama. Keuntungan lain dari *AvgRec* daripada F1 adalah bahwa *AvgRec* tidak berubah dalam hal beralih positif dengan negatif, sedangkan F1 tidak [30].

Ukuran lain yang dapat digunakan yaitu: akurasi dan  $F1_N^P \cdot F1_N^N$  adalah *macro-average F1*, yang dihitung dari kelas positif dan negatif (tidak menggunakan kelas netral).  $F1_N^P$  Itu dihitung sebagai berikut:

$$F1_N^P = \frac{1}{2}(F1^P + F1^N) \quad (11)$$

Dimana  $F1^P$  dan  $F1^N$  merujuk ke F1 sehubungan dengan masing-masing kelas positif dan negatif [30].

### 2.2.13. Media Sosial Twitter

Media sosial merupakan sebuah platform dimana pengguna memiliki halaman pribadi dan dapat saling berbagi informasi kepada publik melalui dunia virtual di Internet.

*Twitter* adalah suatu platform media sosial berupa *microblog* yang memberikan fasilitas bagi pengguna untuk membuat kiriman berupa tulisan dengan panjang maksimal 280 karakter. *Twitter* berdiri pada tahun 2006 dan terus berkembang hingga pada puncaknya mencapai 95 juta *tweet* per hari pada September 2010 [31]. *Twitter* merupakan salah satu media sosial yang unik karena terbatasnya jumlah karakter pada setiap *tweet*. Hal ini mempengaruhi hasil tulisan pengguna yang berbeda dengan media sosial lain. Dimana dalam *twitter* tulisan cenderung tidak basa-basi dan tepat sasaran. Tulisan dalam *twitter* yang singkat dan jelas membuat pelabelan secara manual lebih mudah dilakukan karena kata yang dibaca lebih sedikit dan lebih mudah dipahami manusia.

#### **2.2.14. Politik**

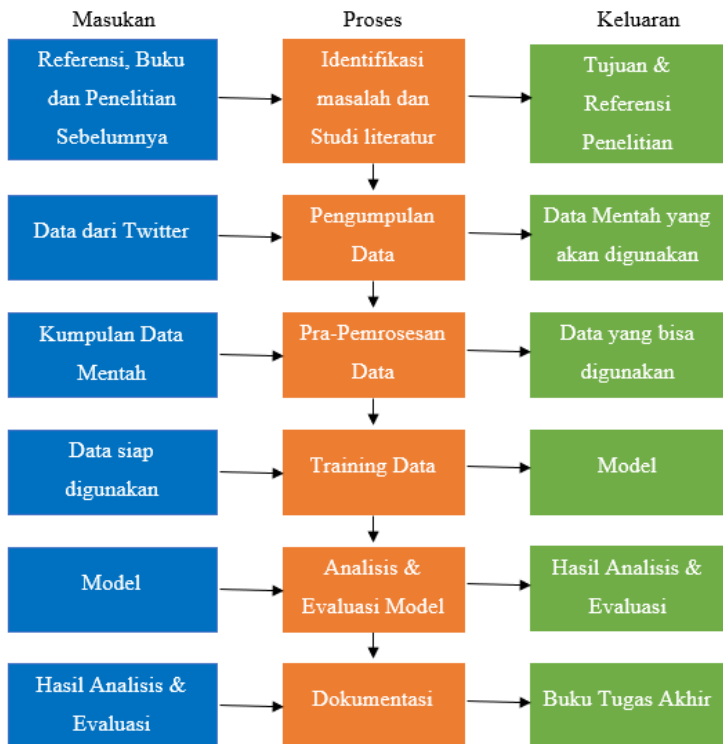
Politik adalah kegiatan suatu bangsa yang bertujuan untuk membuat dan mempertahankan peraturan-peraturan umum yang mengatur kehidupannya [32]. Dimana dalam pelaksanaan politik proses interaksi pembuatan keputusan publik juga tidak lepas dari konflik antara individu dan kelompok. Dimana aspirasi dan kepentingan setiap kelompok dan individu dalam masyarakat tidak selalu sama, melainkan berbeda [33]. Sehingga, masing kelompok saling mempengaruhi agar suatu keputusan publik yang disepakati sesuai dengan kepentingan kelompok tertentu.

## BAB III METODOLOGI PENELITIAN

Bab ini akan menjelaskan tentang gambaran metode dan alur pengerjaan tugas akhir. Metodologi akan digunakan sebagai panduan dalam penyusunan tugas akhir agar terarah dan sistematis.

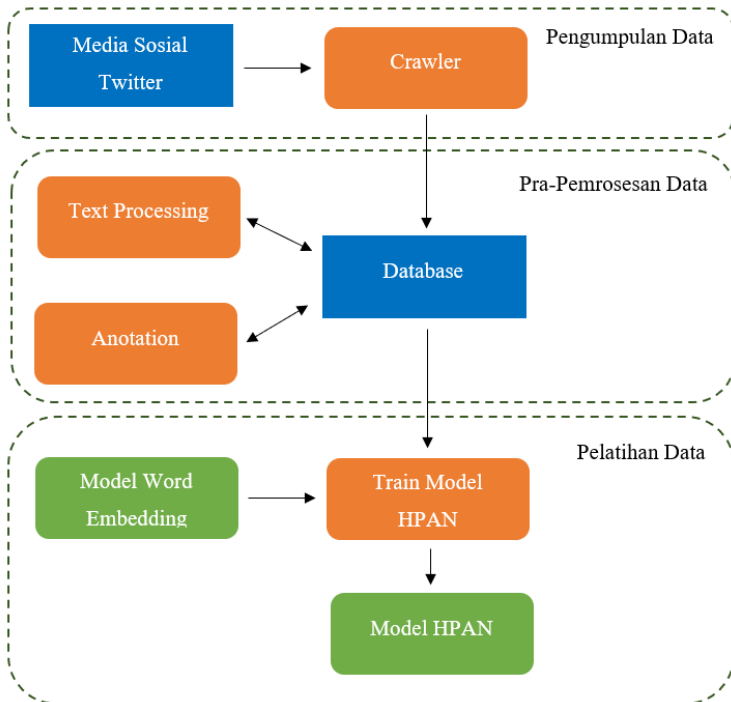
### 3.1. Tahapan Pelaksanaan Penelitian Tugas Akhir

Pada bagian ini akan dijelaskan mengenai metodologi atau tahapan pelaksanaan penelitian tugas akhir.



**Gambar 12.1 Alur Pengerjaan Tugas Akhir**

Untuk menjelaskan ringkasan pada keseluruhan proses penelitian tugas akhir ini, maka dijelaskan pada gambar berikut:



**Gambar 12.2** Arsitektur Penelitian Tugas Akhir

### 3.1.1. Studi Literatur

Tahapan ini merupakan fase pertama dalam pengerjaan tugas akhir ini. Pada tahap ini dilakukan identifikasi masalah penggalan kebutuhan pengetahuan terkait dengan studi kasus yang akan diambil. Kemudian aktivitas selanjutnya adalah melakukan studi literatur berdasarkan referensi penelitian sebelumnya dan buku yang berkaitan dengan topik tugas akhir ini. Literatur yang dijadikan referensi adalah tentang penelitaian dengan topik Analisis Sentimen Level Aspek. Paper utama yang dijadikan pedoman dalam penelitian tugas adalah



paper berjudul “*Hierarchical Attention Based Position-aware Network for Aspect-level Sentiment Analysis*”. Serta penelitian lain berjudul “*Recurrent Attention Network on Memory for Aspect Sentiment Analysis*”. Metode pada kedua paper tersebut nantinya dijadikan dasar untuk pengerjaan penelitian tugas akhir ini.

### 3.1.2. Pengumpulan Data

Pada tahap awal dilakukan pengumpulan data sebagai bahan masukan untuk tugas akhir ini. Data yang dicari merupakan data dengan topik politik dengan tipe data berupa teks. Data akan diambil dari salah satu situs media sosial yaitu *twitter*. Teknik pengambilan data akan dilakukan dengan *crawling* menggunakan API yang disediakan oleh platform media sosial *twitter*. Untuk mengambil data sesuai dengan topik politik, proses pengambilan data menggunakan beberapa kata kunci yang berkaitan dengan politik. Untuk entitas dari kata kunci yang digunakan dijelaskan pada tabel 3.1.

**Tabel 12.1 Daftar Entitas untuk kata kunci dengan topik politik**

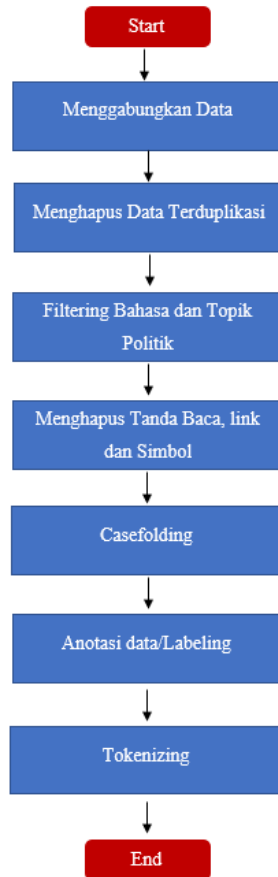
<b>Kategori</b>	<b>Entitas</b>
Partai Politik	Partai Kebangkitan Bangsa
	Partai Gerindra
	Partai Demokrasi Indonesia Perjuangan
	Partai Golkar
	Partau Nasional Demokrat
	Partai Garuda
	Partai Berkarya
	Partai Keadilan Sejahtera
	Partai Perindo
	Partai Persatuan Pembangunan
	Partai Solidaritas Indonesia
	Partai Amanat Nasional
	Partai Hanura
	Partai Demokrat
Partai Bulan Bintang	
Partai Keadilan dan Persatuan Indonesia	
Tokoh Politik	Jokowi

Kategori	Entitas
	Prabowo
	Maaruf Amin
	Sandiaga Uno
	Anies Baswedan
	Basuki Tjahaja Purnama
	Ganjar Pranowo
	Ridwan Kamil
	Tri Rismaharini
	Fahri Hamzah
	Fadli Zon
	Mahfud MD
Isu Politik	DPR
	Kabinet
	Kementerian
	KPK
	Pemilu
	RAPBD
	RKUHP

Data mentah akan dikumpulkan dengan menggunakan *crawler twitter* menggunakan *library tweepy*. Nantinya, data mentah yang telah dikumpulkan akan disimpan bentuk data CSV.

### 3.1.3. Pra-Pemrosesan Data

Data yang telah di peroleh akan masuk ke tahapan pra-pemrosesan data. Dimana setiap data yang akan digunakan pada proses *training* model akan memasuki tahapan pra-pemrosesan data satu per satu, Tahapan dari pra-pemrosesan data adalah pada diagram berikut:



Gambar 12.3 Proses Pra-Pemrosesan Data

#### a. Menggabungkan Data

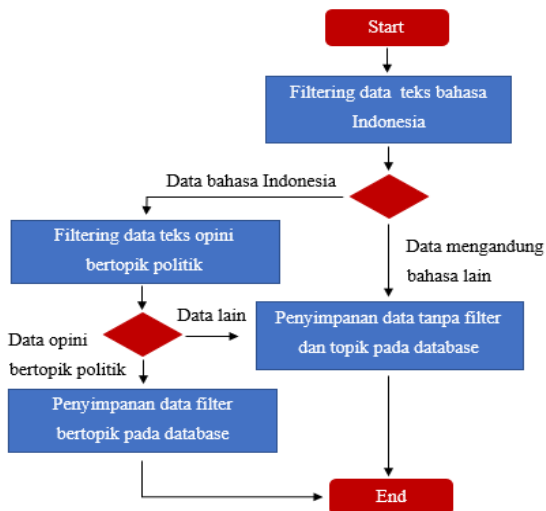
Data yang didapatkan dari proses pengumpulan data masih terpisah. Dimana data yang telah didapatkan akan terbagi ke beberapa *file* sesuai dengan kata kunci yang digunakan saat proses *crawling*. Proses penggabungan data akan dilakukan dengan menggunakan *library pandas* pada *python* dan diubah kembali menjadi *file CSV*.

## b. Menghapus Data Terduplicasi

Data yang telah digabungkan dan disimpan dalam database akan dilakukan deteksi terhadap duplikasi atau kesamaan isi data opini. Proses deteksi dilakukan dengan kombinasi *query* dan pemeriksaan secara manual. Sehingga hanya didapatkan kata yang unik saja. Hal ini dilakukan untuk mengurangi adanya redundansi data sehingga data yang didapatkan menjadi lebih *reliable*.

## c. *Filtering* Bahasa dan Topik Politik

Data yang unik kemudian dilakukan proses penyaringan bahasa, dimana data teks dengan bahasa bukan Indonesia akan dihapus dari *database*. Selain itu penyaringan data juga dilakukan untuk menyaring data yang merupakan kalimat dengan opini topik politik. Data yang hanya merupakan sapaan atau tidak berkaitan dengan politik atau opini terhadap politik walaupun telah mengandung kata kunci akan dihapus dari data unik. Proses *filtering* data ini akan dilakukan secara manual berdasarkan *Human Intelligence Task*.



**Gambar 12.4** *Filtering* Bahasa dan Topik Politik

#### d. Menghapus Tanda Baca, Link dan Simbol

Pada tahapan ini, setiap data opini akan dihapus tanda baca, *link url* dan simbol-simbol yang tidak bermakna. Tujuan dari proses ini adalah menghilangkan karakter yang tak bermakna serta memperkecil peluang kata yang memiliki makna sama namun berbeda penulisan karena penambahan tanda baca. Proses penghapusan dilakukan dengan metode *replacing* pada pola karakter tertentu. Berikut merupakan contohnya yang dapat dilihat pada Tabel 3.2.

**Tabel 12.2 Contoh Penghapusan Tanda Baca, Link dan Simbol**

Sebelum Pra-Proses	Setelah Pra-Proses
Sungguh bangga saya sebagai WNI..Luar biasa Indonesia bisa memiliki Presiden seperti Jokowi. Tanpa perlu bersuara keras tetap sanggup mencuri perhatian dunia dan diakui sebagai... <a href="https://www.instagram.com/p/BxL7ceIFU80/?utm_source=ig_twitter_share&amp;igshid=1d3g617lvoux0...">https://www.instagram.com/p/BxL7ceIFU80/?utm_source=ig_twitter_share&amp;igshid=1d3g617lvoux0 ...</a>	Sungguh bangga saya sebagai WNI Luar biasa Indonesia bisa memiliki Presiden seperti Jokowi Tanpa perlu bersuara keras tetap sanggup mencuri perhatian dunia dan diakui sebagai
Mantap Pak Prabowo ketegasan itu yang kami perlukan tentang Pernyataan saudara AM Hendropriyono yg bersifat RASIS dan Berpotensi untuk Mengadu Domba dan Memecah Belah antara Anak Bangsa! #GileLoeNdroRasisAmat	Mantap Pak Prabowo ketegasan itu yang kami perlukan tentang Pernyataan saudara AM Hendropriyono yg bersifat RASIS dan Berpotensi untuk Mengadu Domba dan Memecah Belah antara Anak Bangsa

### e. Casefolding

Pada tahapan ini akan mengubah setiap kata menjadi *lowercase*/huruf kecil atau disebut *casefolding*. Sehingga nantinya saat pemrosesan data tidak terjadi ambiguitas. Berikut adalah contoh dari proses *lowercase* yang dapat dilihat pada Tabel 3.3.

**Tabel 12.3 Contoh Proses Casefolding**

Sebelum Casefolding	Setelah Casefolding
Sungguh bangga saya sebagai WNI Luar biasa Indonesia bisa memiliki Presiden seperti Jokowi Tanpa perlu bersuara keras tetap sanggup mencuri perhatian dunia dan diakui sebagai	sungguh bangga saya sebagai wni luar biasa indonesia bisa memiliki presiden seperti jokowi tanpa perlu bersuara keras tetap sanggup mencuri perhatian dunia dan diakui sebagai
Mantap Pak Prabowo ketegasan itu yang kami perlukan tentang Pernyataan saudara AM Hendropriyono yg bersifat RASIS dan Berpotensi untuk Mengadu Domba dan Memecah Belah antara Anak Bangsa	mantap pak prabowo ketegasan itu yang kami perlukan tentang pernyataan saudara am hendropriyono yg bersifat rasis dan berpotensi untuk mengadu domba dan memecah belah antara anak bangsa

### f. Pengambilan Target dan Anotasi Data/Labelling

Pada tahap ini akan dilakukan penarikan target atau aspek serta dilakukan anotasi atau pelabelan data. Metode yang digunakan adalah *Human Intelligence* Dimana target/aspek yang diambil merupakan sebuah kata benda dalam kalimat yang dapat berupa subyek atau obyek, dan target/aspek dapat terdiri atas lebih dari satu kata dimana dalam satu kalimat dapat terdiri atas lebih dari satu target/aspek. Label sentimen dibagi menjadi 3 yaitu positif,

negatif dan netral. Berikut merupakan contoh pelabelan yang dapat dilihat pada Tabel 3.2.

**Tabel 12.4 Contoh Pelabelan Data**

<b>Teks</b>	<b>Aspek</b>	<b>Keterangan Aspek</b>	<b>Label Sentimen Aspek</b>	<b>Justifikasi</b>
sungguh bangga saya sebagai wni luar biasa indonesia bisa memiliki presiden seperti jokowi tanpa perlu bersuara keras tetap sanggup mencuri perhatian dunia dan diakui sebagai	presiden seperti jokowi	Obyek	Positif	Aspek diambil dari obyek dalam kalimat yaitu 'presiden seperti jokowi' dimana obyek didapatkan dari kata kunci yang diambil. Sentimen Positif didapatkan berdasarkan kalimat sebelumnya yaitu 'luar biasa'
mantap pak prabowo ketegasan itu yang kami perlukan tentang pernyataan saudara am hendropriyono yg bersifat rasis dan berpotensi untuk mengadu domba dan memecah belah antara anak bangsa	pak prabowo	Subyek	Positif	Aspek diambil dari subyek dalam kalimat yaitu 'pak prabowo' dimana subyek didapatkan dari kata kunci yang diambil. Sentimen Positif didapatkan berdasarkan kalimat sebelumnya yaitu 'Mantap'
mantap pak prabowo ketegasan itu yang kami perlukan	pernyataan saudara am	Obyek	Negatif	Aspek diambil dari objek dalam kalimat yaitu 'pernyataan

tentang pernyataan saudara am hendropriyono yg bersifat rasis dan berpotensi untuk mengadu domba dan memecah belah antara anak bangsa	hendrop riyono			saudara am hendropriyono' yang merupakan objek yang dijelaskan oleh subyek 'pak prabowo'. Dimana subyek didapatkan dari kata kunci yang diambil. Sentimen Negatif didapatkan berdasarkan kalimat setelahnya yaitu 'rasis'
---	----------------	--	--	---

### g. Tokenizing

Pada tahap ini akan dilakukan proses *tokenizing*. *Tokenizing* adalah proses membuat token/pemisahan per kata dari sebuah korpus. Nantinya token-token yang telah terbentuk digunakan untuk merepresentasikan sebuah kata kedalam vektor menggunakan metode *word embedding*.

#### 3.1.4. Pelatihan Data

Pada tahap ini akan dilakukan proses pembuatan model dengan melakukan proses pelatihan pada model *word embedding* dengan *Word2Vec* dan model HPAN.

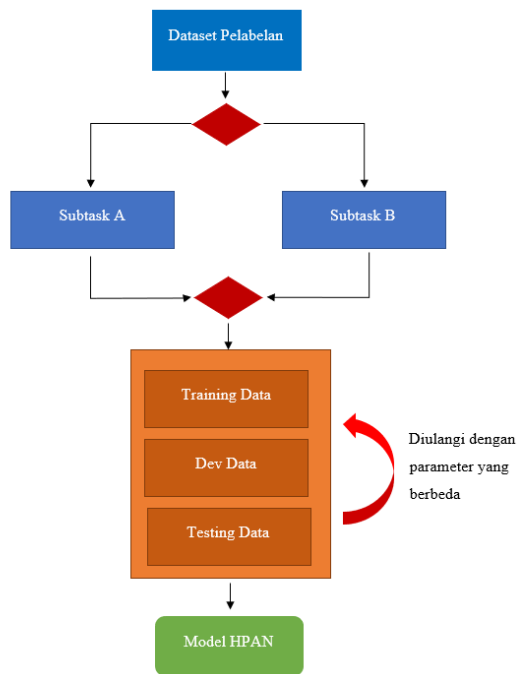
##### 3.1.4.1 Implementasi Model Word Embedding

Pada tahap ini akan dilakukan pengimplementasian *pretrained word embedding model* terhadap data yang telah melewati tahap prapemrosesan data. Implementasi bertujuan untuk mengubah setiap kata pada dataset percakapan menjadi nilai vektor agar dapat diproses dalam arsitektur HAPN. *Word embedding* dilakukan dengan memanfaatkan *library Gensim* yang diimplementasikan menggunakan Python.



### 3.1.4.2 Pembuatan dan Pelatihan Model HAPN

Pembuatan dan pelatihan model HAPN dibangun dengan *library tensorflow* dengan *backend keras* pada *python*. HAPN ini akan terdiri dari empat *layer* yaitu *input embedding*, *Bi-GRU Based Sentence encoder*, *Hierarchical Attention Based Fusion Layer*, *Output and model training*. Sama seperti dalam pembuatan model sebelumnya, pada proses ini juga akan dibuat beberapa skenario dengan perbedaan parameter, sehingga dapat diketahui performa dari proses *training*.



**Gambar 12.5** Alur Proses Model HAPN

Model HAPN yang paling baik akan digunakan untuk diujikan pada dataset yang tersedia, kemudian selanjutnya akan dilakukan proses analisis dan pengujian untuk melihat nilai akurasi dan error yang dimiliki.

### **3.1.5. Analisis dan Evaluasi Model**

Dari hasil proses *training* dengan beberapa skenario yang dibuat, akan dilakukan proses analisis terhadap model yang dihasilkan. Pada proses ini juga akan dilakukan evaluasi performa dari proses *training* yang dihasilkan. Analisis ini termasuk dengan pengujian terhadap kumpulan data baru, untuk mengukur sejauh mana model yang dihasilkan mampu melakukan klasifikasi teks dengan tepat.

### **3.1.6. Dokumentasi**

Pada tahapan terakhir ini akan dilakukan pembuatan laporan dalam bentuk buku tugas akhir yang disusun sesuai format yang telah ditentukan. Buku ini berisi dokumentasi langkah-langkah pengerjaan tugas akhir secara rinci. Buku ini diharapkan dapat bermanfaat sebagai referensi untuk pengerjaan penelitian lain, serta sebagai acuan untuk pengembangan lebih lanjut terhadap topik penelitian yang serupa.

## BAB IV PERANCANGAN

Bab ini akan menjelaskan mengenai rancangan dari luaran penelitian tugas akhir ini. Perancangan yang dibuat berupa rancangan *crawler*, rancangan pra-pemrosesan data, dan rancangan model.

### 4.1. Akuisisi Data Media Sosial

Tahap awal dari perancangan dimulai dengan mengumpulkan seluruh data yang dibutuhkan dari media sosial untuk tahap pengolahan data selanjutnya dalam analisis sentimen. Pengambilan data dilakukan melalui *crawler* yang dirancang untuk mengumpulkan data dari media sosial *twitter* yang berbahasa Indonesia dan bertopik politik. Berikut tabel entitas dan kata kunci yang digunakan dalam mengumpulkan data

**Tabel 13.1 Kata Kunci untuk *Crawling***

<b>Kategori</b>	<b>Entitas</b>	<b>Kata Kunci</b>
Partai Politik	Partai Kebangkitan Bangsa	pkb
	Partai Gerindra	gerindra
	Partai Demokrasi Indonesia Perjuangan	pdip
	Partai Golkar	golkar
	Partau Nasional Demokrat	nasdem
	Partai Garuda	partai garuda
	Partai Berkarya	partai berkarya
	Partai Keadilan Sejahtera	psk
	Partai Perindo	perindo
	Partai Persatuan Pembangunan	ppp
	Partai Solidaritas Indonesia	psi
	Partai Amanat Nasional	pan
	Partai Hanura	hanura
	Partai Demokrat	democrat
	Partai Bulan Bintang	partai bulan bintang
	Partai Keadilan dan Persatuan Indonesia	pkpi

<b>Kategori</b>	<b>Entitas</b>	<b>Kata Kunci</b>
Tokoh Politik	Joko Widodo	jokowi
	Prabowo Subianto	prabowo
	Maaruf Amin	maaruf amin
	Sandiaga Uno	sandiaga uno
	Anies Baswedan	anies
	Basuki Tjahaja Purnama	ahok
	Ganjar Pranowo	ganjar pranowo
	Ridwan Kamil	ridwan kamil
	Tri Rismaharini	bu risma
	Fahri Hamzah	fahri hamzah
	Fadli Zon	fadli zon
	Mahfud MD	mahfud md
Isu Politik	DPR	dpr
	Kabinet	kabinet
	Kementerian	kementerian
	KPK	kpk
	Pemilu	pemilu
	RAPBD	rapbd
	RKUHP	rkuhp

#### **4.2. Perancangan *Crawler***

Untuk mengambil data secara otomatis maka dirancang sebuah *crawler* yang akan mengambil *tweets* berdasarkan kata-kunci yang telah ditentukan dan menyimpan dalam sebuah *file*. *Crawler* yang digunakan akan memanfaatkan *library tweepy* dan API yang disediakan oleh *twitter*

#### **4.3. Desain *Database***

Untuk melakukan perancangan *crawler*, maka akan perlu dilakukan perancangan *database* untuk menyimpan data yang diambil dari media sosial *twitter*.

Data yang diambil dari *twitter* adalah data *tweet* berbahasa Indonesia sesuai dengan kata-kunci yang telah ditentukan. Tabel 4.2 menunjukkan desain *database* untuk menyimpan data posting dari *Twitter*.

**Tabel 13.2 Desain Database Crawler Post Twitter**

<b>Atribut</b>	<b>Tipe Data</b>	<b>Penjelasan</b>
id_tweet	<i>int</i>	Berisi <i>id</i> unik dari setiap <i>tweet</i>
topik	<i>varchar</i>	Berisi topik atau kata kunci dari <i>tweet</i> yang digunakan dalam <i>crawling</i>
text	<i>varchar</i>	Berisi pesan teks dari <i>tweet</i> yang di publikasikan
username	<i>varchar</i>	Berisi nama atau id user yang mempublikasikan <i>tweet</i>
user_id	<i>int</i>	Berisi <i>id</i> unik dari setiap <i>username</i>
created_time	<i>datetime</i>	Berisikan tanggal sebuah posting <i>twitter</i> diterbitkan

Sementara untuk analisis sentimen level aspek dibutuhkan informasi mengenai label sentimen dan aspek dari *tweet*. Tabel 4.3 menunjukkan desain *database* untuk menyimpan data bersih yang telah memiliki label sentimen.

**Tabel 13.3 Desain Database Sentimen**

<b>Atribut</b>	<b>Tipe Data</b>	<b>Penjelasan</b>
id_tweet	<i>int</i>	Berisi <i>id</i> unik dari setiap <i>tweet</i>
topik	<i>varchar</i>	Berisi topik atau kata kunci dari <i>tweet</i> yang digunakan dalam <i>crawling</i>
text	<i>varchar</i>	Berisi pesan teks yang sudah melauhi pra-pemrosesan data dari <i>tweet</i> yang di publikasikan
label	<i>varchar</i>	Berisi label dari setiap <i>tweet</i> yang diambil
aspek	<i>varchar</i>	Berisi aspek dari setiap <i>tweet</i> yang diambil

#### **4.4. Perancangan Pra-Pemrosesan Data**

Data yang ada harus terlebih dahulu mengalami proses persiapan data atau pra-pemrosesan. Pra-pemrosesan merupakan proses untuk melakukan transformasi data mentah menjadi suatu format yang lebih mudah dan efektif dan diolah dengan baik saat pembuatan model pada langkah selanjutnya. Berikut ini merupakan tahapan yang dilakukan pada pemrosesan data berikut.

##### **4.4.1. Perancangan Penggabungan Dataset**

*Dataset* yang dikumpulkan akan digabungkan menjadi satu *database* dikarenakan pengumpulan *dataset* dilakukan di waktu yang berbeda.

##### **4.4.2. Perancangan Penghapusan Data yang Terduplikasi**

Data yang didapatkan dari proses sebelumnya masih terdapat data yang sama, sehingga data perlu dihapus salah satunya agar setiap data yang ada merupakan data yang unik. Untuk melakukan hal ini digunakan fungsi *drop duplicates* pada *library pandas*

##### **4.4.3. Perancangan *Filtering* Bahasa Indonesia**

Data *tweet* yang telah dikumpulkan terkandung yang mengandung bahasa selain Bahasa Indonesia sehingga diperlukan filter terhadap *dataset* agar keseluruhan data menggunakan Bahasa Indonesia. Tahap ini memerlukan *library langdetect* dalam *python* yang efektif untuk melakukan *filtering* bahasa. Setiap kalimat yang akan diseleksi akan memiliki indeks khusus yang merepresentasikan kandungan bahasanya. Dalam penelitian ini indeks yang digunakan untuk Bahasa Indonesia memiliki nilai kepercayaan lebih dari 0,9.

##### **4.4.4. Perancangan Pembersihan dataset**

Dalam *dataset* yang telah dikumpulkan akan ditemui banyak symbol dan karakter yang kurang berguna dalam proses pelatihan data. Sehingga diperlukan aktivitas untuk membersihkan *tweets* dari karakter yang tidak diperlukan.

Selain itu juga dilakukan pembersihan terkait *hashtag* dan akun *mention* dalam *tweets*, hal ini dilakukan untuk penyederhanaan *tweets* agar proses pelatihan data lebih efektif. Untuk lebih lengkapnya berikut urutan proses pembersihan *tweets*.

1. Menghilangkan URL yang terkandung dalam *tweets*
2. Menghilangkan atribut HTML yang terkandung dalam *tweets*
3. Menghilangkan akun yang di-*mention* yang terkandung dalam *tweets*
4. Menghilangkan *hashtag* yang di *mention* yang terkandung dalam *tweets*
5. Menghapus seluruh simbol dan tanda baca yang terdapat pada *tweets*
6. Menghapus huruf berulang dalam satu kata.
7. Merubah *tweets* menjadi huruf kecil

#### **4.5. Perancangan Pemberian Label Data**

*Dataset* yang didapatkan dari *twitter* hanya berisi satu hanya berisi data text yang berisi *tweets* atau opini terhadap topik politik. Data tersebut membutuhkan label saat proses pelatihan model. Sehingga data harus diberikan label secara manual. Label yang diberikan pada setiap *tweet* bertujuan untuk memberikan pendapat sentimen yang sesuai terhadap suatu aspek pada *tweet*, sehingga label tersebut dapat dipelajari oleh model pada saat pelatihan data. Label yang akan di berikan akan dibagi menjadi 2 pengelompokan yaitu:

1. Aspek

Aspek yang akan diterapkan pada penelitian ini menyesuaikan pada aspek kata yang memiliki sentimen pada kalimat. Sehingga jumlah aspek tidak dapat ditentukan di awal pengerjaan pemberian label pada dataset. Aspek yang diambil merupakan sebuah subjek

politik atau objek dalam sebuah kalimat. Contoh pelabelan aspek pada teks dapat dilihat pada Tabel 4.4.

**Tabel 13.4 Label Aspek**

<b>Teks</b>	<b>Aspek</b>
sungguh bangga saya sebagai wni luar biasa indonesia bisa memiliki presiden seperti jokowi tanpa perlu bersuara keras tetap sanggup mencuri perhatian dunia dan diakui sebagai	presiden seperti jokowi
mantap pak prabowo ketegasan itu yang kami perlukan tentang pernyataan saudara am hendropriyono yg bersifat rasis dan berpotensi untuk mengadu domba dan memecah belah antara anak bangsa	pak prabowo
	pernyataan saudara am hendropriyono

## 2. Sentimen

Pada penelitian ini terdapat terdapat 3 jenis label sentimen yang akan diberikan. Pada Tabel 4.5 yang akan menjelaskan mengenai label sentimen.

**Tabel 13.5 Label Sentimen**

<b>Sentimen</b>	<b>Penjelasan</b>
Negatif	Sentimen negatif adalah pendapat yang memiliki makna yang buruk terhadap aspek
Netral	Sentimen netral adalah pendapat yang tidak memiliki makna yang baik maupun buruk terhadap aspek
Positif	Sentimen positif adalah pendapat yang memiliki makna yang baik terhadap aspek



#### 4.6. Perancangan Dataset Model HAPN

*Dataset* yang digunakan dalam HAPN ini terdapat 2 buah, menyesuaikan dengan subtask yang dijelaskan sebelumnya. *Dataset* ini memiliki *tweets* yang sama hanya berbeda untuk perlakuan labelnya. Berikut kondisi label untuk setiap subtask

**Tabel 13.6 Perlakuan Label**

Subtask	Sentimen	Perlakuan
A	Positif, Netral, Negatif	Tidak ada perlakuan
B	Positif, Negatif	Menghapus data yang memiliki label netral

Selain itu juga dilakukan pembagian dataset menjadi train set, dev set, dan test set. Pembagian ini dilakukan dengan rasio 80:20 untuk data train-dev dan data test. Kemudian pada data train-dev akan dilakukan *5-fold cross-validation*, Sehingga data dev atau pengujian yang dilakukan merupakan seluruh data pada train-dev yang terbagi menjadi 5 bagian.

#### 4.7. Perancangan Implementasi Model *Word Embedding*

Kamus *word vector* dari seluruh kata pada *dataset* dibuat berdasarkan *Pre-trained word embedding model* yang telah diperoleh dari penelitian sebelumnya. Tetapi tidak semua kata dari dataset diketahui oleh model *word embedding*. Sehingga nilai vektor secara *random* akan diberikan kepada kata yang tidak diketahui. Kamus *word vector* ini akan dijadikan referensi *initial weights* untuk setiap input kata dalam pembuatan model HAPN.

#### 4.8. Perancangan Model *Hierarchical Attention Position-aware Network*

Analisis sentimen level aspek menggunakan metode HAPN akan dilakukan dengan *library keras* pada *python*. *Library* ini memiliki keunggulan yaitu dapat menjalankan proses *training*

menggunakan GPU, sehingga durasi *training* dapat menjadi lebih cepat. Input utama yang dibutuhkan untuk menjalankan model yaitu data *pretrained word embedding* serta *position embedding* yang akan dibuat serta *dataset train*.

Dalam pembuatan model *Hierarchical Attention Position-aware Network* terdapat beberapa parameter yang harus ditentukan agar mendapatkan hasil terbaik. Perubahan parameter yang akan dilakukan yaitu perubahan pada model *neural network*, *optimizer*, *learning rate* serta pada nilai *dropout*. Skenario pertama yang dilakukan dalam proses pelatihan pada penelitian ini yaitu melakukan perubahan model *Neural Network* yang terdiri atas *SimpleRNN*, *LSTM* dan *GRU*. Sementara untuk baseline parameter lain model yaitu *Optimizer* *Adam*, *Learning rate* 0.001, dan *dropout* 0.05. Dari hasil setiap scenario pada model NN akan didapatkan model NN terbaik dalam setiap subtask yang akan dijadikan dasar atau baseline dalam melakukan perubahan parameter selanjutnya. Skenario selanjutnya dilakukan dengan mengubah nilai *optimizer*, kemudian dilanjutkan dengan mengubah nilai *learning rate* dan yang terakhir mengubah nilai *dropout*. Skenario terakhir merupakan penambahan *character embedding* pada pembangunan model HAPN. Proses *training* dari model HAPN dilakukan sebanyak 30 epoch dengan komposisi parameter yang berbeda. Berikut adalah nilai-nilai parameter yang akan digunakan dalam proses *training* HAPN :

**Tabel 13.7 Parameter yang digunakan dalam Model HAPN**

Parameter	Nilai	Keterangan
<b>Model</b>	1. SimpleRNN 2. LSTM 3. GRU	Model <i>neural network</i> yang digunakan dalam membangun arsitektur saat <i>training</i> model
<b>Optimizer</b>	1. Adam 2. Adadelta 3. SGD 4. RMSprop	Menentukan algoritma <i>optimizer</i> yang digunakan untuk <i>training</i> model. <i>Optimizer</i> adalah algoritma atau metode yang digunakan untuk mengubah atribut

Parameter	Nilai	Keterangan
		model neural network seperti <i>weights</i> dan <i>learning rate</i> untuk mengurangi <i>loss</i>
<b><i>Learning Rate</i></b>	<ol style="list-style-type: none"> <li>1. 0.001</li> <li>2. 0.002</li> <li>3. 0.005</li> <li>4. 0.01</li> </ol>	Menentukan nilai <i>Learning rate</i> dari <i>Optimizer</i> yang dipilih. <i>Learning Rate</i> merupakan parameter yang mengontrol berapa banyak untuk mengubah model dalam menanggapi estimasi kesalahan setiap kali bobot model diperbarui.
<b><i>Dropout</i></b>	<ol style="list-style-type: none"> <li>1. 0</li> <li>2. 0.1</li> <li>3. 0.2</li> <li>4. 0.3</li> <li>5. 0.4</li> <li>6. 0.5</li> </ol>	Nilai indeks untuk proses <i>dropout</i> saat <i>training model</i> . Dimana <i>dropout</i> adalah teknik regularisasi untuk mengurangi <i>overfitting</i> mencegah ke-kompleks-an pada data <i>training</i> .
<b><i>Character Embedding</i></b>	<ol style="list-style-type: none"> <li>1. HAPN tanpa <i>character embedding</i></li> <li>2. HAPN + <i>character embedding</i></li> </ol>	Penggunaan karakter dalam <i>word embedding</i> yang diharapkan dapat meningkatkan kinerja model.

#### 4.9. Perancangan Evaluasi Pengukuran

Tahap ini dilakukan untuk menguji dan mengevaluasi kualitas model *Hierarchical Attention Position-aware Network*. Setiap model dari hasil percobaan training seluruh skenario akan diuji pada dev set dan test set. Untuk melakukan pengukuran performa setiap model, digunakan tiga metrik berbeda yaitu *accuracy*, *average recall* dan *f1-score* untuk setiap *subtask*. Untuk metrik *average recall* dan *f1-score* akan dilakukan perhitungan “*Macroaveraged*”. *Macroaveraged* dipilih dikarenakan pembobotan perhitungan yang dilakukan pada proses evaluasi diperhitungkan berdasarkan *class* dan akan sama untuk setiap *class*. Sehingga lebih baik dari perhitungan

menggunakan *accuracy* di mana bobot yang diberikan akan sama untuk setiap keputusan klasifikasi dalam setiap data tanpa memperhitungkan ketidakseimbangan perhitungan dalam jumlah data setiap *class*.

Berikut adalah evaluasi pengukuran yang digunakan dalam penelitian ini.

- *Accuracy*

*Accuracy* adalah perhitungan yang mengukur tingkat kedekatan antara nilai prediksi dengan nilai aktual.

Rumus *Accuracy*

$$Accuracy = \frac{\text{Jumlah prediksi benar}}{\text{Total Data}}$$

- *Average Recall (AvgRec)*

*Average recall* adalah perhitungan yang mengukur tingkat keberhasilan suatu sistem dalam menentukan nilai kebenaran dari suatu label.

Rumus *Average Recall*

$$AvgRec = \frac{1}{C} (R^P + R^N + R^U)$$

Dimana :

C : Jumlah label di dalam *class*

$R^P$  : Recall terhadap data positif

$R^N$  : Recall terhadap data negatif

$R^U$  : Recall terhadap data netral

Kelebihan menggunakan *AvgRec* dibandingkan dengan “standar” akurasi adalah lebih baik dalam proses yang melibatkan *class imbalance*.

- *F1-Score*

*F1-Score* merupakan pengukuran mempertimbangkan nilai baik dari presisi maupun dari recall untuk menghitung skor. Pada penelitian ini *f1-score* yang digunakan adalah rata-rata penggabungan dari *precision* dan *recall* dari kelas positif dan negatif (tidak menggunakan kelas netral).

Rumus *F1-Score*

$$F_1^{PN} = \frac{1}{2} (F_1^P + F_1^N)$$

Dimana :

$F_1^P$  : Nilai F1-Score pada kelas positif

$F_1^N$  : Nilai F1-Score pada kelas negatif

*Halaman ini sengaja dikosongkan*

## BAB V IMPLEMENTASI

Bab ini menjelaskan mengenai implementasi dari perancangan yang telah dilakukan sesuai dengan metode pengembangan yang dibuat. Bagian implementasi akan menjelaskan mengenai lingkungan implementasi, pembuatan fitur-fitur aplikasi dalam bentuk kode, serta pengujian aplikasi.

### 5.1. Persiapan Implementasi

Penelitian ini menggunakan perangkat keras dengan spesifikasi yang dapat dilihat pada Tabel 5.1.

**Tabel 14.1 Spesifikasi Perangkat Keras Laptop**

<b>Nama Perangkat</b>	Laptop
<b>Processor</b>	Intel® Core™ i7-8550U CPU @ 1.80GHz 1.99 GHz
<b>Memory</b>	8192 MB RAM
<b>GPU</b>	Nvidia GeForce MX130
<b>Sistem Operasi</b>	Windows 10 Home
<b>Arsitektur Sistem</b>	64-bit <i>Operating System, x64-based processor</i>

Aplikasi dikembangkan dengan menggunakan beberapa teknologi seperti code editor, database, bahasa pemrograman, dan *library* yang disajikan dalam Tabel 5.2.

**Tabel 14.2. Daftar Library**

<b>Bahasa Pemrograman</b>	Python
<b>Computational Environment</b>	Jupyter Notebook
<b>Virtual Environment</b>	Anaconda
<b>Library</b>	<ul style="list-style-type: none"><li>• Tweepy</li><li>• Gensim</li><li>• Pandas</li><li>• Numpy</li><li>• Sklearn</li><li>• Keras</li></ul>

	<ul style="list-style-type: none"> <li>• Tensorflow</li> <li>• Matplotlib</li> <li>• Seaborn</li> <li>• langdetect</li> </ul>
--	---

## 5.2. Pembuatan *Crawler Twitter*

Dalam pembuatan *crawler Twitter* digunakan *library Tweepy* yang dikhususkan untuk melakukan proses *crawling* pada *social media Twitter*. *Crawler* yang dibuat akan melakukan pengambilan data dari twitter dengan menggunakan *keyword*. *Tweepy* merupakan *library* yang memiliki performa cukup bagus untuk mendapatkan *tweets* dengan jumlah yang banyak dengan memaksimalkan limit yang disediakan oleh API *twitter*. Dimana *twitter* API hanya memperbolehkan mengakses posting *twitter* hingga satu minggu sebelum hari pengambilan data. *Library tweepy* akan mengambil data paling baru sesuai *keyword* yang digunakan hingga data *tweet* yang diposting satu minggu sebelumnya sesuai dengan ketersediaan data *twitter*.

```
import tweepy
import sys

auth = tweepy.AppAuthHandler('API_KEY', 'API_SECRET')
api = tweepy.API(auth,
wait_on_rate_limit=True, wait_on_rate_limit_notify=True)
if (not api):
    print("Can't Authenticate")
    sys.exit(-1)
```

### Kode 14.1 Potongan Kode Implementasi API untuk *Crawling*

Kode 5.1 diatas merupakan bagian untuk melakukan autentifikasi dan menghubungkan *client* dengan *twitter* API agar dapat melakukan *crawling*. Proses ini membutuhkan autentifikasi dengan API\_KEY dan API\_SECRET yang didapat



dari website *twitter*. Kode diatas juga untuk memastikan autentikasi berhasil dan berhasil terhubung dengan API *Twitter*.

```
keyword= 'jokowi'
searchQuery = keyword+' -filter:retweets -
filter:replies'
tweetsPerQry = 100

sinceId = None
max_id = 9999999999999999999
date_since = '2019-10-21'
tweetCount = 0
```

#### Kode 14.2 Potongan Kode Parameter Proses Crawling

Kode 5.2 menjelaskan variabel yang terdapat pada *crawler* ini. Variabel *keyword* berisi kata kata yang merupakan kata kunci dalam proses *crawling*, sehingga semua *tweets* yang mengandung kata tersebut akan diambil oleh *crawler* ini. Jumlah *tweets* maksimal yang diperbolehkan untuk diambil adalah 100 *tweet* per eksekusi API ( bukan 1 proses *crawling* ). Variabel *sinceId* menyimpan informasi id *tweet* minimal yang dapat diambil oleh *crawler*. *Max\_id* menyimpan informasi id *tweet* maksimal yang dapat diambil oleh *crawler*. Untuk memaksimalkan proses *crawling* *sinceId* diisi dengan *none* dan *max\_id* diisi dengan 9999999999999999999.

```

columns=['user_id','name','username','id_tweet','text',
'created_time']
df = pd.DataFrame(columns=columns)

while True:
    try:
        if (max_id <= 0):
            if (not sinceId):
                new_tweets = api.search(q=searchQuery,
count=tweetsPerQry, since=date_since,
tweet_mode='extended')
            else:
                new_tweets = api.search(q=searchQuery,
count=tweetsPerQry, since=date_since, since_id=sinceId,
tweet_mode='extended')
            else:
                new_tweets = api.search(q=searchQuery,
count=tweetsPerQry, since=date_since,max_id=str(max_id
- 1),since_id=sinceId, tweet_mode='extended')
            if not new_tweets:
                print("No more tweets found")
                break
            tweets = new_tweets
            users_locs = [[tweet.user.id,tweet.user.name,
tweet.user.screen_name, tweet.id ,tweet.full_text,
tweet.created_at] for tweet in tweets]
            tweet_text = pd.DataFrame(data=users_locs,
columns=['user_id','name','username','id_tweet','text',
'created_time'])
            tweet_text =
tweet_text.replace(to_replace=[r"\\t|\\n|\\r",
"\t|\n|\r"], value=["", ""], regex=True)
            df = pd.concat([df, tweet_text], sort=False)
            df['topik'] = keyword
            tweetCount += len(new_tweets)
            max_id = new_tweets[-1].id
        except tweepy.TweepError as e:
            # Just exit if any error
            print("some error : " + str(e))

df = df.reset_index(drop=True)
df.to_csv(keyword+date_since+'.csv')

```

**Kode 14.3 Potongan Kode Perulangan Crawling**

Kode 5.3 menunjukkan *script* yang digunakan untuk melakukan perulangan dalam pengambilan data *twitter* sesuai dengan *keyword* pencarian yang ada dengan menggunakan method `api.search()`. Proses ini akan berulang hingga hasil yang didapatkan dari pengambilan data *twitter* tidak menemukan *tweet* lagi. Kemudian pada proses ini juga menghasilkan sebuah *file csv* yang digunakan untuk menyimpan data *twitter* sesuai dengan *keyword*.

### 5.3. Pra-Pemrosesan Dataset

#### 5.3.1. Penggabungan Dataset

Data yang didapat dari hasil *crawling* yang terpisah akan digabungkan dalam satu *file csv*.

```
import os
import pandas as pd

path = 'politik'
text_files = [f for f in os.listdir(path) if
f.endswith('.csv')]
combined_data = pd.concat([pd.read_csv('politik/'+f)
for f in text_files ])
combined_data = combined_data.reset_index()
combined_data.to_csv( "full_data_politik.csv",
index=False)
```

#### Kode 14.4 Potongan Kode untuk Pnggabungan Dataset

Kode 5.4 menjelaskan proses penggabungan *dataset* yang diambil dari hasil file csv proses *crawler*. Dari file csv tiap *keyword* yang telah disimpan dalam sebuah folder/path dengan nama politik, akan digabungkan dalam sebuah *dataframe* pandas dengan menggunakan fungsi `concat` yang akan membaca seluruh *file csv* dalam *path* politik tersebut.

### 5.3.2. Penghapusan Data yang Terduplikasi

Proses yang di lakukan untuk menghapus data *tweet* yang terduplikasi, dilakukan dengan menggunakan bantuan *library pandas*.

```
import pandas as pd

data = pd.read_csv("full_data_politik.csv")
data =
data.reset_index().drop_duplicates(subset='id_tweet',
keep='first')
data = data.drop(['index'] , axis =1)
data =
data.reset_index().drop_duplicates(subset='text',
keep='first')
data = data.drop(['index'] , axis =1)

data.to_csv('data_politik_non_duplicates.csv')
```

#### Kode 14.5 Potongan Kode untuk Menghapus Kata Duplikat

Pada Kode 5.5 menjelaskan proses menggunakan *method drop duplicates* pada *library pandas* yang memastikan bahwa setiap *tweet* tidak ada yang terduplikasi. Proses ini dilakukan dua kali dengan melihat atribut *id\_tweet* dan *text* atau pesan pada *tweet*. Kemudian data disimpan menjadi sebuah *file* baru.

### 5.3.3. Filtering Bahasa

Tahap ini memiliki tujuan untuk menghapus *tweet* yang bukan bahasa indonesia. Dalam penerapannya menggunakan *library* berbasis *python* yaitu *langdetect*.

```

from langdetect import detect_langs
import pandas as pd

data = pd.read_csv("data_politik_non_duplicates.csv")
for i, row in data.iterrows():
    try:
        language = str(detect_langs(row))
        lang = language[1:3]
        index = float(language[4:16])
    except:
        language = "error"
        data.at[i, 'lang'] = language
        data.at[i, 'lang1'] = lang
        data.at[i, 'langindex'] = index

data_id = data[(data['lang1'] == 'id' ) &
               (data['langindex'] > 0.9)]
data_id.to_csv('data_id.csv')

```

#### Kode 14.6 Potongan Kode Untuk Mendeteksi Bahasa

Pada Kode 5.6 di atas menjelaskan proses mendeteksi bahasa yang digunakan dalam *tweets*. *Method detect\_langs()* dari *library langdetect* berguna untuk mendeteksi bahasa dan mengembalikan dua variabel yaitu bahasa yang terdeteksi dan indeks kepercayaan kandungan bahasa tersebut. *Tweet* yang diambil merupakan tweet dengan bahasa = 'id' dan nilai indeks lebih dari 0.9. Semua *tweet* yang dikategorikan bahasa Indonesia kemudian disimpan lagi kedalam file baru dalam format csv.

### 5.3.4. Pembuatan Pembersihan Dataset

```

import pandas as pd
import re

def replace_mention(string):
    tokens = ['' if token.startswith('@') else token
    for token in string.split()]
    return ''.join(tokens)

def replace_URL(string):
    tokens = ['' if 'http' in token else token
    for token in string.split()]
    return ''.join(tokens)

def replace_mult_occurences(string):
    return re.sub(r'(\.){2,}', r'\1', string)

def replace_hastag(string):
    tokens = ['' if token.startswith('#') else token
    for token in string.split()]
    return ''.join(tokens)

def preprocess(data, text):
    data['text2'] = data[text].apply(replace_mention)
    data['text2'] = data['text2'].apply(replace_URL)
    data['text2'] =
data['text2'].apply(replace_mult_occurences)
    data['text2'] = data['text2'].apply(replace_hastag)
    data['text2'] =
data['text2'].str.replace('[^\w\s@#/:%.,-]', '',
flags=re.UNICODE)
    data['text2'] = data['text2'].str.lower()
    return data

data_bersih = preprocess(data, 'text')

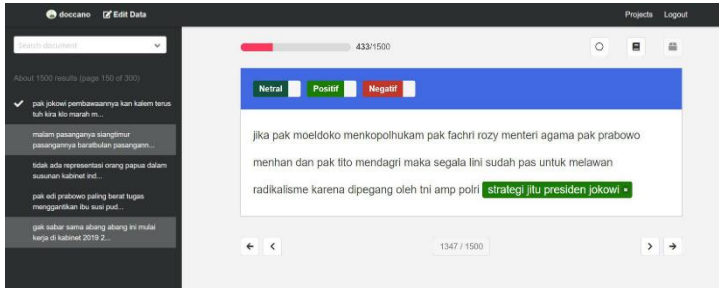
```

**Kode 14.7 Potongan Kode Proses Pembersihan Dataset**

Kode 5.7 menjelaskan tentang proses pembersihan dataset yang menggunakan *library re (regular expression)* dan *pandas*. Dibuat fungsi preprocess untuk proses pembersihan data yang berisi urutan-urutan yang dilakukan. Pertama yang dilakukan memanggil fungsi `replace_mention` dimana fungsi ini akan menghapus mention yang ada pada data *tweet*. Yang kedua merupakan pemanggilan fungsi `replace_URL` yang akan menghapus URL yang ada pada data *tweet*. Kemudian dilanjutkan dengan memanggil fungsi `replace_mult_occurences` yang akan menghapus huruf berulang. Karakter yang berulang lebih dari dua pada ujaran diganti/ *substitute* dengan dua karakter saja. Kemudian dilanjutkan dengan memanggil fungsi `replace_hashtag` yang akan menghapus hashtag. Setelah itu dilakukan penghapusan tanda baca dan simbol pada setiap *tweet*. Proses ini dilakukan agar kamus kata tidak berisi kata-kata yang sama dengan tanda baca berbeda. Untuk melakukannya, dibuat fungsi yang memanfaatkan *library regular expression*. Yang terakhir dilakukan perubahan kata menjadi *lowercase* atau yang disebut *casefolding*. Proses ini memanfaatkan fungsi `lower` dari *pandas*, sehingga seluruh *tweet* dapat diubah menjadi *lowercase*.

#### **5.4. Pembuatan Pelabelan Data**

Pelabelan Data dilakukan dengan menggunakan aplikasi *open source* yaitu *doccano* yang merupakan aplikasi untuk melakukan pelabelan untuk data *Sequence Labelling*. Aplikasi anotasi atau labelling ini memiliki tampilan yang dapat dilihat pada Gambar 5.1.



**Gambar 14.1 Aplikasi Pelabelan Data**

Pada Aplikasi ini anotator akan memilih aspek apa saja yang ada dalam tweet, kemudian memberikan sentimen positif, negatif atau netral terhadap aspek tersebut.

### 5.5. Pembuatan Model *Hierarchical Attention Position Aware Network*

Kode program yang dibuat untuk membuat model *Hierarchical Attention Position Aware Network* menggunakan *library Keras*. *Library* tersebut akan memanfaatkan kemampuan komputasi arsitektur cuda pada GPU untuk melakukan komputasi secara paralel.

Kumpulan data yang digunakan untuk model HAPN adalah data tweet berlabel luaran tahap anotasi data. Untuk menyiapkan data pembuatan model, kumpulan data *tweet* berlabel pada basis data aplikasi anotasi, diubah dan diekspor ke dalam bentuk file csv.

Model HAPN memiliki beberapa struktur untuk membangunnya. Langkah pertama merupakan adalah membuat fungsi untuk *position embedding*. *Position embedding* ini bertujuan untuk mengetahui posisi kata dalam kalimat dan melakukan pembobotan untuk tiap posisi kata dengan membuat vector untuk tiap posisi yang berbeda. Pembuatan *position embedding* akan dijelaskan pada kode berikut :



```

def get_position_input(sentences=[], aspects=[]):
    positions = []
    sentences_length = []
    for sentence, aspect in zip(sentences, aspects):
        aspect = ','.join(str(e) for e in aspect)
        aspect = aspect.replace(',', ' ')
    .....

def get_position_ids(max_len=82):
    position_ids = {}
    position = (max_len - 1) * -1
    position_id = 1
    while position <= max_len - 1:
        position_ids[position] = position_id
    .....

def convert_position(positions, position_ids={}):
    position_inputs = positions
    for line in position_inputs:
        for index, position in enumerate(line):
            line[index] = position_ids[position]
    return position_inputs

def write_position_embedding(vec_size=20, max_len=100):
    length = max_len * 2
    np.random.seed(1000)
    position_matrix = np.random.rand(length, vec_size)
    return np.asarray(position_matrix)

```

#### Kode 14.8 Potongan Kode Pembuatan *Position Embedding*

Kode 5.8 menjelaskan fungsi-fungsi yang dibuat untuk menghasilkan *position embedding*. Fungsi pertama adalah fungsi `get_position_input()` yang merupakan fungsi untuk mencari posisi aspek dalam kalimat. Fungsi kedua `get_position_ids()` merupakan fungsi untuk membuat sebuah array yang berisi urutan posisi mulai dari pertama sampai maksimal panjang kata pada kalimat. Fungsi ketiga `convert_position()` yang merupakan fungsi untuk merubah position dengan position array yang dihasilkan di fungsi sebelumnya. Fungsi terakhir adalah fungsi `write_position_embedding()` yang merupakan fungsi untuk membuat matrix yang berisi *random vector* untuk tiap posisi.

```

def convert_embedding_file(word_num=10000,
word_index={}):
    embedding_file = "../resource/modelapik.bin"
    typeToLoad = "word2vec"
    if(typeToLoad=="word2vec"):
        word2vecDict =
KeyedVectors.load_word2vec_format(embedding_file,
binary=True)
        embed_size = 300
        embeddings_index = dict()
        for word in word2vecDict.wv.vocab:
            embeddings_index[word] =
word2vecDict.word_vec(word)

        all_embs =np.stack(list(embeddings_index.values()))
        emb_mean,emb_std = all_embs.mean(), all_embs.std()
        nb_words = len(word_index)+1
        np.random.seed(1000)
        embedding_matrix = np.random.normal(emb_mean,
emb_std, (nb_words, embed_size))
        embeddedCount = 0
        for word, i in word_index.items():
            i-=1
            embedding_vector = embeddings_index.get(word)
            if embedding_vector is not None:
                embedding_matrix[i] = embedding_vector
                embeddedCount+=1
        print('total embedded:',embeddedCount,'common
words')
        return embedding_matrix

```

#### Kode 14.9 Potongan Kode Implementasi *Word Embedding*

Langkah selanjutnya yaitu mengimplementasikan *word embedding* dikarenakan model *neural network* hanya dapat memproses input berupa angka atau vektor. Sehingga kalimat tweet yang ada perlu diubah terlebih dahulu menjadi vector sehingga akan terdapat kamus *word vector*. Kamus word vector

merupakan kumpulan token kata dalam dataset beserta nilai vector tiap katanya. Kamus *word vector word2vec* yang telah dibuat pada penelitian sebelumnya akan digunakan dalam pembuatan model HAPN. Kode 5.9 menjelaskan fungsi-fungsi untuk mengimplementasikan kamus yang telah dibuat kedalam model HAPN.

Kode 5.9 merupakan penjelasan fungsi `convert_embedding_file()` dimana fungsi ini membutuhkan input berupa `word_num` atau maksimal jumlah kata dan `word_index` yang merupakan kamus token kata unik dari dataset yang akan memiliki indeks unik mulai dari 1 dan seterusnya. Hal pertama yang dilakukan adalah membaca *word embedding word2vec* yang telah dibuat pada Penelitian sebelumnya yang memiliki nama file 'modelapik.bin' dengan fungsi `KeyedVectors.load_word2vec_format()` dan menyimpannya nilai vektornya dalam matrix bernama `embedding_index()`. Kemudian dibuat sebuah matriks `word_embedding` dengan dimensi 300 dan jumlah token unik + 1, dimana isi dari matrix tersebut merupakan nilai random berdasarkan rata-rata dan standar deviasi dari *word embedding word2vec* sebelumnya. Kemudian akan dilakukan pengisian nilai vector dari kata pada `word index` sesuai dengan nilai pada kamus *word embedding word2vec* untuk setiap kata yang terdapat pada kamus tersebut. Sehingga jika kata tidak berada pada kamus nilai vektornya akan tetap random sesuai dengan sebelumnya atau tidak berubah.

Untuk skenario ke 5 akan dilakukan penggunaan *character embedding*. Dimana pembangunan dan pelatihan *character embedding* akan dijelaskan pada Kode 5.10 berikut :

```

class CharCNN:

    CHAR_DICT =
    'abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ01
    23456789 .!?:,\'%-\(\)/$|&>[]"'

    def __init__(self, max_len_of_sentence,
max_num_of_setnence, verbose=10):
        self.num_of_char = 0
        self.num_of_label = 0

    def _build_character_block(...):
        for i in range(len(filters)):
            block = Conv1D(
                filters=filters[i],
kernel_size=kernel_size[i],
                padding=padding, activation=activation,
kernel_initializer=kernel_initializer)(block)

            ...

    def _build_sentence_block(...):
        sent_input = Input(shape=(max_len_of_sentence,
), dtype='int64')
        embedded = Embedding(self.num_of_char,
char_dimension,
input_length=max_len_of_sentence)(sent_input)
        print(embedded)

        ...

    def _build_document_block():
        doc_input = Input(shape=(max_num_of_setnence,
max_len_of_sentence), dtype='int64')
        doc_output =
TimeDistributed(sent_encoder)(doc_input)
        doc_output = Bidirectional(LSTM(128,
return_sequences=False, dropout=dropout))(doc_output)

        ...

```

```

def preprocess(.....):
    self.build_char_dictionary(char_dict,
unknown_label)
    self.convert_labels(labels)

def process(.....):
    .....
    x_preprocess, y_preprocess =
self._transform_raw_data(
        df=df, x_col=x_col, y_col=y_col,
label2indexes=label2indexes)
    .....

def build_model(.....):

    sent_encoder = self._build_sentence_block(..)
    doc_encoder = self._build_document_block(..)
    .....

def train(.....):
    self.get_model().fit(
        x_train, y_train, validation_data=(x_test,
y_test),
        batch_size=batch_size, epochs=epochs,
shuffle=shuffle)

def predict(self, x, return_prob=False):
    return
self.get_model().predict(x_test).argmax(axis=-1)

def get_model(self):
    return self.model['doc_encoder']

```

**Kode 14.10 Potongan Kode Pembuatan *Character Embedding***

Kode 5.10 menjelaskan Class untuk membuat dan melatih *character embedding*. Dimana kode dimulai dengan penerapan fungsi `preprocess()` untuk menyiapkan informasi kamus karakter dan mengubah label dari teks ke numerik. Kemudian dilanjutkan dengan memanggil fungsi `process()` untuk mengubah data pelatihan dan pengujian ke format numpy untuk input keras. Kemudian dilakukan pemanggilan fungsi `build_model()` untuk membangun model CNN untuk *Character Embedding* dengan menggunakan 3 1D lapisan CNN dengan fungsi `_build_charcater_block()` dan `_build_sentence_block()`. Input CNN ini kemudian akan dimasukkan kedalam layer LSTM pada fungsi `_build_document_block()` sebelum dilakukan proses pelatihan. *Weighted Model* dari proses pelatihan kemudian disimpan dan dipanggil untuk dilakukan proses prediksi menggunakan fungsi `predict()` dan `get_model()` untuk mendapatkan kamus *character embedding*.

Setelah itu langkah berikutnya merupakan proses pembangunan model HAPN yang akan dijelaskan pada Kode 5.11 berikut :

```

def build_model(...):

    max_len = max_len
    sentence_input = Input(shape=(max_len,),
dtype='int32', name='sentence_input')
    position_input = Input(shape=(max_len,),
dtype='int32', name='position_input')
    aspect_input = Input(shape=(aspect_max_len,),
dtype='int32', name='aspect_input')

    sentence_embedding_layer = Embedding(num_words + 1,
EMBEDDING_DIM, weights=[embedding_matrix],
input_length=max_len, trainable=True, mask_zero=True)
    sentence_embedding =
sentence_embedding_layer(sentence_input)
    position_embedding = Embedding(max_len * 2,
POSITION_EMBEDDING_DIM,
weights=[position_embedding_matrix], input_length=max_le
n, trainable=True, mask_zero=True)(position_input)
    aspect_embedding_layer = Embedding(num_words + 1,
EMBEDDING_DIM, weights=[embedding_matrix],
input_length=aspect_max_len, trainable=True,
mask_zero=True)
    aspect_embedding =
aspect_embedding_layer(aspect_input)
    input_embedding =
keras.layers.concatenate([sentence_embedding,
position_embedding, char_embedding])
    .....
    model.compile(loss=['categorical_crossentropy'],
optimizer=opt, metrics=['accuracy'])
    return model

```

**Kode 14.11 Potongan Kode Pembuatan Model HAPN**

Kode 5.11 menjelaskan prose pembuatan model *Hierarchical Attention Position Aware Network* pada fungsi `build_model()`. Fungsi ini memiliki berbagai argument `class_num`, `optimizer`, `dropout` dll yang akan memiliki nilai berbeda-beda sesuai skenario yang akan diujicoba. Hal pertama yang dilakukan yaitu mengisi nilai `max_len` atau maksimal panjang atau jumlah kata pada kalimat dalam dataset. Setelah itu akan dibuat *sentence input*, *aspect input*, dan *position input* yang berisi indeks kata serta indeks posisi dalam kalimat. Indeks indeks ini kemudian akan diubah menjadi *vector* dalam layer *embedding* berdasarkan kamus *word embedding* dan *position embedding* yang telah dibuat. Setelah itu tiap *embedding* layer baik *sentence*, *aspect* dan *position* akan di-*concanate* atau digabung menjadi 1 layer yaitu *input embedding*. Setelah itu *input embedding* akan dimasukkan ke dalam layer *neural network* dan kemudian dicari atensinya. Model kemudian di-*compile* dengan *optimizer* sesuai scenario, *loss categorical\_crossentropy* dan metrik pengukuran yaitu *accuracy*.

## 5.6. Proses Pelatihan Model HAPN

Model HAPN akan dilatih sesuai dengan skenario yang telah ditentukan sebelumnya. Untuk pembuatan skenario dan pembagian data yang akan dilakukan pada saat proses pelatihan dapat dilihat dalam potongan kode 5.12 berikut :



```

subtask = 'B'
target_names = ['negative', 'netral', 'positive']
if subtask == 'B':
    data = data[data['label'] != 'neutral']
    target_names = ['negative', 'positive']

Optimizer = ['Adam', 'Adadelata', 'SGD', 'RMSprop']
Lr = [0.001, 0.0015, 0.005, 0.01]
Model = ['SimpleRNN', 'LSTM', 'GRU']
dropout = [0.05, 0.01, 0.02, 0.03, 0.04]
scen = [Model, Optimizer, Lr, dropout]
data_scen = list(itertools.product(*scen))
scenario = pd.DataFrame(data_scen,
                        columns=['Model', 'Optimizer', 'Lr', 'Dropout' ])

row_count = data.shape[0]
split_point = int(row_count*0.8)
data80, data20 = data[:split_point], data[split_point:]

skf = KFold(n_splits=5, random_state = 1, shuffle =
            False)

```

#### Kode 14.12 Potongan Kode Inisiasi Skenario dan Pembagian Data

Kode 5.12 di atas menjelaskan inisiasi dari scenario dimana inisiasi diawali dengan mengisi nilai substask. Dimana jika nilai substask yaitu 'B' maka dataset akan difilter sesuai dengan definisi substask yaitu hanya memiliki 2 label. Setelah itu akan diisi nilai-nilai parameter lain untuk skenario yaitu *optimizer*, *learning rate*, model *neural network*, *dropout*. sehingga didapatkan kombinasi parameter untuk tiap skenario. Kemudian dilakukan pembeagian data, seperti diketahui pada bagian 4.6 sebelumnya *dataset* akan dibagi menjadi 3 bagian yaitu *Train set*, *Dev set* dan *Test set*. *Train set* dan *Dev set* akan digunakan dalam proses *training* untuk mendapatkan model terbaik.

Sedangkan *Test set* akan digunakan pada proses evaluasi model untuk menguji seberapa baik performa model pada data yang belum pernah dilihat sebelumnya. Data akan dibagi dengan komposisi 80:20 untuk *Train-Dev* set dan *Test set* menggunakan indeks split point. Sementara pada data *Train-Dev* akan dilakukan *5-fold cross validation* menggunakan *method KFold* dengan  $n\_splits = 5$ .

```
def train_model(sentence_inputs=[], position_inputs=[],
                aspect_input=[], labels=[], model=None,
                sentence_inputs_test=[], position_inputs_test=[],
                aspect_input_test=[], labels_test=[], epoch=None):
    valX = {'sentence_input': sentence_inputs_test,
            'position_input': position_inputs_test, 'aspect_input':
            aspect_input_test}
    valY = labels_test
    history = model.fit({'sentence_input':
                        sentence_inputs, 'position_input': position_inputs,
                        'aspect_input': aspect_input}, labels, epochs=epoch,
                        batch_size=128, verbose=0, validation_data=(valX,
                                                                    valY))
    return model, history
```

#### Kode 14.13 Potongan Kode Pembuatan Fungsi *Training Model*

Kode 5.13 diatas menjelaskan fungsi `train_model()` untuk melakukan proses pelatihan model. Dimana diperlukan argument berupa *sentence\_inputs*, *position\_inputs*, *aspect\_input*, *labels*, *model*, *sentence\_inputs\_test*, *position\_inputs\_test*, *aspect\_input\_test*, *labels\_test*, dan *epoch*. Setelah itu proses pelatihan dimulai dengan memanggil *method fit* dari *library keras*. Setelah memiliki fungsi untuk melakukan *training model*, proses pelatihan model sesuai skenario dapat dijalankan yang akan dijelaskan pada Kode 5.14 berikut:

```

fold = 1
for train_index, test_index in skf.split(data80.text):
    train_sentence_inputs ,test_sentence_inputs =
    sentence_inputs[train_index],
    sentence_inputs[test_index]
    train_aspects, test_aspects =
    pad_aspect[train_index], pad_aspect[test_index]
    train_aspect_labels, test_aspect_labels =
    aspect_label[train_index], aspect_label[test_index]
    train_positions, test_positions =
    position_input[train_index], position_input[test_index]
    train_true_label = aspect_true_label[train_index]
    test_true_label = aspect_true_label[test_index]

    for scen_ind in range(len(scenario)):
        arch = scenario.iloc[scen_ind]['Model']
        optimizer=scenario.iloc[scen_ind]['Optimizer']
        lr = scenario.iloc[scen_ind]['Lr']
        dropout = scenario.iloc[scen_ind]['Dropout']
        model = build_model(...)
        evaluator = Evaluator_3(test_true_label)
        epoch = 1
        max_acc = 0
        while epoch <= 30:
            model, history = train_model(...)
            results = get_predict(...)
            F, FPN, acc, P , R =
            evaluator.get_macro_f1(...)
            if acc > max_acc:
                fold_results_argmax =
                np.argmax(results, axis=1)
                max_acc = acc
                model.save_weights(...)
        fold = fold +1

```

**Kode 14.14 Potongan Kode Penjalanan Skenario Pelatihan Model**

Kode 5.14 di atas menjelaskan proses pelatihan model sesuai skenario yang telah ditentukan. Kode diawali dengan inisiasi nilai `fold = 1` dikarenakan proses pelatihan akan dilaksanakan sebanyak 5 kali untuk tiap skenario. Hal ini dikarenakan pada proses pelatihan akan dilakukan *5-fold cross validation*. Kemudian looping `for` dijalankan untuk setiap `fold`. Dimana data akan dibagi menjadi data train dan data dev(data test) dengan menggunakan indeks data tiap `fold` yang telah ditentukan sebelumnya. Kemudian akan dijalankan *for looping* kembali untuk melakukan proses pelatihan sesuai parameter pada skenario dimana akan dipanggil fungsi `build_model()` dan `train_model()` untuk menjalankan pelatihan dengan epoch sebesar 30. Model dengan akurasi terbaik kemudian akan disimpan *weights*-nya dalam sebuah *file*.

## 5.7. Pengujian Performa Model HAPN

Pada tahap ini, setiap model dari skenario akan diuji pada test set untuk mengevaluasi performa masing-masing model dan membandingkannya satu sama lain. Metrik pengukuran yang digunakan sesuai pada bab 4.9. yaitu *accuracy*, *recall*, dan *f1-score*.

```
def get_predict(sentence_inputs=[], position_inputs=[],
               aspect_input=[], model=None):
    results = model.predict({'sentence_input':
                             sentence_inputs, 'position_input': position_inputs,
                             'aspect_input': aspect_input}, batch_size=64,
                             verbose=0)
    return results
```

### Kode 14.15 Potongan Kode Pengambilan Hasil Prediksi

Kode 5.15 di atas menjelaskan fungsi `get_predict()` untuk melakukan proses pengambilan hasil prediksi dari model yang telah dibuat. Dimana diperlukan argument berupa *sentence\_inputs*, *position\_inputs*, *aspect\_input*, *model*,

Kemudian dilakukan proses prediksi dengan memanggil *method predict* dari *library keras*.

```

class Evaluator_2(object):

    def __init__(self, true_labels=[]):
        .....

    def get_positive_example_num(self, true_labels=[]):
        temp = [0] * 2
        for label in true_labels:
            temp[label] += 1
        self.num0 = temp[0]
        self.num1 = temp[1]

    def calculate_f_score(self, p_p=0, pr_p=0, num=0):
        .....
        F = 2 * P * R / (P + R)
        return P, R, F

    def calculate_acc(self, predicted_positives=[],
        .....
        acc = float(temp2) / temp1
        return acc

    def get_macro_f1(self, predictions=[], epoch=-1,
result_train_test=[], poch = -1):
        predictions =
self.get_predicted_label(predictions)
        p_temp = [0] * 2
        pr_temp = [0] * 2
        .....
        return F, FPN, acc, P, R

```

**Kode 14.16 Potongan Kode Class Untuk Menghitung Metriks Evaluasi**

Kode 5.16 diatas menjelaskan class `Evaluator_2()` untuk melakukan proses penghitungan metrics evaluasi dari hasil prediksi. Pada class `Evaluator_2` terdapat fungsi-fungsi untuk menghitung metrics evaluasi. Fungsi pertama yaitu `get_positive_example()` yang akan membuat 2 array untuk setiap label dan mengisinya dengan nilai 1 pada indeks label sesuai dengan label yang benar, Fungsi kedua yaitu `calculate_f_score()` yang akan menghitung *f1-score* dan *recall* untuk tiap label. Sementara fungsi ke 4 yaitu `calculate_acc()` yang akan menghitung *accuracy*. Dan fungsi terakhir yaitu `get_macro_f1()` yang akan menghitung dan mengembalikan nilai macroaverage dari *f1-score*, *recall* serta nilai *accuracy*.

```

model_path =
"./politik_models/B4_sub_B_scen_0_opt_RMSprop_lr_0.001
_arch_GRU_drop_0.01_fold_1_acc_0.74291_14"
model.load_weights(model_path)

test_sentence_inputs = sentence_inputs[split_point:]
test_aspects = pad_aspect[split_point:]
test_aspect_labels = aspect_label[split_point:]
test_positions = position_input[split_point:]
test_true_label = aspect_true_label[split_point:]
evaluator = Evaluator_2(test_true_label)

results =
get_predict(sentence_inputs=test_sentence_inputs,positi
on_inputs=test_positions,aspect_input=test_aspects,mode
l=model)
F, FPN, acc, P , R =
evaluator.get_macro_f1(predictions=results, epoch=1,
result_train_test=result_train_test, poch=epoch)

```

**Kode 14.17 Potongan Kode Prediksi Data dan Penghitungan Metriks Evaluasi**

Kode 5.17 diatas menjelaskan proses prediksi dan penghitungan metrics evaluasi dari model HAPN yang telah dibuat dan dilatih sebelumnya. Model yang telah disimpan dalam sebuah file akan dimasukkan atau di-load kembali menggunakan fungsi `load_weights`. Setelah itu input untuk data test akan diisi sesuai dengan indeks `split points` yang telah didefinisikan sebelumnya. Kemudian akan dibuat class evaluator, dan dilakukan prediksi data dengan memanggil fungsi `get_predict()` dengan parameter data test dan model yang telah diload sebelumnya. Kemudian dilakukan penghitungan metrics evaluasi menggunakan fungsi `get_macro_f1()` dari class evaluator berdasarkan hasil prediksi sebelumnya dan label asli.

*Halaman ini sengaja dikosongkan*



## BAB VI HASIL DAN PEMBAHASAN

Pada bab ini akan dijelaskan mengenai hasil dan analisis terhadap hasil yang diperoleh dari proses implementasi penelitian.

### 6.1. Hasil Data *Crawling*

Proses pengambilan data yang didapatkan dari tanggal 22 Oktober 2019 hingga 31 Oktober 2019 adalah sebanyak 321,591 *tweet*. Hasil tersebut adalah hasil kotor dari proses *crawling*, sebelum diterapkan nya pra-pemrosesan data.









**Tabel 15.1 Hasil *Crawling Tweet***

Keyword	Tweet
ahok	Saya mencontohkan apa yang dilakukan Pak BTP waktu menjabat Gubernur. Beliau selama 13 jam dalam sehari menyisir anggaran bersama dengan seluruh SKPD terkait." <a href="https://t.co/T1QZzHmeOQY">https://t.co/T1QZzHmeOQY</a> jangan dibandingkan Ahok dgn sultan Aibon yg kerjanya hamburkan anggaran#ExtraordinaryYou
jokowi	wadidaw pak jokowi lagi screening calon" menterinyaa... oke otw istana siapa tau ada tempat ye kannn buat fresh graduate.. jadi apa ya negara klo diri ini jadi menteri? btw, cocoknya jadi menteri apaan yaakk? (tolong jgn dihujat, disadarkan ajaa) ehehe <a href="https://t.co/YWfRAwXpxG">https://t.co/YWfRAwXpxG</a>
kpk	Dengan diundangkannya UU KPK, semua pihak harus menghormati. Bagaimanapun revisi UU KPK sudah menjadi UU baru #UUKPKTerbaru <a href="https://t.co/H2XwM8UjWB">https://t.co/H2XwM8UjWB</a> <a href="https://t.co/xd6jK4W8PU">https://t.co/xd6jK4W8PU</a>
prabowo	Yg selama ini sdh 5 tahun berhimpun di dalam dan habis æmasa kontraknyaâ€• mesti legowo... Meski yang sekarang masuk dalam masa kontrak berikutnya, namanya Prabowo. Katanya, politik itu dinamis....Makanya jangan cuma mau manisnya saja, cecap juga pahitnya agar tetap legit...

## 6.2. Hasil Penghapusan Duplikasi Dataset

Hasil dari *crawling twitter* masih menyisakan duplikasi data atau data dengan *tweet* yang sama sehingga diperlukan proses penghapusan data duplikat. Tabel 6.2 berikut menunjukkan contoh data *tweet* terduplikasi.

**Tabel 15.2 Hasil Duplikasi Dataset**

Id_tweet	Tweet
1186876502639439872	 Yg milih Prabski : Gila kecewa bgt sama Prabowo, malah masuk kabinet Jokowi.  Me : Lu pendukung Prabowo apa pembenci Jokowi? Bedha lho~  #kabinetJokowiAmin <a href="https://t.co/1ATIVCJrXE">https://t.co/1ATIVCJrXE</a>
1186876502639439872	 Yg milih Prabski : Gila kecewa bgt sama Prabowo, malah masuk kabinet Jokowi.  Me : Lu pendukung Prabowo apa pembenci Jokowi? Bedha lho~  #kabinetJokowiAmin <a href="https://t.co/1ATIVCJrXE">https://t.co/1ATIVCJrXE</a>
1187377904931373056	#PSI udah GUE RIJEK Maaf, #Demokrat PARTAI LAMA, dan bukan PARTAI Kemarin Sore !!#SAP #TPP 
1187377904931373056	#PSI udah GUE RIJEK Maaf, #Demokrat PARTAI LAMA, dan bukan PARTAI Kemarin Sore !!#SAP #TPP 

Setelah menghapus data yang terduplikasi maka dihasilkan 255,931 data posting *twitter* yang unik yang akan masuk dalam proses filtering bahasa. Nantinya akan dipilih sejumlah data yang akan melalui proses *labeling* untuk proses *training* model *Hierarchical Attention Position-aware Network*

### 6.3. Filtering Bahasa

Total data yang didapatkan dari proses *crawling* adalah 255,931 data posting *twitter* yang unik. Namun di dalam dataset tersebut masih mengandung data yang tidak mengandung Bahasa Indonesia. Maka dilakukan proses filterisasi bahasa terhadap dataset untuk memastikan bahwa keseluruhan bahasa yang digunakan dari dataset adalah data yang mengandung Bahasa Indonesia. Untuk melakukan filterisasi bahasa digunakan *library* *langdetect* dengan nilai *indeks* kepercayaan bahasa sebesar lebih dari 90%. Nilai *confidency* sebesar lebih dari 90% digunakan untuk memastikan bahwa seluruh data yang didapat setelah proses *filter* adalah data yang menggunakan Bahasa Indonesia serta untuk mencegah data yang menggunakan bahasa serumpun seperti Bahasa Melayu ikut masuk ke dalam kumpulan dataset.

**Tabel 15.3 Hasil Seleksi Bahasa Indonesia**

<i>Tweet</i>	<b>Bahasa</b>	<b>Index</b>
FIRMA Y DIFUNDE- $\&$ ; Anies Baswedan: Untuk Anies Baswedan: Stop razia kucing - ¡Firma la petición! <a href="https://t.co/3syxzVUII">https://t.co/3syxzVUII</a> vía @ChangeorgLatino	Espanyol	0.428570
Ahok recién me entero que mi viejo quedo internado lpm	Espanyol	0.999997
Jadi ini yg salah sistem warisan ahok atau pegawai yg salah input ? Jadi lem dan pulpen yg dibutuhkan brp sih sebenarnya. Di timeline gue ada 2 golongan yg bahas ini. Yg nyalahin sistem. Yg nyalahin anies dan jajarannya	Indonesia	0.999996
Valid : Sandiaga Uno hari ini jadi menteri...,hhehe	Indonesia	0.857139
idc about the president, i want to see what Sandiaga Uno can do as minister (kalau kepilih)	English	0.571427

Setelah dilakukan proses *filter* dihasilkan 163,331 data yang menggunakan Bahasa Indonesia. Dari hasil seleksi Bahasa Indonesia menggunakan *library langdetect*, *library* ini cukup berhasil menentukan bahasa dari sebuah *tweet*. Tetapi hasil yang diberikan kurang akurat jika jumlah kata pada kalimat *tweet* sedikit.

#### 6.4. Hasil Pembersihan Dataset

Sebelum dataset akan digunakan untuk pembentukan model HAPN akan dilakukan proses pembersihan disetiap *tweet* dengan tujuan menghilangkan karakter yang tak bermakna serta memperkecil peluang kata yang memiliki makna sama namun berbeda penulisan karena penambahan tanda baca. Berikut hasil pemberihan dataset.

**Tabel 15.4 Hasil Pembersihan Dataset**

Sebelum	Sesudah
Ini orang2 masih "bully" Anies karena ngerasa oposisi. Coba kalau PDIP mendadak dukung Anies, mingkem deh cebong2 barisan sakit hati pembela Ahok.	ini orang2 masih bully anies karena ngerasa oposisi coba kalau pdip mendadak dukung anies mingkem deh cebong2 barisan sakit hati pembela ahok
Mau kagak mau kalo resehhh mulu, aku akan selalu bandingin sama AHOK! Memangg kita appes kok kalo inget2 pilkada 2017!!! Menderitaa kagak berujungg buah dari pilkada 2017! HOT TOPIC PAGI @mncrijayafm om @doudyTATIPANG om IKE @enweka om @LatiefSiregar	mau kagak mau kalo resehh mulu aku akan selalu bandingin sama ahok memangg kita appes kok kalo inget2 pilkada 2017 menderitaa kagak berujungg buah dari pilkada 2017 hot topic pagi om om ike om
Wawancara dg Eddi Prabowo top penjelasan nya ....dg apa yg hrs jd kebijakan nya...slamat tugas Mas Eddi Prabowo...smga amanah	wawancara dg eddi prabowo top penjelasan nya dg apa yg hrs jd kebijakan nya selamat tugas mas eddi prabowo smga amanah
Dengan diundangkannya UU KPK, semua pihak harus	dengan diundangkannya uu kpk semua pihak harus

menghormati. Bagaimanapun revisi UU KPK sudah menjadi UU baru #UUKPKTerbaru <a href="https://t.co/H2XwM8UjWB">https://t.co/H2XwM8UjWB</a> <a href="https://t.co/xd6jK4W8PU">https://t.co/xd6jK4W8PU</a>	menghormati bagaimanapun revisi uu kpk sudah menjadi uu baru
---	--

### 6.5. Hasil Pelabelan Data

Data yang telah dikumpulkan melalui proses *crawling* yang digunakan dalam model *Hierarchical Attention Position-aware Network* memiliki jumlah total 3022 data yang berasal dari *keyword* berdasarkan entitas politik yang berdasarkan pada sub bab 3.1.2 dengan pembagian data seperti pada Tabel 6.5.

**Tabel 15.5 Pembagian Data Berdasarkan Topik**

Kategori	Entitas	Kata Kunci	Jumlah
Partai Politik	Partai Kebangkitan Bangsa	pkb	-
	Partai Gerindra	gerindra	69
	Partai Demokrasi Indonesia Perjuangan	pdip	8
	Partai Golkar	golkar	10
	Partau Nasional Demokrat	nasdem	36
	Partai Garuda	partai garuda	-
	Partai Berkarya	partai berkarya	-
	Partai Keadilan Sejahtera	psk	42
	Partai Perindo	perindo	1
	Partai Persatuan Pembangunan	ppp	-
	Partai Solidaritas Indonesia	psi	17
	Partai Amanat Nasional	pan	-
	Partai Hanura	hanura	4
	Partai Demokrat	democrat	41
	Partai Bulan Bintang	partai bulan bintang	1

Kategori	Entitas	Kata Kunci	Jumlah
	Partai Keadilan dan Persatuan Indonesia	pkpi	-
Tokoh Politik	Joko Widodo	jokowi	1125
	Prabowo Subianto	prabowo	447
	Maaruf Amin	maaruf amin	4
	Sandiaga Uno	sandiaga uno	2
	Anies Baswedan	anies	35
	Basuki Tjahaja Purnama	ahok	83
	Ganjar Pranowo	ganjar pranowo	-
	Ridwan Kamil	ridwan kamil	2
	Tri Rismaharini	bu risma	5
	Fahri Hamzah	fahri hamzah	21
	Fadli Zon	fadli zon	9
Mahfud MD	mahfud md	36	
Isu Politik	DPR	dpr	77
	Kabinet	kabinet	736
	Kementerian	kementerian	79
	KPK	kpk	50
	Pemilu	pemilu	68
	RAPBD	rapbd	10
	RKUHP	rkuhp	2

Dapat dilihat bahwa terdapat *keyword-keyword* yang tidak memiliki topik. Selain itu 3 *keyword* dengan jumlah *tweet* terbanyak dapat dilihat yaitu Jokowi, Kabinet dan Prabowo.

Proses pelabelan kemudian dilakukan dengan menggunakan 3 label yaitu positif, negatif dan netral pada 3022 data *tweet*. Hasil penyebaran label dapat dilihat sebagai berikut :

**Tabel 15.6 Penyebaran Data Hasil Final Pelabelan**

Label	Jumlah Data
<b>Positif</b>	1912
<b>Netral</b>	568
<b>Negatif</b>	1395
<b>Total Data</b>	3875

Dari hasil ditunjukkan pada dataset yang akan digunakan terdapat imbalance data, dapat dilihat pada data netral yang hanya memiliki jumlah 568 sementara data positif dan negatif hampir 3x jumlah data netral. Hal ini diakibatkan dari keterbatasan yang dimiliki dari proses pengumpulan data. Tabel 6.7 memperlihatkan contoh hasil pelabelan yang telah dilakukan.

**Tabel 15.7 Hasil Pelabelan Aspek dan Sentimen**

Teks	Aspek	Label Sentimen Aspek
koq sejak dilantik kabinet kerja jilid2 banyak twit ttg anggaran yg seperti disusun kucing2an atau tikus2an	kabinet kerja jilid2	Negatif
kemaren ada seneng ada sedihnya gitu waktu jokowi ngumumin kabinetnya yg baru kabinet indonesia maju yaa kaya kehidupan kali ya ada ups and downsnya	kabinet indonesia maju	Netral
w sih gak ngerti2 amat soal politik yaatapi seneng aja ngeliat prabowo ada di jajaran kabinet jokowi maruf setelah pemilu kemaren jadi gimana nih pendukung fanatiknya prabowo	prabowo	Positif

## 6.6. Hasil Implementasi Model *Word Embedding*

Implementasi *pretrained word embedding model word2vec* dari penelitian sebelumnya menghasilkan 9062 kata yang terdeteksi dan 1040 kata yang tidak terdeteksi dalam dataset pada subtask A. Dan 8360 kata yang terdeteksi dan 935 kata yang tidak terdeteksi dalam dataset pada subtask B. Kata yang diketahui

memiliki nilai representasi vektor, sementara kata yang tidak diketahui diberi nilai vektor secara random. Sehingga total kata pada kamus *word vector* adalah 10102 kata untuk Subtask A dan 9295 kata untuk Subtask B.

**Tabel 15.8 Jumlah Kata Terdeteksi dari Model *Word Embedding Word2vec***

Subtask	Kata Terdeteksi	Kata Tidak Terdeteksi	Total Kata
Subtask A	9062	1040	10102
Subtask B	8360	935	9295

## 6.7. Hasil Pengujian Data

Skenario pembuatan model dalam penelitian ini terdapat 5 jenis seperti telah dijelaskan di bab 4.8 sebelumnya.

### 6.7.1 Konfigurasi Parameter Awal

Sebelum melakukan skenario pertama, harus terlebih dahulu ditentukan konfigurasi parameter model untuk proses *training* diawal agar semua skenario dapat dibandingkan hasilnya dengan jelas. *Hyperparameter* akan ditentukan berdasarkan penelitian sebelumnya, berikut tabel terkait konfigurasi awal.

**Tabel 15.9 Konfigurasi Awal Model HAPN**

Parameter	Nilai
<i>Epoch</i>	30
<i>Model Neural Network</i>	GRU
<i>Optimizer</i>	Adam
<i>Learning Rate</i>	0.001
<i>Dropout</i>	0.5
<i>Word Embedding Model</i>	Word2Vec Skip-gram
<i>Character Embedding Model</i>	-

### 6.7.2 Hasil Percobaan Subtask A

*Subtask A* merupakan subtask dengan 3 *point scale* dengan label yang digunakan adalah label positif, label negatif dan label netral. Dengan penyebaran data sebagai berikut :



**Tabel 15.10 Penyebaran Data pada Subtask A**

<b>Label</b>	<b>Jumlah Data</b>
<b>Positif</b>	1912
<b>Netral</b>	568
<b>Negatif</b>	1395
<b>Total Data</b>	3875

Distribusi data berdasarkan label setelah melalui implementasi data splitting dapat dilihat pada tabel 6.11.

**Tabel 15.11 Penyebaran Label Pada tiap Data Subtask A**

<b>Data</b>	<b>Positif</b>	<b>Netral</b>	<b>Negatif</b>	<b>Total</b>
<b>Train - Fold 1</b>	279	110	231	620
<b>Train - Fold 2</b>	331	90	199	620
<b>Train - Fold 3</b>	327	88	205	620
<b>Train - Fold 4</b>	283	103	234	620
<b>Train - Fold 5</b>	315	70	235	620
<b>All Train Data</b>	1535	461	1104	3100
<b>Test Data</b>	377	107	291	775

### **6.7.2.1 Pengaruh Model *Neural Network***

Skenario satu adalah menguji efek perubahan model *neural network* terhadap akurasi model. Model *neural network* yang digunakan adalah SimpleRNN, LSTM dan GRU. Output dari skenario ini adalah model *neural network* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *F1-score*. Berikut hasil dari skenario pertama untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.12 Hasil Akurasi Percobaan Model *Neural Network* Pada Data *Train* Subtask A**

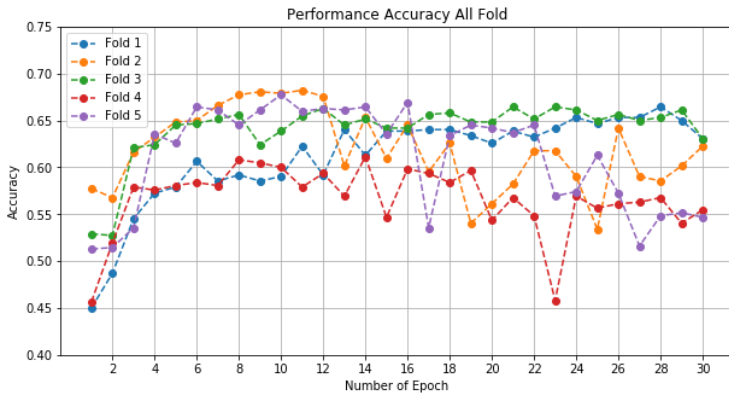
<b>Data / Model</b>	<b>Simple RNN</b>	<b>LSTM</b>	<b>GRU</b>
<b>Train - Fold 1 (%)</b>	65.65	66.45	64.84
<b>Train - Fold 2 (%)</b>	69.19	68.23	69.19
<b>Train - Fold 3 (%)</b>	66.61	66.45	66.61
<b>Train - Fold 4 (%)</b>	60.16	61.13	62.10
<b>Train - Fold 5 (%)</b>	68.06	67.74	67.42
<b>All Train Fold (%)</b>	65.94	66.00	66.03

Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. Model NN GRU merupakan model yang memiliki akurasi terbaik tetapi model ini hanya unggul 0.03% dari model LSTM untuk keseluruhan data. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.13 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.13 Hasil Evaluasi Pengukuran Percobaan Model *Neural Network* Pada Data *Test* Subtask A**

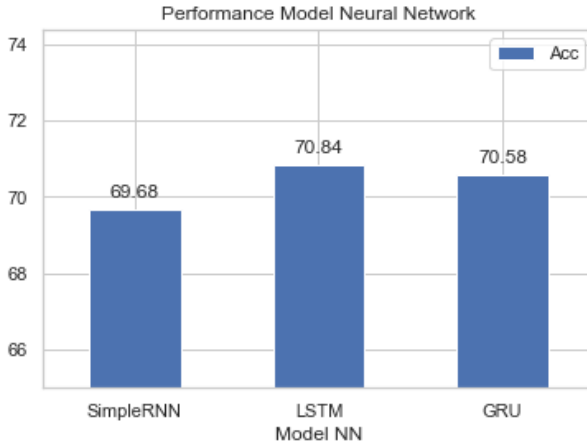
<b>Model NN</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>SimpleRNN</b>	69.68	55.89	73.96
<b>LSTM</b>	70.84	62.87	74.64
<b>GRU</b>	70.58	56.48	74.19

Pada percobaan yang dilakukan disimpulkan bahwa model terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah model *neural network* LSTM. model *neural network* LSTM mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh model pada data test.



**Gambar 15.1** Performa Model Terbaik *Neural Network* Subtask A

Pada Gambar 6.1 Jika dilihat dari performa yang dihasilkan model terbaik LSTM berdasarkan perubahan setiap *epoch* pada tiap fold, kebanyakan fold memiliki accuracy yang meningkat tajam pada epoch 2-6. Sementara kebanyakan fold akan mencapai puncak nilai *accuracy* pada *epoch* 10-20. Pada model LSTM 2 fold mengalami performa stabil pada *epoch* 15++ yaitu fold 1 dan fold 3, sementara 3 fold lainnya mengalami naik turun yang cukup ekstrem.



**Gambar 15.2 Perbandingan Akurasi pada Percobaan Model *Neural Network* Subtask A**

Berdasarkan model *neural network* yang digunakan model LSTM memiliki performa yang lebih baik. Akurasi dapat meningkat hingga 1.16 % jika menggunakan model LSTM. Sehingga model LSTM akan digunakan menjadi *baseline* parameter model *neural network* pada skenario berikutnya.

### 6.7.2.2 Pengaruh *Optimizer*

Skenario kedua adalah menguji efek perubahan optimizer terhadap akurasi model. *Optimizer* yang digunakan adalah Adam, Adadelta, SGD dan RMSprop. Output dari skenario ini adalah *optimizer* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.14 Hasil Akurasi Percobaan *Optimizer* Pada Data *Train* Subtask A**

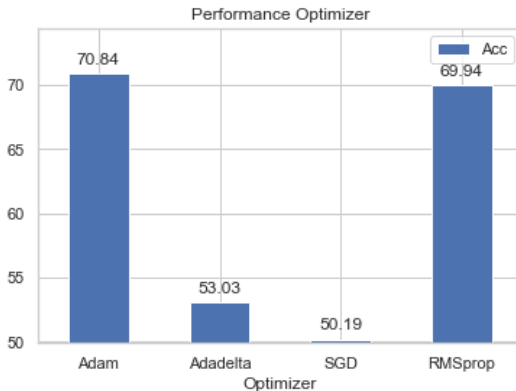
<b>Data / Optimizer</b>	<b>Adam</b>	<b>Adadelta</b>	<b>SGD</b>	<b>RMSprop</b>
<b>Train - Fold 1 (%)</b>	66.45	37.58	45.65	66.45
<b>Train - Fold 2 (%)</b>	68.23	54.52	54.03	69.52
<b>Train - Fold 3 (%)</b>	66.45	52.74	52.74	67.90
<b>Train - Fold 4 (%)</b>	61.13	46.29	45.97	63.06
<b>Train - Fold 5 (%)</b>	67.74	53.23	51.29	68.23
<b>All Train Fold (%)</b>	66.00	48.87	49.94	67.03

Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. *Optimizer* RMSprop merupakan optimizer yang memiliki akurasi terbaik untuk seluruh fold. Pada keseluruhan data *optimizer* RMSprop ini hanya unggul 1.03% dari *optimizer* Adam. Sementara 2 *optimizer* lainnya yaitu Adadelta dan SGD memiliki akurasi yang cukup buruk untuk seluruh data fold dimana nilai akurasi berada dibawah 55%. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.15 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.15 Hasil Evaluasi Pengukuran Percobaan *Optimizer* Pada Data *Test* Subtask A**

<b>Optimizer</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>Adam</b>	70.84	62.87	74.64
<b>Adadelata</b>	53.03	37.65	47.17
<b>SGD</b>	50.19	34.76	37.59
<b>RMSprop</b>	69.94	58.62	74.74

Pada percobaan yang dilakukan disimpulkan bahwa optimizer terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-Score* adalah *optimizer* Adam. *Optimizer* Adam mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh *optimizer* pada data test.



**Gambar 15.3 Perbandingan Akurasi pada Percobaan *Optimizer* Subtask A**

Berdasarkan Gambar 6.3 *Optimizer* Adam memiliki performa yang paling baik. Akurasi dapat meningkat hingga 20.65% jika menggunakan *optimizer* Adam. Sehingga *optimizer* Adam akan digunakan menjadi *baseline* parameter *optimizer* pada skenario berikutnya.

### 6.7.2.3 Pengaruh *Learning Rate*

Skenario ketiga adalah menguji efek perubahan *learning rate* terhadap akurasi model. Nilai *learning rate* yang digunakan adalah 0.0005, 0.001, 0.002, 0.005, dan 0.01 . Output dari skenario ini adalah nilai *learning rate* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.16 Hasil Akurasi Percobaan *Learning Rate* Pada Data *Train* Subtask A**

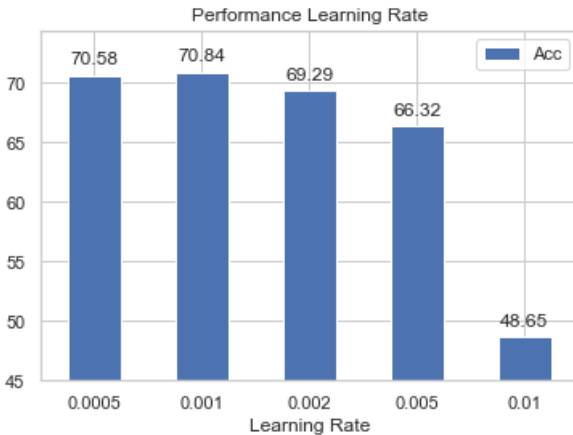
<b>Data / Learning Rate</b>	<b>0.0005</b>	<b>0.001</b>	<b>0.002</b>	<b>0.005</b>	<b>0.01</b>
<b>Train - Fold 1 (%)</b>	64.68	66.45	65.97	64.68	37.26
<b>Train - Fold 2 (%)</b>	69.68	68.23	68.87	66.94	32.10
<b>Train - Fold 3 (%)</b>	66.61	66.45	65.65	65.81	52.74
<b>Train - Fold 4 (%)</b>	62.10	61.13	60.97	61.13	45.65
<b>Train - Fold 5 (%)</b>	67.58	67.74	67.58	65.81	50.81
<b>All Train Fold (%)</b>	66.13	66.00	65.81	64.87	43.71

Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. *Learning rate* 0.0005 merupakan *learning rate* yang memiliki akurasi terbaik tetapi *learning rate* ini hanya unggul 1.13% dari *learning rate* 0.001 untuk keseluruhan data fold. *learning rate* 0.0005 unggul pada fold 2,4 dan 5 sementara *learning rate* 0.001 unggul pada fold 1 dan 3. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.15 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.17 Hasil Evaluasi Pengukuran Percobaan *Learning Rate* Pada Data *Test* Subtask A**

<b>Learning Rate</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>0.0005</b>	70.58	59.15	74.56
<b>0.001</b>	70.84	62.87	74.64
<b>0.002</b>	69.29	55.17	73.28
<b>0.005</b>	66.32	53.90	70.21
<b>0.01</b>	48.65	33.33	32.73

Pada percobaan yang dilakukan disimpulkan bahwa *learning rate* terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah 0.001. *Learning rate* 0.001 mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh nilai *learning rate* pada data test.



**Gambar 15.4 Perbandingan Akurasi pada Percobaan *Learning Rate* Subtask A**

Berdasarkan Gambar 6.4 nilai *learning Rate* 0.001 memiliki performa yang paling baik. Dapat dilihat juga bahwa semakin besar nilai *learning rate* maka akurasi semakin jelek. nilai *learning rate* dapat meningkatkan *accuracy* hingga 22.19 % jika menggunakan nilai *learning rate* 0.001. Sehingga nilai



*learning rate* 0.001 akan digunakan menjadi *baseline* parameter *learning rate* pada skenario berikutnya.

#### 6.7.2.4 Pengaruh *Dropout*

Skenario keempat adalah menguji efek perubahan nilai *dropout* terhadap akurasi model. Nilai *dropout* yang digunakan adalah 0, 0.1, 0.2, 0.3, 0.4 dan 0.5 . Output dari skenario ini adalah nilai *dropout* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.18 Hasil Akurasi Percobaan *Dropout* Pada Data *Train* Subtask A**

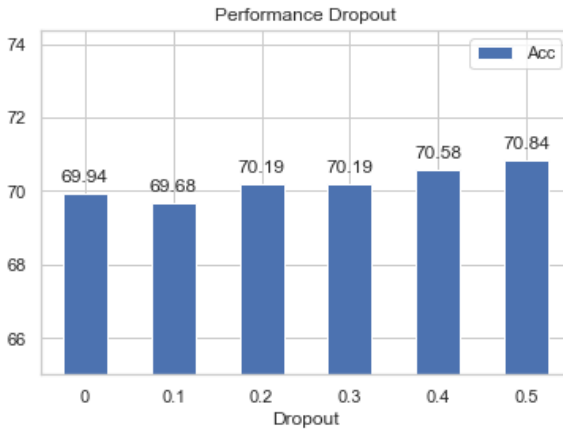
<b>Data / Learning Rate</b>	<b>0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.5</b>
<b>Train - Fold 1 (%)</b>	63.35	65.32	65.48	63.55	63.55	66.45
<b>Train - Fold 2 (%)</b>	66.77	67.42	67.10	68.39	67.74	68.23
<b>Train - Fold 3 (%)</b>	68.06	65.97	67.74	67.10	67.58	66.45
<b>Train - Fold 4 (%)</b>	61.13	63.39	61.45	61.45	62.10	61.13
<b>Train - Fold 5 (%)</b>	68.71	67.90	67.74	68.06	67.90	67.74
<b>All Train Fold (%)</b>	65.65	66.00	65.90	65.71	65.77	66.00

Dari hasil percobaan tersebut, data train fold 5 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. Nilai *dropout* 0.5 dan 0.1 merupakan nilai *dropout* yang memiliki akurasi terbaik. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.15 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.19 Hasil Evaluasi Pengukuran Percobaan Dropout Pada Data Test Subtask A**

<b>Dropout</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>0</b>	69.94	56.50	74.32
<b>0.1</b>	69.68	61.98	73.29
<b>0.2</b>	70.19	62.57	72.88
<b>0.3</b>	70.19	62.72	73.79
<b>0.4</b>	70.58	57.62	74.29
<b>0.5</b>	70.84	62.87	74.64

Pada percobaan yang dilakukan disimpulkan bahwa nilai *dropout* terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah 0.5. nilai *dropout* 0.5 mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh nilai *dropout* pada data test.



**Gambar 15.5 Perbandingan Akurasi pada Percobaan Dropout Subtask A**

Berdasarkan Gambar 6.5 nilai *dropout* 0.5 memiliki performa yang paling baik. Dapat dilihat juga bahwa semakin kecil nilai *dropout* maka akurasi semakin jelek. Nilai *dropout* dapat meningkat *accuracy* hingga 1.16 % jika menggunakan nilai

*dropout* 0.5. Sehingga nilai *dropout* 0.5 akan digunakan menjadi *baseline* parameter *dropout* pada skenario berikutnya.

### 6.7.2.5 Pengaruh *Character Embedding*

Skenario terakhir adalah menguji efek penambahan *character embedding* pada pembangunan model HAPN terhadap akurasi model. *Character embedding* yang digunakan dibangun menggunakan model neural network CNN dan LSTM. Output dari skenario ini adalah memilih model terbaik antara model HAPN dengan *character embedding* atau tidak. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *F1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.20 Hasil Akurasi Percobaan *Character Embedding* Pada Data Train Subtask A**

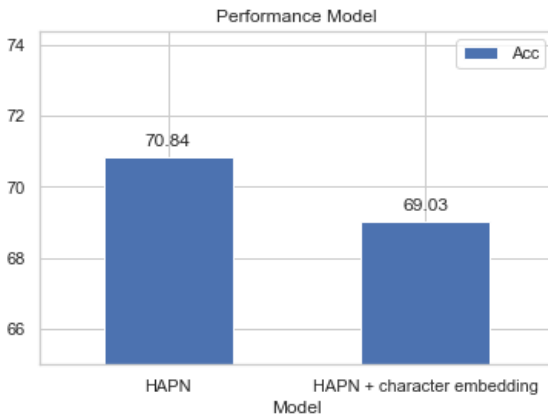
Data / Model HAPN	HAPN	HAPN + <i>Character Embedding</i>
<b>Train - Fold 1 (%)</b>	66.45	65.81
<b>Train - Fold 2 (%)</b>	68.23	67.90
<b>Train - Fold 3 (%)</b>	66.45	66.29
<b>Train - Fold 4 (%)</b>	61.13	61.94
<b>Train - Fold 5 (%)</b>	67.74	68.06
<b>All Train Fold (%)</b>	66.00	66.00

Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. Model HAPN dan HAPN + *character embedding* memiliki nilai akurasi yang sama untuk keseluruhan data fold. Model HAPN unggul pada fold 1,2 dan 3, sementara model HAPN + *character embedding* unggul pada fold 4 dan 5. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.15 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.21 Hasil Evaluasi Pengukuran Percobaan *Character Embedding Test Subtask A***

Model HAPN	Accuracy (%)	AvgRec (%)	F1-Score (%)
<b>HAPN</b>	70.84	62.87	74.64
<b>HAPN + <i>Character Embedding</i></b>	69.03	58.53	72.41

Pada percobaan yang dilakukan disimpulkan bahwa model terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah model HAPN yang menggungguli nilai *accuracy*, *average recall*, dan *f1-score* dari model HAPN + *character embedding*.



**Gambar 15.6 Perbandingan Akurasi pada Percobaan *Character Embedding Subtask A***

Berdasarkan Gambar 6.6 Model HAPN memiliki performa yang paling baik. Model HAPN dapat meningkat *accuracy* hingga 1.81%. Sehingga dari hasil pengukuran evaluasi model terbaik untuk subtask A adalah model HAPN.

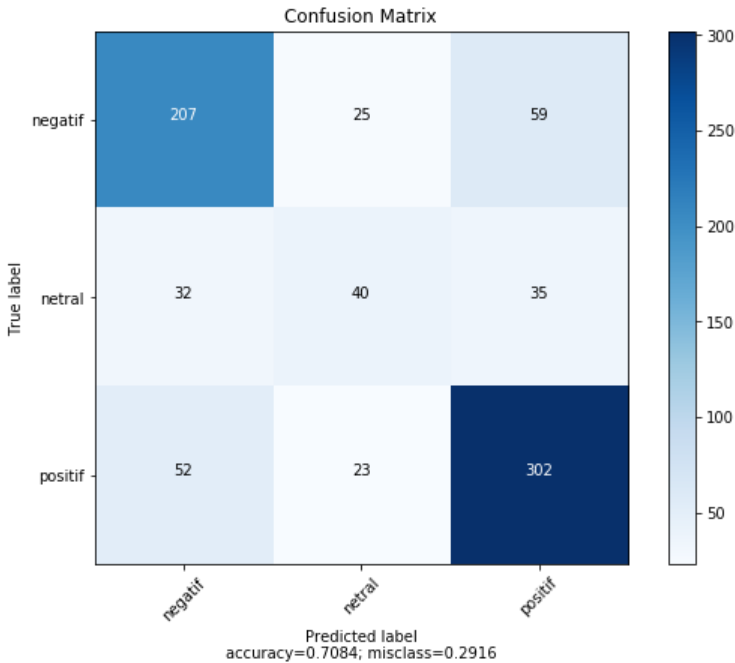
### 6.7.2.6 Pembahasan Hasil Subtask A

Dari hasil subtask A didapatkan performa terbaik yang dapat dilihat sebagai berikut:

Tabel 15.22 Pengukuran Evaluasi Model Terbaik dari Subtask A

Parameter	Nilai	Accuracy (%)	AvgRec (%)	F1-Score (%)
<b>Model Neural Network</b>	LSTM	70.84	62.87	74.64
<b>Optimizer</b>	Adam			
<b>Learning Rate</b>	0.001			
<b>Dropout</b>	0.5			
<b>Character Embedding</b>	-			

Selanjutnya untuk kinerja dari model terbaik dalam melakukan pengklasifikasian data akan dilihat berdasarkan *confusion matrix*. Hasil *confusion matrix* dapat dilihat pada Gambar 6.7. Dapat dilihat bahwa, kesalahan dalam melakukan klasifikasi untuk label positif dan negatif cukup kecil dilihat dari perbandingan kebenaran model dalam melakukan klasifikasi. Sementara untuk label netral model hampir tidak dapat melakukan klasifikasi dengan baik dikarenakan dari 107 data dengan label netral, model berhasil mengklasifikasikan dengan benar sebesar 40 atau 1/3 data. Dimana 1/3 data lain diklasifikasikan menjadi positif dan negatif.



**Gambar 15.7 Confusion Matrix Model Terbaik Subtask A**

Pengecekan selanjutnya dilihat berdasarkan penyebaran nilai *recall* dan *f1-score* pada tiap labelnya. Setelah dilakukan perhitungan maka dihasilkan data sebagai berikut :

**Tabel 15.23 Perhitungan *Recall* dan *F1-Score* per Label pada Model Terbaik Subtask A**

Label	Recall	F1-Score
<b>Negatif</b>	71.0	71.0
<b>Netral</b>	37.0	47.0
<b>Positif</b>	80.0	78.0

Berdasarkan hasil perhitungan yang dilakukan pada subtask A, nilai *recall* dan *f1-score* dalam proses klasifikasi paling rendah adalah netral. Hal ini membuktikan bahwa klasifikasi label netral sering mengalami kesalahan. Hal ini yang menyebabkan nilai *average recall* model secara keseluruhan menjadi rendah.

Sementara nilai *recall* dan *f1-score* dalam proses klasifikasi terbaik adalah label positif.

### 6.7.3 Hasil Percobaan Subtask B

*Subtask B* merupakan subtask dengan 2 *point scale* dengan label yang digunakan adalah label positif dan label negatif. Dengan penyebaran data sebagai berikut :

**Tabel 15.24 Penyebaran Data pada Subtask B**

Label	Jumlah Data
Positif	1912
Negatif	1395
Total Data	3307

Distribusi data berdasarkan label setelah melalui implementasi data splitting dapat dilihat pada tabel 6.24

**Tabel 15.25 Penyebaran Label Pada tiap Data Subtask B**

Data	Positif	Negatif	Total
Train - Fold 1	293	236	529
Train - Fold 2	329	200	529
Train - Fold 3	324	205	529
Train - Fold 4	287	242	529
Train - Fold 5	304	225	529
All Train Data	1537	1108	2645
Test Data	375	287	662

#### 6.7.3.1 Pengaruh Model *Neural Network*

Skenario satu adalah menguji efek perubahan model *neural network* terhadap akurasi model. Model *neural network* yang digunakan adalah SimpleRNN, LSTM dan GRU. Output dari skenario ini adalah model *neural network* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario pertama untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.26 Hasil Akurasi Percobaan Model Neural Network Pada Data Train Subtask B**

<b>Data / Model</b>	<b>Simple RNN</b>	<b>LSTM</b>	<b>GRU</b>
<b>Train - Fold 1 (%)</b>	78.83	77.88	77.88
<b>Train - Fold 2 (%)</b>	80.72	80.15	81.29
<b>Train - Fold 3 (%)</b>	77.69	78.64	78.07
<b>Train - Fold 4 (%)</b>	73.35	73.91	75.05
<b>Train - Fold 5 (%)</b>	77.13	77.13	78.26
<b>All Train Fold (%)</b>	77.54	77.54	78.11

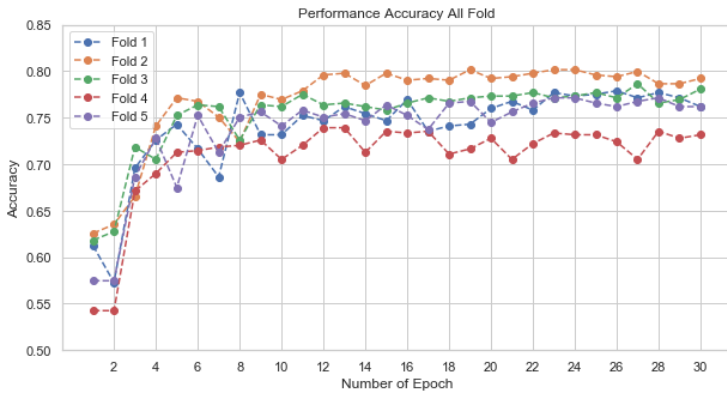
Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. Model NN GRU merupakan model yang memiliki akurasi terbaik tetapi model ini hanya unggul 0.57% dari model LSTM dan SimpleRNN. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.27 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.27 Hasil Evaluasi Pengukuran Percobaan Model Neural Network Pada Data Test Subtask B**

<b>Model NN</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>SimpleRNN</b>	78.55	78.90	78.42
<b>LSTM</b>	80.51	79.57	79.89
<b>GRU</b>	79.61	78.97	79.12

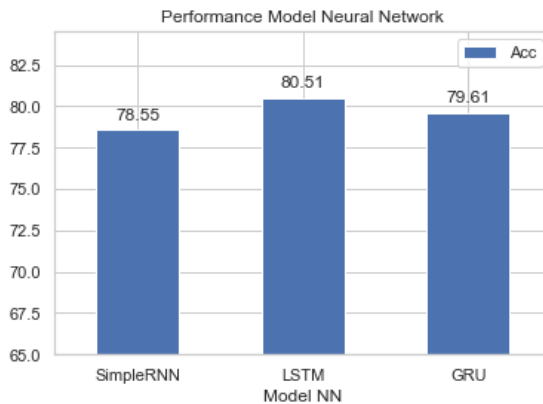
Pada percobaan yang dilakukan disimpulkan bahwa model terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah model *neural network* LSTM. model *neural network* LSTM mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh model pada data test.





**Gambar 15.8** Performa Model Terbaik *Neural Network* Subtask B

Pada Gambar 6.8 Jika dilihat dari performa yang dihasilkan model terbaik LSTM berdasarkan perubahan setiap epoch pada tiap fold, kebanyakan fold memiliki accuracy yang meningkat tajam pada *epoch* 2-6. Sementara kebanyakan fold akan mencapai puncak nilai *accuracy* pada *epoch* 12-28. Pada model LSTM seluruh fold memiliki nilai *accuracy* yang stabil pada *epoch* 12++.



**Gambar 15.9** Perbandingan Akurasi pada Percobaan *Model Neural Network* Subtask B

Berdasarkan *model neural network* yang digunakan model LSTM memiliki performa yang lebih baik. Akurasi dapat meningkat hingga 1.96% jika menggunakan model LSTM. Sehingga model LSTM akan digunakan menjadi *baseline* parameter model neural network pada skenario berikutnya.

### 6.7.3.2 Pengaruh *Optimizer*

Skenario kedua adalah menguji efek perubahan *optimizer* terhadap akurasi model. *optimizer* yang digunakan adalah Adam, Adadelta, SGD dan RMSprop. Output dari skenario ini adalah *optimizer* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.28 Hasil Akurasi Percobaan *Optimizer* Pada Data Train Subtask B**

<b>Data / Optimizer</b>	<b>Adam</b>	<b>Adadelta</b>	<b>SGD</b>	<b>RMSprop</b>
<b>Train - Fold 1 (%)</b>	77.88	55.95	55.77	79.40
<b>Train - Fold 2 (%)</b>	80.15	64.65	64.06	81.10
<b>Train - Fold 3 (%)</b>	78.64	62.38	62.19	78.07
<b>Train - Fold 4 (%)</b>	73.91	57.66	57.28	75.24
<b>Train - Fold 5 (%)</b>	77.13	56.14	58.41	78.26
<b>All Train Fold (%)</b>	77.54	59.36	59.55	78.41

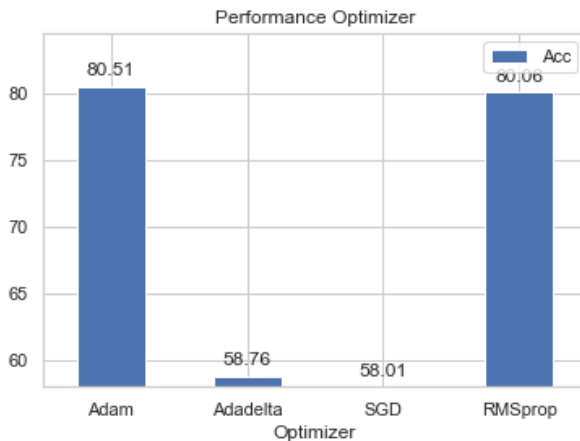
Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. *Optimizer* RMSprop merupakan *optimizer* yang memiliki akurasi terbaik untuk seluruh fold kecuali fold 3. Pada keseluruhan data *optimizer* RMSprop ini hanya unggul 0.87% dari *optimizer* Adam.

Sementara 2 *optimizer* lainnya yaitu Adadelta dan SGD memiliki akurasi yang cukup buruk untuk seluruh data fold dimana nilai akurasi berada dibawah 65%. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.29 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.29 Hasil Evaluasi Pengukuran Percobaan *Optimizer* Pada Data Test Subtask B**

<b>Optimizer</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>Adam</b>	80.51	79.57	79.89
<b>Adadelta</b>	58.76	56.90	56.74
<b>SGD</b>	58.01	59.38	57.98
<b>RMSprop</b>	80.06	79.82	79.75

Pada percobaan yang dilakukan disimpulkan bahwa *optimizer* terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah *optimizer* Adam. *Optimizer* Adam mengungguli nilai *accuracy* dan *f1-score* dari seluruh *optimizer* pada data test, sementara *average recall* terbaik dimiliki *optimizer* RMSprop



**Gambar 15.10 Perbandingan Akurasi pada Percobaan *Optimizer* Subtask B**

Berdasarkan Gambar 6.10 *optimizer* Adam memiliki performa yang paling baik. Akurasi dapat meningkat hingga 20.65 % jika menggunakan *optimizer* Adam. Sehingga *optimizer* Adam akan digunakan menjadi *baseline* parameter *optimizer* pada skenario berikutnya.

### 6.7.3.3 Pengaruh *Learning Rate*

Skenario ketiga adalah menguji efek perubahan *learning rate* terhadap akurasi model. Nilai *learning rate* yang digunakan adalah 0.0005, 0.001, 0.002, 0.005, dan 0.01 . Output dari skenario ini adalah nilai *learning rate* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.30 Hasil Akurasi Percobaan *Learning Rate* Pada Data *Train Subtask A***

<b>Data / Learning Rate</b>	<b>0.0005</b>	<b>0.001</b>	<b>0.002</b>	<b>0.005</b>	<b>0.01</b>
<b>Train - Fold 1 (%)</b>	77.50	77.88	78.83	76.75	55.39
<b>Train - Fold 2 (%)</b>	79.77	80.15	80.34	78.45	62.19
<b>Train - Fold 3 (%)</b>	77.69	78.64	78.64	77.32	75.80
<b>Train - Fold 4 (%)</b>	74.48	73.91	72.40	71.83	45.75
<b>Train - Fold 5 (%)</b>	78.83	77.13	77.69	76.94	57.47
<b>All Train Fold (%)</b>	77.66	77.54	77.58	76.26	59.32

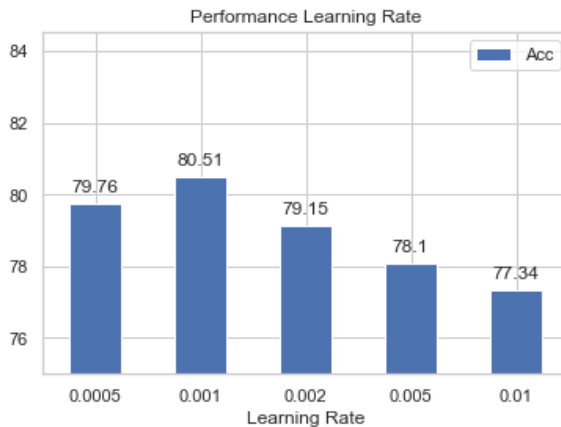
Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. *Learning rate* 0.0005 merupakan *learning rate* yang memiliki akurasi terbaik tetapi *learning rate* ini hanya unggul 0.08% dari *learning rate* 0.002

untuk keseluruhan data fold. *learning rate* 0.0005 unggul pada fold 4 dan 5 sementara *learning rate* 0.002 unggul pada fold 1,2 dan 3. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.31 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.31 Hasil Evaluasi Pengukuran Percobaan *Learning Rate* Pada Data Test Subtask B**

<b>Learning Rate</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>0.0005</b>	79.76	78.62	79.00
<b>0.001</b>	80.51	79.57	79.89
<b>0.002</b>	79.15	78.57	78.69
<b>0.005</b>	78.10	77.44	77.57
<b>0.01</b>	77.34	77.14	77.02

Pada percobaan yang dilakukan disimpulkan bahwa *learning rate* terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-Score* adalah 0.001. *Learning rate* 0.001 mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh nilai *learning rate* pada data test.



**Gambar 15.11 Perbandingan Akurasi pada Percobaan *Learning Rate* Subtask B**

Berdasarkan Gambar 6.11 nilai *learning Rate* 0.001 memiliki performa yang paling baik. Dapat dilihat juga bahwa semakin besar nilai *learning rate* maka akurasi semakin jelak. nilai *learning Rate* dapat meningkat akurasi hingga 3.17 % jika menggunakan nilai *learning Rate* 0.001. Sehingga nilai *learning Rate* 0.001 akan digunakan menjadi *baseline* parameter *learning Rate* pada skenario berikutnya.

#### 6.7.3.4 Pengaruh Dropout

Skenario keempat adalah menguji efek perubahan nilai *dropout* terhadap akurasi model. Nilai *dropout* yang digunakan adalah 0, 0.1, 0.2, 0.3, 0.4 dan 0.5 . Output dari skenario ini adalah nilai *dropout* terbaik. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *f1-score*. Berikut hasil dari skenario kedua untuk subtask A pada data *train-dev set* dan *test set*.

**Tabel 15.32 Hasil Akurasi Percobaan Dropout Pada Data Train Subtask B**

<b>Data / Learning Rate</b>	<b>0</b>	<b>0.1</b>	<b>0.2</b>	<b>0.3</b>	<b>0.4</b>	<b>0.5</b>
<b>Train - Fold 1 (%)</b>	77.50	78.45	79.02	79.02	77.69	77.88
<b>Train - Fold 2 (%)</b>	81.10	81.10	81.66	81.66	80.34	80.15
<b>Train - Fold 3 (%)</b>	77.50	78.45	77.69	77.88	78.45	78.64
<b>Train - Fold 4 (%)</b>	75.05	74.67	74.48	74.10	74.10	73.91
<b>Train - Fold 5 (%)</b>	76.18	78.45	79.96	78.07	78.26	77.13
<b>All Train Fold (%)</b>	77.47	78.22	78.56	78.15	77.77	77.54

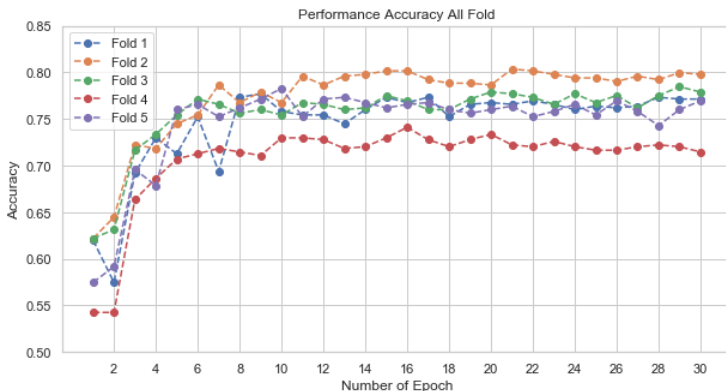
Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. Nilai *dropout* 0.2 merupakan nilai *dropout* yang memiliki akurasi terbaik. Untuk menentukan

model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.33 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.33 Hasil Evaluasi Pengukuran Percobaan Dropout Pada Data Test Subtask B**

Dropout	Accuracy (%)	AvgRec (%)	F1-Score (%)
0	80.06	80.23	79.88
0.1	80.21	79.47	79.68
0.2	80.36	79.56	79.80
0.3	80.21	80.20	79.98
0.4	81.12	80.84	80.80
0.5	80.51	79.57	79.89

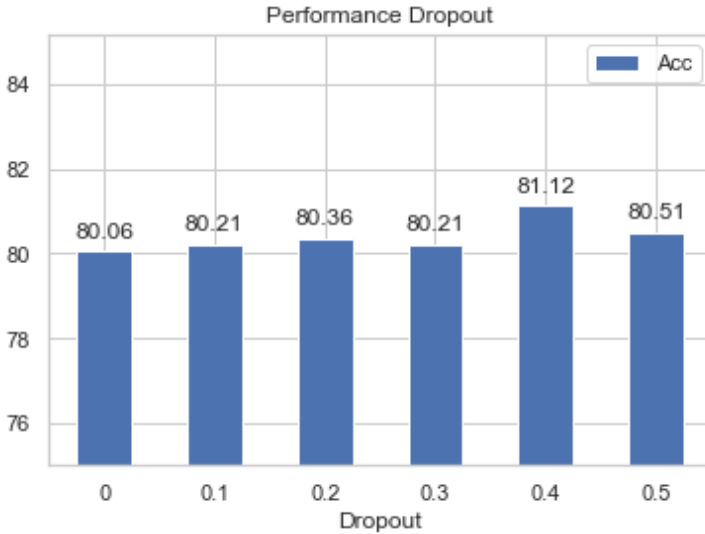
Pada percobaan yang dilakukan disimpulkan bahwa nilai *dropout* terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah 0.4. nilai dropout 0.4 mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari seluruh nilai *dropout* pada data test.



**Gambar 15.12 Performa Dropout Terbaik Subtask B**

Pada Gambar 6.12 Jika dilihat dari performa yang dihasilkan pada *dropout* 0.4 berdasarkan perubahan setiap epoch pada tiap

fold, kebanyakan fold memiliki *accuracy* yang meningkat tajam pada *epoch* 2-7. Sementara kebanyakan fold akan mencapai puncak nilai *accuracy* pada *epoch* 10-20. Pada *dropout* 0.4 seluruh fold memiliki nilai *accuracy* yang stabil pada *epoch* 12++.



**Gambar 15.13 Perbandingan Akurasi pada Percobaan *Dropout* Subtask B**

Berdasarkan Gambar 6.13 Nilai *dropout* 0.4 memiliki performa yang paling baik. Dapat dilihat juga bahwa semakin kecil nilai *dropout* pada kisaran nilai 0.2 sampai 0 maka akurasi semakin jelek. Sementara pada nilai *dropout* 0.3 keatas didapatkan hasil akurasi yang naik turun. Nilai *dropout* dapat meningkatkan *accuracy* hingga 1.06 % jika menggunakan nilai *dropout* 0.4. Sehingga nilai *dropout* 0.4 akan digunakan menjadi *baseline* parameter *dropout* pada skenario berikutnya.



### 6.7.3.5 Pengaruh Character Embedding

Skenario terakhir adalah menguji efek penambahan *character embedding* pada pembangunan model HAPN terhadap akurasi model. *Character embedding* yang digunakan dibangun menggunakan model neural network CNN dan LSTM. Output dari skenario ini adalah memilih model terbaik antara model HAPN dengan *character embedding* atau tidak. Pengukuran utama yang digunakan dalam skenario ini adalah *accuracy* dan dibantu dengan pengukuran lainnya yaitu *average recall* dan *F1-score*. Berikut hasil dari skenario kedua untuk subtask B pada data *train-dev set* dan *test set*.

**Tabel 15.34 Hasil Akurasi Percobaan *Character Embedding* Pada Data Train Subtask B**

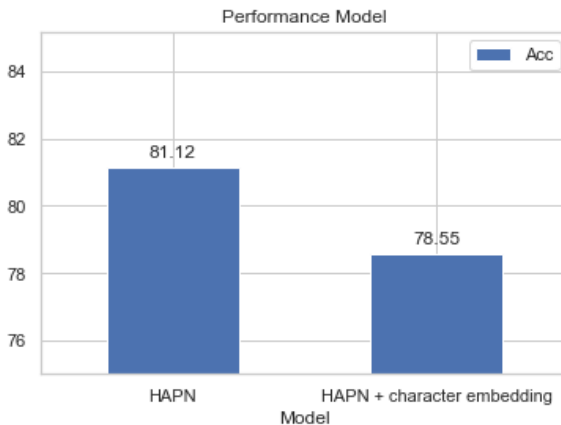
Data / Model HAPN	HAPN	HAPN + <i>Character Embedding</i>
<b>Train - Fold 1 (%)</b>	77.69	78.45
<b>Train - Fold 2 (%)</b>	80.34	81.29
<b>Train - Fold 3 (%)</b>	78.45	79.21
<b>Train - Fold 4 (%)</b>	74.10	74.86
<b>Train - Fold 5 (%)</b>	78.26	76.37
<b>All Train Fold (%)</b>	77.77	78.03

Dari hasil percobaan tersebut, data train fold 2 memiliki akurasi yang paling tinggi untuk setiap model, sementara data train fold 4 memiliki akurasi paling rendah. Model HAPN + *character embedding* memiliki nilai akurasi yang lebih tinggi untuk keseluruhan data fold sebesar 0.26%. Model HAPN HAPN + *character embedding* unggul pada fold 1-4 sementara pada fold 5 model HAPN memili akurasi tertinggi. Untuk menentukan model terbaik pada skenario ini dilakukan pengujian kembali pada data test. Tabel 6.15 merupakan hasil evaluasi pengukuran pada data test.

**Tabel 15.35 Hasil Evaluasi Pengukuran Percobaan *Character Embedding Test Subtask B***

<b>Model HAPN</b>	<b>Accuracy (%)</b>	<b>AvgRec (%)</b>	<b>F1-Score (%)</b>
<b>HAPN</b>	81.12	80.84	80.80
<b>HAPN + <i>Character Embedding</i></b>	78,55	78.27	78.38

Pada percobaan yang dilakukan disimpulkan bahwa model terbaik dengan *baseline* perhitungan evaluasi menggunakan pengukuran *accuracy*, *average recall*, dan *f1-score* adalah *model HAPN* yang mengungguli nilai *accuracy*, *average recall*, dan *f1-score* dari model HAPN + *character embedding*.



**Gambar 15.14 Perbandingan Akurasi pada Percobaan *Character Embedding Subtask B***

Berdasarkan Gambar 6.14 Model *HAPN* memiliki performa yang paling baik. Model HAPN dapat meningkat *accuracy* hingga 2.57%. Sehingga dari hasil pengukuran evaluasi model terbaik untuk subtask B adalah model HAPN.

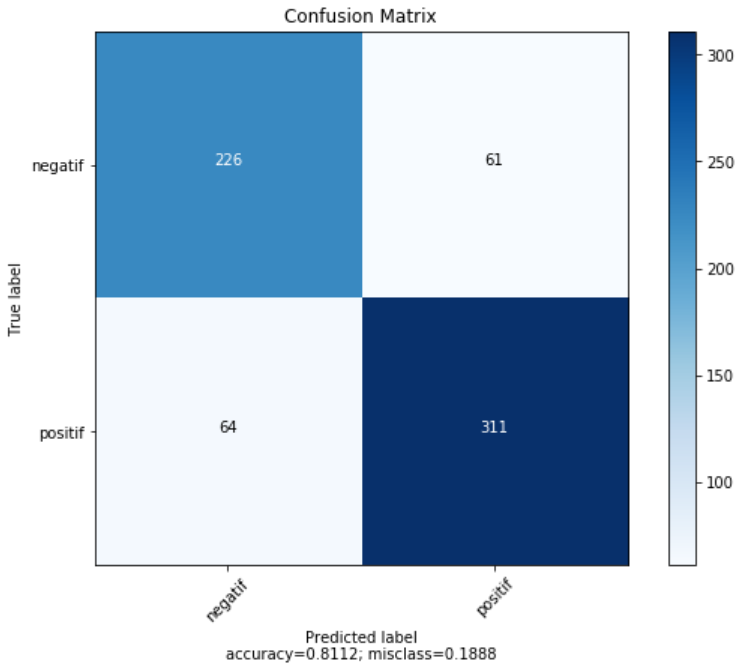
### 6.7.3.6 Pembahasan Hasil Subtask B

Dari hasil subtask B didapatkan performa terbaik yang dapat dilihat sebagai berikut:

**Tabel 15.36 Pengukuran Evaluasi Model Terbaik dari Subtask B**

Parameter	Nilai	Accuracy (%)	AvgRec (%)	F1-Score (%)
<b>Model Neural Network</b>	LSTM	81.12	80.84	80.80
<b>Optimizer</b>	Adam			
<b>Learning Rate</b>	0.001			
<b>Dropout</b>	0.4			
<b>Character Embedding</b>	-			

Selanjutnya untuk kinerja dari model terbaik dalam melakukan pengklasifikasian data akan dilihat berdasarkan *confusion matrix*. Hasil *confusion matrix* dapat dilihat pada Gambar 6.15. Dapat dilihat bahwa, kesalahan dalam melakukan klasifikasi untuk label positif dan negatif cukup kecil dilihat dari perbandingan kebenaran model dalam melakukan klasifikasi.



**Gambar 15.15 Confusion Matrix Model Terbaik Subtask B**

Pengecekan selanjutnya dilihat berdasarkan penyebaran nilai *recall* dan *f1-score* pada tiap labelnya. Setelah dilakukan perhitungan maka dihasilkan data sebagai berikut :

**Tabel 15.37 Perhitungan Recall dan F1-Score per Label pada Model Terbaik Subtask B**

Label	Recall	F1-Score
<b>Negatif</b>	79.0	78.0
<b>Positif</b>	83.0	83.0

Dari hasil perhitungan yang dilakukan pada subtask B, nilai *recall* paling rendah dalam proses klasifikasi adalah label negative dimana nilai terendahnya adalah 79 sementara nilai tertingginya yaitu *recall* positif dengan nilai 83.0. Hal ini membuktikan bahwa klasifikasi label negatif dan positif sudah cukup baik.

#### 6.7.4 Analisis Antar Subtask

Berdasarkan dari seluruh skenario yang dijalankan, subtask B memiliki nilai akurasi tertinggi diantara subtask lainnya, dimana model terbaik subtask B memiliki nilai akurasi 81.12 %, sedangkan untuk subtask A memiliki nilai akurasi 70.84 %. Hasil akurasi tersebut didapat dari model terbaik disetiap subtask dengan konfigurasi yang berbeda. Berikut tabel konfigurasi model terbaik untuk setiap subtask.

**Tabel 15.38 Perbandingan Konfigurasi Model Terbaik**

Subtask	A	B
<b>Model <i>Neural Network</i></b>	LSTM	LSTM
<b><i>Optimizer</i></b>	Adam	Adam
<b><i>Learning Rate</i></b>	0.001	0.001
<b><i>Dropout</i></b>	0.5	0.4
<b><i>Character Embedding</i></b>	-	-

Dengan masing masing hasil pengukuran sebagai berikut ini.

**Tabel 15.39 Hasil Evaluasi Pengukuran**

Subtask	Accuracy (%)	AvgRec (%)	F1-Score (%)
<b>A</b>	70.84	62.87	74.64
<b>B</b>	81.12	80.84	80.80

Berdasarkan hasil tersebut, salah satu faktor yang membuat akurasi untuk subtask B adalah yang terbaik merupakan sedikitnya jumlah label yang harus ditentukan. Subtask B hanya memprediksi *tweet* apakah sentimen aspek yang dimiliki merupakan positif atau negatif, sehingga peluang untuk melakukan klasifikasi di kelas yang benar semakin besar yaitu 50 %. Selain itu dengan jumlah kelas yang sedikit maka model akan lebih mudah mendapatkan ‘fitur’ untuk menentukan apakah itu merupakan kelas negatif atau kelas positif dari dataset yang diberikan. Jumlah dataset yang berbeda mungkin mempengaruhi, namun tidak dapat dipastikan didalam penelitian ini karena subtask A menggunakan dataset dengan jumlah label negatif 1912 dan label positif 1395 dengan total

data 3307 data, dimana subtask lain menggunakan jumlah data 3875.

### 6.7.5 Uji Signifikansi

Berdasarkan pada penelitian yang dilakukan dengan menggunakan perubahan parameter sesuai skenario pada subtask A dan subtask B telah didapatkan model terbaik yaitu dapat dilihat pada tabel 6.38 pada subbab sebelumnya. Sehingga dilakukan proses uji signifikansi untuk menentukan apakah hipotesis yang dibuat akan diterima atau ditolak dari hasil percobaan yang dilakukan pada subtask A dan subtask B.

Terdapat dua hipotesis yang dibuat yaitu hipotesis nul ( $H_0$ ) dan hipotesis alternative ( $H_A$ ). Berikut merupakan proses uji signifikansi yang dilakukan pada subtask A dan subtask B. Model dengan parameter terbaik yang dihasilkan pada subtask A dan subtask B akan dibandingkan dengan konfigurasi parameter awal dan dilakukan proses training sebanyak 15.

#### 6.7.5.1 Uji Signifikansi Subtask A

**Tabel 15.40 Data Percobaan Uji Signifikansi Subtask A**

Percobaan	Model Terbaik	Model Awal
1	70.06	69.68
2	70.84	69.81
3	70.45	70.58
4	70.84	68.81
5	70.45	69.81
6	72.13	69.42
7	70.06	70.19
8	69.94	70.45
9	71.10	69.68
10	70.45	69.81
11	70.45	69.94
12	71.87	69.29
13	71.23	69.81
14	70.58	69.94
15	70.84	69.81

Level signifikan : 0.05

$H_0$  : Tidak ada perbedaan antara model konfigurasi parameter terbaik dengan model konfigurasi parameter awal

$H_A$  : Ada perbedaan antara model konfigurasi parameter terbaik dengan model konfigurasi parameter awal

Proses perhitungan signifikansi dapat dilihat pada perhitungan berikut :

$$\mu_s = 70.752 \quad s_s^2 = 0.124097$$

$$\mu_{ns} = 69.802 \quad s_{ns}^2 = 0.217941$$

$$t = 5.333153 \quad df = 28$$

Berdasarkan hasil perhitungan yang dilakukan didapatkan nilai  $p$ -value  $< 0.0001$ . Hasil yang didapatkan adalah signifikan karena nilai  $p < 0.05$  atau  $H_0$  ditolak. Maka, dapat disimpulkan terdapat cukup bukti untuk menyatakan bahwa ada perbedaan signifikan antara konfigurasi parameter terbaik dengan model konfigurasi parameter awal dengan hasil nilai rata-rata model konfigurasi parameter terbaik lebih baik daripada model konfigurasi parameter awal.

### 6.7.5.2 Uji Signifikansi Subtask B

**Tabel 15.41 Data Percobaan Uji Signifikansi Subtask B**

Percobaan	Model Terbaik	Model Awal
1	80.51	79.91
2	80.51	79.91
3	80.21	79.61
4	81.57	80.06
5	81.12	80.66
6	80.51	80.66
7	81.12	79.91
8	80.51	80.06
9	80.51	79.91

Percobaan	Model Terbaik	Model Awal
10	80.51	79.00
11	80.51	80.06
12	80.97	79.91
13	80.51	80.21
14	80.82	79.76
15	80.82	79.00

Level signifikan : 0.05

$H_0$  : Tidak ada perbedaan antara model konfigurasi parameter terbaik dengan model konfigurasi parameter awal

$H_A$  : Ada perbedaan antara model konfigurasi parameter terbaik dengan model konfigurasi parameter awal

Proses perhitungan signifikansi dapat dilihat pada perhitungan berikut :

$$\mu_s = 80.714 \quad s_s^2 = 0.395221$$

$$\mu_{ns} = 79.909 \quad s_{ns}^2 = 0.188289$$

$$t = 4.820031 \quad df = 28$$

Berdasarkan hasil perhitungan yang dilakukan didapatkan nilai  $p$ -value  $< 0.0001$ . Hasil yang didapatkan adalah signifikan karena nilai  $p < 0.05$  atau  $H_0$  ditolak. Maka, dapat disimpulkan terdapat cukup bukti untuk menyatakan bahwa ada perbedaan signifikan antara konfigurasi parameter terbaik dengan model konfigurasi parameter awal dengan hasil nilai rata-rata model konfigurasi parameter terbaik lebih baik daripada model konfigurasi parameter awal.



## BAB VII KESIMPULAN DAN SARAN

Pada bab ini dibahas mengenai kesimpulan dari semua proses yang telah dilakukan dan saran yang dapat diberikan untuk pengembangan yang lebih baik

### 7.1. Kesimpulan

Kesimpulan yang didapatkan dari proses pengerjaan tugas akhir yang telah dilakukan antara lain :

1. Penelitian ini membuktikan bahwa *tuning* parameter sangatlah penting. Menggunakan parameter terbaik dalam pembuatan model analisis sentimen dapat meningkatkan akurasi hingga 22.19 % pada subtask A dan 23.11% pada Subtask B.
2. *Optimizer* memiliki pengaruh terbesar terhadap akurasi sehingga harus diperhatikan dengan baik. Dimana *optimizer* terbaik dalam pembuatan model HAPN adalah Adam.
3. *Learning rate* memiliki pengaruh besar juga terhadap akurasi, namun tidak sebesar *optimizer*. Dimana semakin besar nilai *learning rate* maka akurasi akan semakin menurun. Dimana nilai *learning rate* terbaik dalam pembuatan model HAPN adalah 0.001.
4. Model *Recurrent Neural Network* terbaik untuk model HAPN adalah *Long Short Term Memory*.
5. Semakin tinggi nilai *dropout* pada rentang 0 sampai 0.5 memungkinkan nilai akurasi yang semakin tinggi. Pada subtask A nilai *dropout* terbaik adalah 0.5 sementara pada subtask B nilai *dropout* terbaik adalah 0.4.
6. Penggunaan *Character Embedding* dalam Model HAPN masih belum mampu meningkatkan akurasi model.
7. Semakin banyak jumlah label dalam proses pelatihan membuat nilai evaluasi pengukuran semakin menurun. Hal ini dibuktikan dari nilai akurasi subtask B yang paling tinggi dibanding subtask A. Hal ini dikarenakan oleh

kompleksitas yang semakin tinggi dalam proses klasifikasi untuk menentukan label suatu teks.

8. Berdasarkan hasil klasifikasi *confusion matrix* pada subtask A disimpulkan bahwa kesalahan prediksi label paling sering terjadi pada label netral di mana hasil yang didapatkan menunjukkan label di prediksi menjadi label positif atau negatif. Hal ini dipengaruhi juga oleh jumlah label netral yang sedikit dibandingkan dengan model lain. Sehingga model kesulitan memelajari data untuk label netral.
9. Keseimbangan label atau sentimen pada data *training* sangat berpengaruh pada hasil performa evaluasi model HAPN.

## 7.2. Saran

Dari pengerjaan tugas akhir ini, adapun beberapa saran untuk pengembangan penelitian ke depan.

1. Jenis kata unik yang ada pada model *word embedding* dirasa masih kurang untuk digunakan karena masih terdapat 12% dari total kata yang tidak dapat ditemukan, sehingga akan lebih baik untuk menambah data pada model *word embedding* atau mencoba *word embedding* lain selain *Word2Vec* seperti *Fasttext* atau *GloVe*.
2. Menggunakan kombinasi *optimizer*, *learning rate*, *dropout* yang lebih variatif, sehingga mendapatkan penemuan yang baru.
3. Memperbanyak perulangan *training* seluruh skenario model, sehingga hasil pengujian lebih akurat.
4. Melihat pengaruh pemrosesan data text seperti *stemming* terhadap akurasi model.
5. Dataset yang digunakan harus memiliki label yang seimbang agar memiliki performa yang lebih baik untuk semua sentimen yang ada, tidak hanya pada satu jenis sentimen saja yang memiliki performa baik.

6. Dalam proses pelatihan *5-fold cross validation* dapat dilihat penyebaran data pada proses *training* tiap foldnya untuk melihat kesalahan prediksi yang sering terjadi. Sehingga dataset dapat dievaluasi kembali.

*Halaman ini sengaja dikosongkan*

## DAFTAR PUSTAKA

- [1] Asosiasi Penyelenggara Jasa Internet Indonesia - APJII, "Penetrasi & Perilaku Pengguna Internet Indonesia - Survey 2017," 2017.
- [2] Budiyono, "Media Sosial dan Komunikasi Politik: Media Sosial sebagai Komunikasi Politik Menjelang PILKADA DKI JAKARTA 2017" *Jurnal Komunikasi* Volume 11, Nomor 1, 2016.
- [3] L. Jemadu, "Jelang Pemilu 2019, Partai Mana Paling Top di Media Sosial?," *SUARA*,. Tersedia pada <https://www.suara.com/tekno/2019/03/18/082212/jelang-pemilu-2019-partai-mana-paling-top-di-media-sosial> , Diakses: 28-Mei-2019, 2019.
- [4] Liu B, "Sentiment analysis: mining opinions, sentiments, and emotions," *The Cambridge University Press*, 2015.
- [5] A. Farman, K. Kyung-Sup, dan K. Yong-Gi, "Opinion mining based on fuzzy domain ontology and Support Vector Machine: A proposal to automate online review classification," *Elsevier*, 47:235-250, 2016.
- [6] L. Li, Y. Liu, dan A. Zhou, "Hierarchical Attention Based Position-aware Network for Aspect-level Sentiment Analysis," *Proceedings of the 22nd Conference on Computational Natural Language Learning (CoNLL)*, 2018.
- [7] Y. Goldberg , "A primer on neural network models for natural language processing," *Journal of Artificial Intelligence Research*, 2016.
- [8] T Lin, dkk., "Learning LongTerm Dependencies in NARX Recurrent Neural Networks, Transactions on neural networks", *IEEE Transactional on Neural Network*, 1996.
- [9] Chen P, dkk. "Recurrent Attention Network on Memory for Aspect Sentiment Analysis," *Proceedings of the*

- 2017 *Conference on Empirical Methods in Natural Language Processing*, Copenhagen Denmark. 2017.
- [10] P. W. A. Wianto, “Analisis Sentimen Teks Bahasa Indonesia Pada Media Sosial Menggunakan Algoritma Convolutional Neural Network (Studi Kasus : Politik)” Tugas Akhir S1 Sistem Informasi ITS. 2018.
- [11] J. Brownlee, “What Is Natural Language Processing?,” *Machine Learning Mastery*,. Tersedia pada <https://machinelearningmastery.com/naturallanguage-processing/>. Diakses: 28-Maret-2019, 2017.
- [12] D. M. Neves, “Introduction to Natural Language Processing,” hal. 1–25, 2016.
- [13] L. Zhang, S. Wang, B. Liu, “Deep Learning for Sentiment Analysis: A Survey,” *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 8(4) e1253, 2018.
- [14] B. Pang dan L. Lee. “Opinion mining and sentiment analysis,” *Foundations and Trends in Information Retrieval*, 2008.
- [15] Y. Bengio, A. Courville, P. Vincent, “Representation learning: a review and new perspectives,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013.
- [16] Y. Bengio, P. Simard, P. Frasconi “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, 1994.
- [17] M. Schuster, K.K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, 1997.
- [18] J. Chung, C. Gulcehre, K. Cho, Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” *NIPS Deep Learning and Representation Learning Workshop*, 2014.
- [19] S. Guizhu, dkk.”Deep Learning with Gated Recurrent Unit Networks for Financial Sequence Predictions”. *8th International Congress of Information and Communication Technology*, 2018.

- [20] D. Bahdanau, K. Cho, Y. Bengio. “Neural Machine Translation By Jointly Learning To Align And Translate,” *arXiv preprint arXiv:1409.0473*, 2014.
- [21] R. Collobert, dkk. “Natural language processing (almost) from scratch,” *Journal of Machine Learning Research*, 2011.
- [22] T. Mikolov, G. Corrado, K. Chen, dan J. Dean, “Efficient Estimation of Word Representations in Vector Space,” *Proc. Int. Conf. Learn. Represent. (ICLR 2013)*, pp. 1–12, 2013.
- [23] T. Mikolov, K. Chen, G. Corrado, dan J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” *CrossRef List. Deleted DOIs, vol. 1*, pp. 1–9, 2013.
- [24] K. Cho, dkk. “Learning Phrase Representations Using Rnn Encoderdecoder For Statistical Machine Translation,” *arXiv preprint arXiv:1406.01078*, 2014.
- [25] S. Zheng dan R. Xia, “Left-Center-Right Separated Neural Network for Aspect-based Sentiment Analysis with Rotatory Attention,” *Arxiv preprint arXiv:1802.00892*, 2018.
- [26] T. Shen, dkk. “DiSAN: Directional Self-Attention Network for RNN/CNN-Free Language Understanding,” *arXiv preprint arXiv: 1709.04696*, 2017.
- [27] X. Zhang, Y. LeCun. “Text Understanding from Scratch,” *arXiv preprint arXiv: 1502.01710*, 2016.
- [28] E. Ma, “Besides Word Embedding, why you need to know Character Embedding?,” *Toward Data Science Medium*, Tersedia pada <https://towardsdatascience.com/besides-word-embedding-why-you-need-to-know-character-embedding-6096a34a3b10/>. Diakses: 11-Januari-2020, 2020.
- [29] C.J. Van Rijsbergen, “Information Retrieval” *London, UK: Butterworths*. 1975.

- [30] S. Rosenthal, N. Farra, P. Nakov, “SemEval-2017 Task 4: Sentiment Analysis in Twitter,” *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017) Association for Computational Ling*, 2017.
- [31] V. Qazvinian, E. Rosengren, D. R. Radev, and Q. Mei, “Rumor has it: Identifying Misinformation in Microblogs,” *Conf. Empir. Methods Nat. Lang. Process.*, pp. 1589–1599, 2011.
- [32] M. Budiardjo, “Dasar-Dasar Ilmu Politik,” Jakarta: PT Gramedia Pustaka Utama, Hlm 16, 2007.
- [33] S. Ramlan, “Memahami Ilmu Politik,” Jakarta: PT Grasindo, Hlm 18, 1992.



## BIODATA PENULIS



Penulis lahir di Gresik pada tanggal 08 Juli 1998. Merupakan anak ketujuh dari 7 bersaudara. Penulis telah menempuh beberapa pendidikan formal yaitu; SD Muhammadiyah Sidayu, SMP Negeri 1 Sidayu, dan SMA Negeri 1 Sidayu.

Pada Tahun 2016 pasca kelulusan SMA, penulis melanjutkan pendidikan di Departemen Sistem Informasi Fakultas Teknologi Informasi dan Komunikasi – Institut Teknologi Sepuluh Nopember (ITS) Surabaya dan terdaftar sebagai mahasiswa dengan NRP 0521164000029. Selama menjadi mahasiswa, penulis mengikuti berbagai kegiatan kemahasiswaan seperti beberapa kepanitian serta pernah menjabat sebagai Staf Departemen Riset dan Teknologi HMSI FTIK ITS dan pada tahun ketiga menjabat sebagai Staf Ahli Departemen Riset dan Teknologi Application HMSI FTIK ITS. Di bidang akademik, penulis aktif menjadi asisten dosen praktikum pada mata kuliah Penggalan Data. Selain itu, pada tahun 2019 penulis mengikuti perlombaan dan menjadi Juara 2 pada Lomba Penggalan Data GEMASTIK XII 2019 yang diadakan oleh Kemenristekdikti serta Juara 3 pada Lomba Data Science Competition IT TODAY 2019 yang diadakan oleh Departemen Ilmu Komputer FMIPA IPB. Penulis juga sempat magang sebagai *Business Intelligence Analyst* di GOJEK selama 2 bulan.

Pada tahun keempat, karena penulis memiliki ketertarikan di bidang pengolahan data, maka penulis mengambil bidang minat Akuisisi Data dan Diseminasi Informasi (ADDI). Penulis dapat dihubungi melalui *email* di [maulidaniw@gmail.com](mailto:maulidaniw@gmail.com).

*Halaman ini sengaja dikosongkan*

## LAMPIRAN A DATA BERLABEL

Tweet	Label	Aspek
jadi para ahlu ngopi dan narasi ini lg kembangkan fitnah bhw pengen diajak koalisi gt ya cari makan kok gt amat ya pdhal dari pilpres bahkan sebelumnya narasi pks selalu oposisi lho hidup dari bohong itu rasanya kayak gmn ya	Negatif	para ahlu ngopi dan narasi
tapi dengan kemungkinan besar gerindra jadi koalisi itu masalah sih oposisi jadi kempes ga akan cukup kuat buat head to head sama pemerintah ditambah lagi ketua dpr nya pdip masa jeruk makan jeruk	Negatif	oposisi
tapi dengan kemungkinan besar gerindra jadi koalisi itu masalah sih oposisi jadi kempes ga akan cukup kuat buat head to head sama pemerintah ditambah lagi ketua dpr nya pdip masa jeruk makan jeruk	Negatif	pdip
tapi dengan kemungkinan besar gerindra jadi koalisi itu masalah sih oposisi jadi kempes ga akan cukup kuat buat head to head sama pemerintah ditambah lagi ketua dpr nya pdip masa jeruk makan jeruk	Negatif	gerindra
kalau kasus munir udin novel dll gitu wilayah kementerian apa ya	Netral	kementerian
erick sindir orang orang yang tidak berkeringat ngebet masuk kabinet berasa banget paling berjasa buat jokowi yaks	Negatif	erick
ternyata 5 tahun yang lalu jokowi juga dilantik tgl 20 oktober loh baru nyadar	Neutral	jokowi
aku berharap pelantikan presiden berjalan lancar tak ada halangan sama sekali agar jokowi segera dapat mengumumkan kabinet kerja 2nya dan publik dapat kembali berharap dari kerja menteri yg dipilih kritik kembali mendapatkan posisinya kepada hal substantif bukan personal	Positif	pelantikan presiden
kapan ya saya bisa melihat hasil hitung kpu yang pemilu 2019 kok belm pernah denger serius yang bisa kasih tau terimakasih	Negatif	kpu

<b>Tweet</b>	<b>Label</b>	<b>Aspek</b>
memantau dng hikmat pengumuman kabinet menteri melalui timeline twitter karna seru disertai komentar dan sarkasme rakyat twitter terutama memantau krna sarkasnya selalu maksimal luv ibanku	Negatif	pengumuman kabinet menteri
ini kayaknya script pengumuman jokowi tadi dikirim ke grup whatsapp admin twitter resmi lembaga lembaga negara terus tinggal copy paste doang	netral	script pengumuman jokowi
area kerja kehutanan dilekatkan pada kementerian agraria dan tata ruang dan terpisah dg kementerian lingkungan hidup jd merasa konservasi hutan akan lbh dikesampingkan ya	Negatif	kementerian agraria dan tata ruang
jabatan menteri menjadi rebutan idaman para elit politik disitu tempat pengelolaan apbn tugas kpk tambah berat loh koq malah dibatasi geraknya	Negatif	jabatan menteri
jabatan menteri menjadi rebutan idaman para elit politik disitu tempat pengelolaan apbn tugas kpk tambah berat loh koq malah dibatasi geraknya	Negatif	elit politik
jujur cuma bisa senyum liat kabinet jokowi periode 2 sama menteri menterinya	Negatif	kabinet jokowi periode 2
jujur cuma bisa senyum liat kabinet jokowi periode 2 sama menteri menterinya	Negatif	menteri menterinya
kemaren ada seneng ada sedihnya gitu waktu jokowi ngumumin kabinetnya yg baru kabinet indonesia maju yaa kaya kehidupan kali ya ada ups and downsnya bu susi	Netral	jokowi
kemaren ada seneng ada sedihnya gitu waktu jokowi ngumumin kabinetnya yg baru kabinet indonesia maju yaa kaya kehidupan kali ya ada ups and downsnya bu susi	Netral	kabinet indonesia maju
kita laporkan ke mabes tni dulu lantas kita baru bisa menggugat jokowi aq punya bukti lengkap jangan takut kita muslim mayoritas disini tak ada kilafah bersamanya semua hanya omong kosong orang cina masuk treos sampek detik ini	Negatif	orang cina
kita laporkan ke mabes tni dulu lantas kita baru bisa menggugat jokowi aq punya bukti lengkap jangan takut kita muslim mayoritas disini tak ada kilafah bersamanya semua	Negatif	jokowi

<b>Tweet</b>	<b>Label</b>	<b>Aspek</b>
hanya omong kosong orang cina masuk treos sampek detik ini		
kita laporkan ke mabes tni dulu lantas kita baru bisa menggugat jokowi aq punya bukti lengkap jangan takut kita muslim mayoritas disini tak ada kilafah bersamanya semua hanya omong kosong orang cina masuk treos sampek detik ini	Positif	muslim mayoritas
semoga kabinet pemerintahan pak jokowi part 2 bisa menjawab tantangan amp pertumbuhan signifikan indonesia ke depan indonesia jaya	Positif	kabinet pemerintahan pak jokowi
soal kabinet kerja siapa yang akan dipanggil jadi menteri nantinya kita sebagai rakyat cukup mendukung dan berfikir positif saja kalau bukan ahli pertanian jadi menteri perdagangan ya mungkin ahli perdagangan jadi menteri pertanian saya kira sekitar itulah	Positif	kabinet kerja
jokowi sangat menghormati dunia santri bagi saya ini investasi moral untuk masa depan	Positif	jokowi
harap harap cemas bu susi bu retno amp pak jonan akan dipanggil sbgi calon menteri msh berharap beliau beliau ttp lanjut di periode ke 2 pak jokowi	Positif	bu susi
welcome back jokowi saya yakin periode2 ini bapak menunjukan kualitas asli bapak	Positif	jokowi
kemarin aku sempat kecewa berat saat masuknya pak prabowo gerindra dlm kabinet krn itu aku tdklagi peduli dg akun2 yg adaampign tutup akunku ini tp saat hari ini kulihat wajah pak wowo dlm pelantikan hatiku mulai bsa berdamai justru aku iba pdnyaampkini mendukung bliau	Positif	pak wowo
kemarin aku sempat kecewa berat saat masuknya pak prabowo gerindra dlm kabinet krn itu aku tdklagi peduli dg akun2 yg adaampign tutup akunku ini tp saat hari ini kulihat wajah pak wowo dlm pelantikan hatiku mulai bsa berdamai justru aku iba pdnyaampkini mendukung bliau	Negatif	gerindra
kemarin aku sempat kecewa berat saat masuknya pak prabowo gerindra dlm kabinet krn itu aku tdklagi peduli dg akun2 yg adaampign tutup akunku ini tp saat hari ini kulihat wajah pak wowo dlm pelantikan	Negatif	pak prabowo

Tweet	Label	Aspek
hatiku mulai bsa berdamai justru aku iba pdnyaampkini mendukung bliau		
bagi gw projo yg bilang pak jokowi sama kek kadrun berarti projo karbitan hanya karena menteri pilihan mereka tidak masuk kabinet	Negatif	projo
pak jokowi memperlihatkan bahwa kondisi politik indonesia baik2 saja apakah kedepannya sesuai dengan harapan	Positif	kondisi politik indonesia
jokowi masih mengecewakan krn memilih wamen dari eci pengkhianat musuh pribumi musuh ideologi pancasila dan musuh negara pancasila referensi negara adlh kekuasaan politik ini milik pribumi eci tdk msk unsur dlm bhinneka tunggal ika krn eci bukan bangsa indonesia	Negatif	jokowi
keberadaan mahfud md di kabinet indonesia maju menduduki jabatan menkopolhukam tidak untuk membawa atau merepresentasikan agenda sekelompok orang namun untuk mendukung visi misi pemerintah di periode kedua kepemimpinan presiden jokowi	Positif	mahfud md
hebatnya jokowi gitu org yg lo dewa2kan levelnya cmn pembantu dia nyesek ga ditinggalin gitu doang ama dia cape2 nyiptain hoax sana sini malah kena prank awowkwokwowkwowko	Positif	jokowi
saya kagum dengan sikap pks yg memilih jadi oposisi kemungkinan menyusul pan dan demokrat biarpun kecil secara kuantitas tapi dimata rakyat lebih berkualitas ketimbang gemuk tapi rapuh	Positif	demokrat
saya kagum dengan sikap pks yg memilih jadi oposisi kemungkinan menyusul pan dan demokrat biarpun kecil secara kuantitas tapi dimata rakyat lebih berkualitas ketimbang gemuk tapi rapuh	positif	pks

**LAMPIRAN B HASIL EVALUASI MODEL  
TERBAIK SUBTASK A**

Fold 1

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	1,0076	0,4927	1,0173	0,4500	0,3333	0,3103
<b>2</b>	0,9584	0,5234	0,9828	0,4871	0,3685	0,4298
<b>3</b>	0,9008	0,5827	0,9286	0,5452	0,4321	0,5850
<b>4</b>	0,8346	0,6391	0,8997	0,5726	0,4572	0,6201
<b>5</b>	0,7554	0,6847	0,8675	0,5790	0,4585	0,6198
<b>6</b>	0,6799	0,7226	0,8289	0,6065	0,4872	0,6608
<b>7</b>	0,6179	0,7516	0,8325	0,5855	0,4637	0,6294
<b>8</b>	0,6044	0,7605	0,8525	0,5919	0,4718	0,6316
<b>9</b>	0,5391	0,7907	0,8812	0,5855	0,4874	0,6122
<b>10</b>	0,5235	0,8113	0,8508	0,5903	0,5041	0,6139
<b>11</b>	0,4592	0,8218	0,8586	0,6226	0,5581	0,6359
<b>12</b>	0,4344	0,8335	0,8382	0,5919	0,5139	0,6133
<b>13</b>	0,4134	0,8528	0,8237	0,6403	0,5281	0,6956
<b>14</b>	0,4153	0,8577	0,8590	0,6129	0,5335	0,6311
<b>15</b>	0,3190	0,8750	0,8288	0,6387	0,5466	0,6868
<b>16</b>	0,2706	0,8944	0,8355	0,6387	0,5399	0,6909
<b>17</b>	0,2451	0,9020	0,8606	0,6403	0,5661	0,6804
<b>18</b>	0,2079	0,9222	0,9019	0,6403	0,5857	0,6836
<b>19</b>	0,2094	0,9222	0,9251	0,6339	0,5923	0,6735
<b>20</b>	0,3795	0,9044	1,0550	0,6258	0,5747	0,6739
<b>21</b>	0,2716	0,9085	0,8054	0,6387	0,5773	0,6927
<b>22</b>	0,2083	0,9254	0,8652	0,6323	0,5388	0,6697
<b>23</b>	0,1822	0,9282	0,8913	0,6419	0,5770	0,6716
<b>24</b>	0,1688	0,9379	0,8810	0,6532	0,5899	0,6937
<b>25</b>	0,1515	0,9480	0,9327	0,6468	0,5683	0,6862
<b>26</b>	0,1423	0,9492	0,9994	0,6532	0,5723	0,6914

<b>Epoch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>27</b>	0,1318	0,9476	0,9280	0,6532	0,5943	0,6992
<b>28</b>	0,1223	0,9520	0,9829	0,6645	0,5930	0,7084
<b>29</b>	0,1282	0,9484	0,9823	0,6500	0,5810	0,6849
<b>30</b>	0,1701	0,9484	0,9858	0,6306	0,5721	0,6632

## Fold 2

<b>Epoch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	1,0041	0,4972	0,9790	0,5774	0,4420	0,6071
<b>2</b>	0,9690	0,5157	0,9180	0,5677	0,3718	0,4670
<b>3</b>	0,8963	0,5855	0,8784	0,6161	0,4568	0,6351
<b>4</b>	0,8369	0,6306	0,8270	0,6323	0,4696	0,6542
<b>5</b>	0,7752	0,6746	0,8031	0,6484	0,5030	0,6886
<b>6</b>	0,7281	0,7085	0,8020	0,6500	0,4747	0,6648
<b>7</b>	0,6493	0,7319	0,7854	0,6661	0,5015	0,6950
<b>8</b>	0,5985	0,7544	0,7858	0,6774	0,5375	0,7072
<b>9</b>	0,5353	0,7871	0,7884	0,6806	0,5484	0,7019
<b>10</b>	0,5099	0,8125	0,7956	0,6790	0,5264	0,7022
<b>11</b>	0,5077	0,8286	0,7891	0,6823	0,5445	0,7127
<b>12</b>	1,5218	0,7464	0,8698	0,6758	0,5634	0,7045
<b>13</b>	0,7087	0,7810	0,9070	0,6016	0,4339	0,5582
<b>14</b>	0,7474	0,8044	0,8837	0,6532	0,5360	0,6729
<b>15</b>	0,8386	0,8032	1,1747	0,6097	0,5385	0,6297
<b>16</b>	1,4390	0,7625	1,4837	0,6452	0,5307	0,6762
<b>17</b>	1,4154	0,7823	3,1751	0,5952	0,4523	0,5188
<b>18</b>	3,1952	0,6895	3,8079	0,6258	0,4873	0,6111
<b>19</b>	3,6950	0,7133	4,5969	0,5403	0,4788	0,4938
<b>20</b>	3,2786	0,7476	5,5380	0,5613	0,4976	0,5794
<b>21</b>	3,8464	0,7242	5,8786	0,5823	0,4136	0,5731



<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>22</b>	3,7418	0,7202	4,5489	0,6177	0,5315	0,6233
<b>23</b>	2,9195	0,7653	4,8154	0,6177	0,5109	0,6446
<b>24</b>	2,8880	0,7758	4,8172	0,5903	0,5067	0,6167
<b>25</b>	2,6246	0,7907	5,1184	0,5339	0,5199	0,5837
<b>26</b>	2,5914	0,7835	3,8190	0,6419	0,5182	0,6428
<b>27</b>	2,4910	0,7988	4,8545	0,5903	0,4655	0,4993
<b>28</b>	2,9426	0,7702	5,3034	0,5855	0,5136	0,6252
<b>29</b>	2,7945	0,7935	5,3231	0,6016	0,4757	0,5777
<b>30</b>	2,8207	0,7875	4,4643	0,6226	0,4711	0,6150

Fold 3

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	1,0191	0,4661	0,9635	0,5290	0,3356	0,3550
<b>2</b>	0,9676	0,5222	0,9189	0,5274	0,3376	0,3752
<b>3</b>	0,9033	0,5831	0,8817	0,6210	0,4762	0,6563
<b>4</b>	0,8434	0,6415	0,8429	0,6242	0,4897	0,6660
<b>5</b>	0,7594	0,6790	0,8128	0,6452	0,4924	0,6781
<b>6</b>	0,7401	0,7137	0,8373	0,6468	0,5159	0,6922
<b>7</b>	0,7028	0,7371	0,7962	0,6516	0,4959	0,6839
<b>8</b>	0,5983	0,7556	0,8335	0,6565	0,5329	0,6967
<b>9</b>	0,5581	0,7730	0,8561	0,6242	0,5433	0,6759
<b>10</b>	0,5103	0,7907	0,8358	0,6387	0,5481	0,6775
<b>11</b>	0,5134	0,8250	0,8048	0,6548	0,5443	0,6905
<b>12</b>	0,4542	0,8290	0,8093	0,6629	0,5227	0,6997
<b>13</b>	0,3959	0,8431	0,8863	0,6452	0,5394	0,6852
<b>14</b>	0,3591	0,8790	0,8721	0,6516	0,5234	0,6764
<b>15</b>	0,3995	0,8698	0,8608	0,6419	0,5541	0,6857
<b>16</b>	0,2959	0,9004	0,8841	0,6419	0,5515	0,6881

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>17</b>	0,2716	0,9065	0,8718	0,6565	0,5640	0,6894
<b>18</b>	0,2894	0,9185	0,9328	0,6581	0,5238	0,6828
<b>19</b>	0,2342	0,9226	0,9556	0,6484	0,5616	0,6869
<b>20</b>	0,2001	0,9266	0,9652	0,6484	0,5431	0,6879
<b>21</b>	0,1797	0,9351	0,9751	0,6645	0,5485	0,6929
<b>22</b>	0,1597	0,9448	0,9755	0,6516	0,5606	0,6897
<b>23</b>	0,1536	0,9427	0,9997	0,6645	0,5536	0,7005
<b>24</b>	0,1524	0,9399	1,0584	0,6613	0,5356	0,6926
<b>25</b>	0,1332	0,9472	1,0518	0,6500	0,5495	0,6798
<b>26</b>	0,1319	0,9577	1,1040	0,6565	0,5397	0,6858
<b>27</b>	0,1259	0,9480	1,1193	0,6500	0,5455	0,6814
<b>28</b>	0,1199	0,9516	1,1236	0,6532	0,5465	0,6856
<b>29</b>	0,1109	0,9601	1,1162	0,6613	0,5461	0,6928
<b>30</b>	0,1030	0,9573	1,2174	0,6306	0,5403	0,6727

## Fold 4

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	1,0026	0,4919	1,0023	0,4565	0,3333	0,3134
<b>2</b>	0,9554	0,5315	0,9653	0,5194	0,3960	0,5187
<b>3</b>	0,8796	0,6028	0,9178	0,5790	0,4631	0,6308
<b>4</b>	0,8250	0,6468	0,9103	0,5758	0,4664	0,6280
<b>5</b>	0,7593	0,6931	0,8873	0,5806	0,4593	0,6288
<b>6</b>	0,6879	0,7323	0,8827	0,5839	0,4636	0,6334
<b>7</b>	0,6263	0,7565	0,9015	0,5806	0,4589	0,6242
<b>8</b>	0,5448	0,7827	0,8974	0,6081	0,4999	0,6577
<b>9</b>	0,4907	0,7927	0,9558	0,6048	0,5085	0,6582
<b>10</b>	0,4671	0,8202	0,9025	0,6000	0,4917	0,6411
<b>11</b>	0,4169	0,8347	0,9883	0,5790	0,5460	0,6341

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
12	0,3988	0,8613	1,0403	0,5935	0,5008	0,6223
13	0,3986	0,8528	0,9424	0,5694	0,5149	0,6338
14	0,3432	0,8819	0,9351	0,6113	0,5366	0,6592
15	0,2689	0,9032	1,0839	0,5468	0,5200	0,6031
16	0,2614	0,9065	0,9914	0,5984	0,5347	0,6444
17	0,2478	0,9141	1,0031	0,5935	0,5418	0,6421
18	0,2232	0,9210	1,0256	0,5839	0,5316	0,6227
19	0,2293	0,9347	1,2538	0,5968	0,5369	0,6392
20	0,6463	0,8734	1,3147	0,5435	0,5020	0,5604
21	1,6367	0,7899	2,9867	0,5677	0,4716	0,6136
22	3,7758	0,6992	5,2693	0,5484	0,4537	0,5932
23	4,1193	0,6827	7,3612	0,4581	0,4883	0,4222
24	4,4854	0,6810	6,3040	0,5694	0,4933	0,6098
25	4,0732	0,7153	6,2737	0,5565	0,4681	0,5949
26	3,8011	0,7323	6,2075	0,5613	0,4969	0,5932
27	4,4114	0,6984	6,0711	0,5629	0,4901	0,6023
28	3,3697	0,7661	6,1185	0,5677	0,5000	0,6074
29	3,7214	0,7500	6,5679	0,5403	0,4521	0,5636
30	3,4919	0,7569	6,4678	0,5548	0,4868	0,5926

Fold 5

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
1	1,0240	0,4677	0,9531	0,5129	0,3383	0,3579
2	0,9768	0,5161	0,9000	0,5145	0,3408	0,3730
3	0,9300	0,5589	0,9029	0,5355	0,3600	0,4310
4	0,8678	0,6177	0,8288	0,6355	0,4638	0,6589
5	0,7986	0,6653	0,8342	0,6258	0,4509	0,6387
6	0,7566	0,7073	0,8276	0,6645	0,4972	0,7011

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>7</b>	0,6610	0,7290	0,7985	0,6613	0,5012	0,6966
<b>8</b>	0,6382	0,7714	0,8244	0,6452	0,5122	0,6871
<b>9</b>	0,5480	0,7980	0,8530	0,6613	0,5203	0,6991
<b>10</b>	0,4628	0,8222	0,8286	0,6774	0,5553	0,7104
<b>11</b>	0,4395	0,8387	0,8213	0,6597	0,5663	0,7030
<b>12</b>	0,4374	0,8617	0,8007	0,6629	0,5591	0,7034
<b>13</b>	0,3388	0,8694	0,8258	0,6613	0,5583	0,6962
<b>14</b>	0,2814	0,8980	0,8476	0,6645	0,5566	0,6968
<b>15</b>	0,2544	0,9169	0,9142	0,6355	0,5711	0,6842
<b>16</b>	0,2291	0,9222	0,8809	0,6694	0,5667	0,7063
<b>17</b>	0,3951	0,9048	1,0951	0,5355	0,5253	0,5846
<b>18</b>	0,3214	0,8863	0,8863	0,6339	0,5558	0,6786
<b>19</b>	0,2248	0,9218	0,8810	0,6452	0,5545	0,6805
<b>20</b>	0,1856	0,9339	0,9561	0,6419	0,5611	0,6833
<b>21</b>	0,1638	0,9415	0,9867	0,6371	0,5672	0,6802
<b>22</b>	0,1470	0,9383	1,0006	0,6452	0,5639	0,6836
<b>23</b>	0,8870	0,8863	3,4297	0,5694	0,5198	0,5984
<b>24</b>	3,5593	0,7250	6,4634	0,5742	0,3987	0,5376
<b>25</b>	5,3519	0,6310	5,9740	0,6129	0,4655	0,6492
<b>26</b>	5,5754	0,6290	6,6422	0,5726	0,4295	0,5265
<b>27</b>	7,3083	0,5242	7,4691	0,5161	0,3741	0,5081
<b>28</b>	6,5165	0,5665	7,1083	0,5484	0,3761	0,4678
<b>29</b>	6,6766	0,5673	6,9807	0,5516	0,3789	0,4744
<b>30</b>	6,5237	0,5734	7,0587	0,5468	0,4311	0,5785

**LAMPIRAN C HASIL EVALUASI MODEL  
TERBAIK SUBTASK B**

Fold 1

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	0,6828	0,5747	0,6621	0,6200	0,5886	0,5651
<b>2</b>	0,6279	0,6300	0,6317	0,5747	0,5241	0,4124
<b>3</b>	0,5521	0,7198	0,5709	0,6919	0,6679	0,6625
<b>4</b>	0,4727	0,7750	0,5301	0,7297	0,7139	0,7155
<b>5</b>	0,3963	0,8341	0,5584	0,7127	0,6920	0,6907
<b>6</b>	0,3796	0,8313	0,5031	0,7524	0,7608	0,7524
<b>7</b>	0,3061	0,8677	0,5935	0,6938	0,6654	0,6552
<b>8</b>	0,2428	0,9055	0,4850	0,7732	0,7709	0,7707
<b>9</b>	0,2084	0,9083	0,5005	0,7769	0,7706	0,7724
<b>10</b>	0,1740	0,9291	0,5690	0,7580	0,7424	0,7451
<b>11</b>	0,1822	0,9192	0,5130	0,7543	0,7608	0,7541
<b>12</b>	0,1674	0,9343	0,5502	0,7543	0,7415	0,7442
<b>13</b>	0,1672	0,9315	0,5819	0,7448	0,7247	0,7256
<b>14</b>	0,1249	0,9471	0,5509	0,7599	0,7536	0,7551
<b>15</b>	0,1143	0,9518	0,6313	0,7732	0,7577	0,7611
<b>16</b>	0,1067	0,9532	0,5804	0,7675	0,7621	0,7633
<b>17</b>	0,0965	0,9612	0,6644	0,7732	0,7581	0,7615
<b>18</b>	0,0958	0,9575	0,6201	0,7524	0,7435	0,7456
<b>19</b>	0,0881	0,9617	0,6396	0,7656	0,7608	0,7618
<b>20</b>	0,0769	0,9674	0,6588	0,7675	0,7617	0,7631
<b>21</b>	0,0731	0,9698	0,6984	0,7656	0,7616	0,7622
<b>22</b>	0,0671	0,9745	0,7139	0,7694	0,7646	0,7656
<b>23</b>	0,0580	0,9787	0,7361	0,7656	0,7600	0,7613
<b>24</b>	0,0612	0,9759	0,7588	0,7599	0,7507	0,7531
<b>25</b>	0,0760	0,9674	0,6931	0,7637	0,7620	0,7614
<b>26</b>	0,0604	0,9778	0,6941	0,7618	0,7553	0,7569

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>27</b>	0,0542	0,9787	0,8293	0,7637	0,7525	0,7553
<b>28</b>	0,0521	0,9797	0,8112	0,7732	0,7668	0,7685
<b>29</b>	0,0429	0,9839	0,8777	0,7713	0,7610	0,7638
<b>30</b>	0,0423	0,9820	0,8581	0,7713	0,7684	0,7685

## Fold 2

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	0,6909	0,5586	0,6406	0,6219	0,5000	0,3834
<b>2</b>	0,6389	0,6248	0,5818	0,6446	0,5339	0,4606
<b>3</b>	0,5485	0,7094	0,5122	0,7221	0,6648	0,6701
<b>4</b>	0,4889	0,7557	0,5021	0,7183	0,6500	0,6513
<b>5</b>	0,4206	0,8190	0,4825	0,7448	0,6948	0,7032
<b>6</b>	0,3329	0,8445	0,4951	0,7543	0,7005	0,7098
<b>7</b>	0,2868	0,8875	0,4549	0,7864	0,7577	0,7649
<b>8</b>	0,2340	0,8998	0,5062	0,7675	0,7209	0,7314
<b>9</b>	0,2552	0,8979	0,5233	0,7788	0,7379	0,7484
<b>10</b>	0,2310	0,9249	0,5103	0,7675	0,7219	0,7322
<b>11</b>	0,2676	0,9102	0,4580	0,7958	0,7986	0,7887
<b>12</b>	0,1671	0,9367	0,5159	0,7864	0,7518	0,7611
<b>13</b>	0,1435	0,9423	0,5665	0,7958	0,7800	0,7816
<b>14</b>	0,1227	0,9504	0,5435	0,7977	0,7776	0,7815
<b>15</b>	0,1102	0,9608	0,5579	0,8015	0,7904	0,7896
<b>16</b>	0,1135	0,9532	0,5098	0,8015	0,7787	0,7842
<b>17</b>	0,0965	0,9551	0,5866	0,7921	0,7613	0,7697
<b>18</b>	0,0915	0,9627	0,5573	0,7883	0,7612	0,7678
<b>19</b>	0,0863	0,9608	0,5573	0,7883	0,7563	0,7648
<b>20</b>	0,0792	0,9688	0,6003	0,7864	0,7577	0,7649
<b>21</b>	0,0712	0,9716	0,6222	0,8034	0,7831	0,7874

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>22</b>	0,0615	0,9721	0,6793	0,8015	0,7757	0,7826
<b>23</b>	0,0710	0,9712	0,6659	0,7977	0,7707	0,7779
<b>24</b>	0,0734	0,9664	0,6184	0,7940	0,7628	0,7714
<b>25</b>	0,0665	0,9773	0,6735	0,7940	0,7687	0,7749
<b>26</b>	0,0610	0,9731	0,6950	0,7902	0,7588	0,7673
<b>27</b>	0,0551	0,9764	0,7361	0,7958	0,7771	0,7802
<b>28</b>	0,0690	0,9783	0,7471	0,7921	0,7613	0,7697
<b>29</b>	0,0566	0,9773	0,6679	0,7996	0,7664	0,7762
<b>30</b>	0,0656	0,9735	0,6882	0,7977	0,7815	0,7834

Fold 3

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	0,6929	0,5543	0,6452	0,6219	0,5194	0,4373
<b>2</b>	0,6397	0,6281	0,5854	0,6314	0,5387	0,4850
<b>3</b>	0,5572	0,7037	0,5247	0,7164	0,6906	0,6941
<b>4</b>	0,4592	0,7807	0,5043	0,7335	0,6964	0,7028
<b>5</b>	0,3899	0,8284	0,4916	0,7543	0,7206	0,7278
<b>6</b>	0,3324	0,8653	0,4711	0,7713	0,7694	0,7635
<b>7</b>	0,2631	0,8941	0,5307	0,7656	0,7468	0,7498
<b>8</b>	0,2429	0,9003	0,5287	0,7561	0,7140	0,7226
<b>9</b>	0,1942	0,9197	0,6417	0,7599	0,7323	0,7383
<b>10</b>	0,1907	0,9301	0,4935	0,7543	0,7313	0,7355
<b>11</b>	0,1494	0,9362	0,5254	0,7675	0,7609	0,7578
<b>12</b>	0,1336	0,9457	0,5634	0,7656	0,7639	0,7578
<b>13</b>	0,1326	0,9499	0,5382	0,7599	0,7350	0,7401
<b>14</b>	0,1195	0,9542	0,5633	0,7618	0,7500	0,7496
<b>15</b>	0,1121	0,9518	0,5533	0,7750	0,7545	0,7586
<b>16</b>	0,1003	0,9608	0,5684	0,7694	0,7553	0,7562

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>17</b>	0,0864	0,9664	0,6104	0,7599	0,7422	0,7445
<b>18</b>	0,0773	0,9664	0,6820	0,7599	0,7476	0,7474
<b>19</b>	0,0965	0,9702	0,6034	0,7713	0,7631	0,7609
<b>20</b>	0,0934	0,9655	0,5370	0,7788	0,7639	0,7655
<b>21</b>	0,0731	0,9740	0,6084	0,7769	0,7659	0,7655
<b>22</b>	0,0803	0,9693	0,6031	0,7732	0,7557	0,7583
<b>23</b>	0,0579	0,9792	0,6589	0,7656	0,7361	0,7430
<b>24</b>	0,0623	0,9745	0,6810	0,7769	0,7588	0,7619
<b>25</b>	0,0497	0,9825	0,7419	0,7675	0,7484	0,7515
<b>26</b>	0,0535	0,9802	0,7598	0,7750	0,7554	0,7591
<b>27</b>	0,0415	0,9868	0,8144	0,7618	0,7276	0,7354
<b>28</b>	0,0535	0,9802	0,7145	0,7750	0,7635	0,7633
<b>29</b>	0,0387	0,9872	0,7805	0,7845	0,7712	0,7722
<b>30</b>	0,0345	0,9896	0,8781	0,7788	0,7666	0,7668

## Fold 4

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	0,7081	0,5383	0,7007	0,5425	0,5000	0,3517
<b>2</b>	0,6531	0,6016	0,6606	0,5425	0,5000	0,3517
<b>3</b>	0,5960	0,6574	0,5798	0,6635	0,6714	0,6631
<b>4</b>	0,4955	0,7429	0,5722	0,6862	0,6758	0,6757
<b>5</b>	0,4307	0,8062	0,5371	0,7070	0,7102	0,7068
<b>6</b>	0,3601	0,8469	0,5283	0,7127	0,7054	0,7064
<b>7</b>	0,2974	0,8856	0,5470	0,7183	0,7151	0,7156
<b>8</b>	0,2563	0,8974	0,5469	0,7146	0,7042	0,7048
<b>9</b>	0,2131	0,9159	0,6012	0,7108	0,7020	0,7029
<b>10</b>	0,1690	0,9338	0,6100	0,7297	0,7243	0,7254
<b>11</b>	0,1625	0,9400	0,6656	0,7297	0,7275	0,7276



<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>12</b>	0,1265	0,9494	0,6245	0,7278	0,7252	0,7254
<b>13</b>	0,1223	0,9457	0,6558	0,7183	0,7174	0,7169
<b>14</b>	0,1029	0,9551	0,7257	0,7202	0,7166	0,7172
<b>15</b>	0,1070	0,9565	0,7138	0,7297	0,7269	0,7272
<b>16</b>	0,0942	0,9594	0,7646	0,7410	0,7455	0,7410
<b>17</b>	0,0866	0,9674	0,7351	0,7278	0,7281	0,7269
<b>18</b>	0,0761	0,9674	0,8736	0,7202	0,7140	0,7151
<b>19</b>	0,0773	0,9698	0,7559	0,7278	0,7303	0,7275
<b>20</b>	0,0822	0,9636	0,8343	0,7335	0,7375	0,7334
<b>21</b>	0,0713	0,9707	0,8226	0,7221	0,7164	0,7174
<b>22</b>	0,0645	0,9750	0,8729	0,7202	0,7166	0,7172
<b>23</b>	0,0523	0,9778	0,9718	0,7259	0,7279	0,7255
<b>24</b>	0,0654	0,9740	0,9564	0,7202	0,7149	0,7159
<b>25</b>	0,0555	0,9764	0,9528	0,7164	0,7137	0,7140
<b>26</b>	0,0530	0,9797	0,8812	0,7164	0,7137	0,7140
<b>27</b>	0,0484	0,9811	0,8971	0,7202	0,7179	0,7180
<b>28</b>	0,0468	0,9806	0,9829	0,7221	0,7248	0,7219
<b>29</b>	0,0453	0,9839	0,9491	0,7202	0,7175	0,7178
<b>30</b>	0,0461	0,9802	0,9529	0,7146	0,7113	0,7118

Fold 5

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>1</b>	0,6898	0,5595	0,6639	0,5747	0,5000	0,3649
<b>2</b>	0,6458	0,6129	0,6158	0,5917	0,5212	0,4145
<b>3</b>	0,5666	0,7179	0,5431	0,6957	0,6642	0,6633
<b>4</b>	0,4929	0,7632	0,5411	0,6786	0,6395	0,6316

<b>Epo ch</b>	<b>Train Loss</b>	<b>Train Acc</b>	<b>Test Loss</b>	<b>Test Acc</b>	<b>Test R</b>	<b>Test FPN</b>
<b>5</b>	0,4351	0,8152	0,4827	0,7599	0,7420	0,7464
<b>6</b>	0,3755	0,8644	0,4676	0,7656	0,7585	0,7594
<b>7</b>	0,3065	0,8823	0,4815	0,7524	0,7314	0,7359
<b>8</b>	0,2322	0,9064	0,5008	0,7618	0,7454	0,7496
<b>9</b>	0,2300	0,9102	0,4895	0,7713	0,7681	0,7669
<b>10</b>	0,1794	0,9234	0,5030	0,7826	0,7791	0,7783
<b>11</b>	0,1783	0,9409	0,5138	0,7524	0,7407	0,7432
<b>12</b>	0,1496	0,9362	0,5214	0,7713	0,7675	0,7667
<b>13</b>	0,1191	0,9490	0,5402	0,7732	0,7680	0,7680
<b>14</b>	0,1152	0,9579	0,5521	0,7675	0,7630	0,7626
<b>15</b>	0,1025	0,9603	0,5810	0,7618	0,7506	0,7532
<b>16</b>	0,0877	0,9627	0,6395	0,7656	0,7568	0,7584
<b>17</b>	0,0953	0,9627	0,5907	0,7675	0,7700	0,7652
<b>18</b>	0,0826	0,9655	0,6370	0,7599	0,7501	0,7521
<b>19</b>	0,1010	0,9664	0,6122	0,7561	0,7520	0,7513
<b>20</b>	0,0732	0,9721	0,6610	0,7599	0,7559	0,7551
<b>21</b>	0,0634	0,9750	0,6755	0,7637	0,7609	0,7595
<b>22</b>	0,0664	0,9754	0,7366	0,7524	0,7522	0,7491
<b>23</b>	0,0525	0,9806	0,7398	0,7580	0,7548	0,7536
<b>24</b>	0,0499	0,9811	0,7395	0,7656	0,7591	0,7597
<b>25</b>	0,0514	0,9797	0,8504	0,7543	0,7556	0,7516
<b>26</b>	0,0876	0,9764	0,7186	0,7694	0,7601	0,7620
<b>27</b>	0,0589	0,9792	0,7277	0,7580	0,7398	0,7442
<b>28</b>	0,0568	0,9806	0,7811	0,7429	0,7209	0,7250
<b>29</b>	0,0441	0,9849	0,7857	0,7599	0,7449	0,7487
<b>30</b>	0,0436	0,9816	0,7645	0,7694	0,7572	0,7603