

**LONG SHORT TERM MEMORY (LSTM) UNTUK IDENTIFIKASI  
CYBERBULLYING PADA KOMENTAR INSTAGRAM**

**TUGAS AKHIR**

**Tugas Akhir ini sebagai salah satu syarat untuk memperoleh gelar sarjana  
Teknik Informatika Universitas Pembangunan Nasional "Veteran" Yogyakarta**



Disusun Oleh:

**ARIF KURNIAWAN**

**123160068**

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

## ABSTRAK

Instagram merupakan salah satu media sosial populer. Seperti halnya media sosial lainnya, media sosial instagram merupakan media yang dapat digunakan untuk mencari informasi, berkomunikasi serta mengembangkan bisnis. Berkembangnya teknologi komunikasi serta informasi telah mengubah pola interaksi individu yang semula dilakukan secara konvensional menjadi modern. Kehadiran internet dan media sosial membuat masyarakat lebih mudah dalam berkomunikasi serta mendapatkan informasi di manapun dan kapanpun tanpa adanya batasan ruang, waktu dan jarak.

Terlepas sisi positif yang ditawarkan, kehadiran internet dan media sosial juga membawa dampak negatif, salah satunya berkembangnya fenomena *cyberbullying*. *Cyberbullying* dapat diartikan sebagai bentuk penggunaan teknologi untuk mengintimidasi, menjadikan korban, atau mengganggu individu atau sekelompok orang. Instagram merupakan salah satu media sosial dengan tingkat *cyberbullying* yang tinggi. Maka dari itu diperlukan sebuah sistem yang dapat mengidentifikasi komentar *cyberbullying* pada komentar instagram. *Sentimen analysis* dapat diterapkan untuk melakukan identifikasi *cyberbullying* pada komentar instagram. Algoritma yang digunakan untuk *sentimen analysis* adalah *Long Short Term Memory*. Algortima ini akan melakukan proses analisis komentar dengan mengubah komentar menjadi data berbentuk *sequential*. Pada proses *embedding* memanfaatkan metode *word2vec* untuk menemukan makna semantik antar kata. Proses pada penelitian ini terdiri dari pengolahan data awal, *text preprocessing*, pembentukan model LSTM dan identifikasi. Identifikasi terdiri dari empat kelas yaitu *cyberbullying*, *irrelevant*, *neutral* dan *non cyberbullying*.

Pengujian dilakukan dengan skenario pengujian *k-fold cross validation* untuk menentukan tingkat akurasi, presisi, *recall* dan *F1-Score* dari model LSTM yang dibangun. Konfigurasi pelatihan meliputi 100 *epoch* dengan menambahkan *L1L2 regularizer* serta nilai *dropout* sebesar 0.3. Pelatihan dilakukan dengan membagi data latih sebesar 90% dan data uji sebesar 10% dengan membagi data menjadi 10 *subset fold*. Performa dari pengujian *10-fold* mendapatkan performa rata-rata hasil akurasi sebesar 68.1%, presisi 70.08%, *recall* 68.1% dan *F1-Score* 68.8%. Skenario lain membandingkan performa algoritma LSTM dengan *random forest classifier*. Hasil perbandingan menunjukkan keunggulan algoritma LSTM dibandingkan *random forest classifier* pada nilai akurasi 70% berbanding 62%, nilai presisi 69% berbanding 61%, nilai *recall* 70% berbanding 62% dan nilai *F1-Score* 69% berbanding 60%.

**Kata kunci:** *Cyberbullying*, *Sentimen analysis*, *Long Short Term Memory*.

## DAFTAR ISI

HALAMAN JUDUL.....	i
HALAMAN PENGESAHAN PEMBIMBING .....	ii
HALAMAN PENGESAHAN PENGUJI.....	iii
SURAT PERNYATAAN KARYA ASLI TUGAS AKHIR.....	iv
PERNYATAAN BEBAS PLAGIAT.....	v
HALAMAN PERSEMBAHAN.....	vi
ABSTRAK .....	vii
KATA PENGANTAR.....	viii
DAFTAR ISI .....	x
DAFTAR GAMBAR.....	xii
DAFTAR TABEL .....	xiv
DAFTAR MODUL PROGRAM.....	.xv
BAB I PENDAHULUAN.....	1
1.1. Latar Belakang .....	1
1.2. Perumusan Masalah.....	3
1.3. Batasan Masalah.....	4
1.4. Tujuan Penelitian.....	4
1.5. Manfaat Penelitian.....	4
1.6. Metodologi Penelitian.....	5
1.7. Sistematika Penulisan .....	7
BAB II TINJAUAN PUSTAKA.....	9
2.1. <i>Cyberbullying</i> .....	9
2.1.1. Pengertian <i>Cyberbullying</i> .....	9
2.1.2. Jenis-Jenis <i>Cyberbullying</i> .....	10
2.1.3. Dampak <i>Cyberbullying</i> .....	11
2.2. <i>Web Scraping</i> .....	11
2.3. Analisis Sentimen.....	12
2.4. <i>Deep Learning</i> .....	13
2.5. <i>Long Short Term Memory (LSTM)</i> .....	13
2.6. <i>Natural Language Processing</i> .....	20
2.7. <i>Influencer</i> .....	21
2.8. Instagram.....	21
2.9. Identifikasi.....	22
2.10. <i>K-Fold Cross Validation</i> .....	22
2.11. <i>Confusion Matrix</i> .....	22
2.12. Penelitian Terdahulu.....	23
BAB III METODE PENELITIAN DAN PENGEMBANGAN SISTEM.....	26
3.1. Metodologi Penelitian.....	26
3.1.1. Metode Pengumpulan Data .....	26

3.1.2. Metodologi Pengembangan Sistem .....	46
3.2. Analisis kebutuhan Sistem.....	47
3.3. Analisis.....	47
3.3.1. Analisis Pembuatan Model LSTM .....	48
3.3.2. Analisis Pengujian Model LSTM .....	68
3.4. Perancangan Sistem .....	69
3.4.1. Perancangan Arsitektur Sistem Secara Keseluruhan .....	69
3.4.2. Struktur Menu .....	71
3.4.3. Perancangan Model .....	72
3.4.4. Perancangan <i>User Interface</i> .....	85
3.5. Rancangan Pengujian .....	92
BAB IV HASIL, PENGUJIAN DAN PEMBAHASAN.....	93
4.1. Hasil Kuesioner .....	93
4.2. Implementasi.....	107
4.2.1. Implementasi Pembuatan Model.....	107
4.2.2. Implementasi Pengujian. Model.....	118
4.2.3. Implementasi Sistem.....	122
4.3. Pengujian.....	140
4.3.1. Pengujian <i>Confusion Matrix</i> .....	141
4.3.2. Perbandingan Algoritma.....	142
4.3.3. Pembahasan.....	143
BAB V PENUTUP.....	145
5.1. Kesimpulan .....	145
5.2. Saran.....	145
DAFTAR PUSTAKA.....	146

## DAFTAR GAMBAR

<b>Gambar 2.1</b> Struktur Jaringan LSTM .....	14
<b>Gambar 2.2</b> Notasi Pada Jaringan LSTM .....	15
<b>Gambar 2.3</b> Forget gate .....	16
<b>Gambar 2.4</b> Input gate dan kandidat konteks .....	17
<b>Gambar 2.5</b> Proses memperbarui memori sel .....	19
<b>Gambar 2.6</b> Output gate dan Hidden State .....	20
<b>Gambar 3.1</b> Kuesioner Pertanyaan Nomor 1 dan 2 .....	27
<b>Gambar 3.2</b> Kuesioner Pertanyaan Nomor Tiga (3) .....	27
<b>Gambar 3.3</b> Kuesioner Pertanyaan Nomor empat (4) .....	28
<b>Gambar 3.4</b> Pertanyaan Kuesioner Nomor Lima (5) .....	28
<b>Gambar 3.5</b> Pendahuluan .....	29
<b>Gambar 3.6</b> Kuesioner Pertanyaan Nomor Enam (6) .....	29
<b>Gambar 3.7</b> Kuesioner Pertanyaan Nomor Tujuh (7) .....	29
<b>Gambar 3.8</b> Kuesioner Pertanyaan Nomor Delapan (8) .....	30
<b>Gambar 3.9</b> Kuesioner Pertanyaan Nomor Sembilan (9) .....	30
<b>Gambar 3.10</b> Kuesioner Pertanyaan Nomor Sepuluh (10) .....	31
<b>Gambar 3.11</b> Kuesioner Pertanyaan Nomor Sebelas (11) .....	31
<b>Gambar 3.12</b> Kuesioner Pertanyaan Nomor Dua Belas (12) .....	31
<b>Gambar 3.13</b> Kuesioner Pertanyaan Nomor Tiga Belas (13) .....	32
<b>Gambar 3.14</b> Kuesioner Pertanyaan Nomor Empat Belas (14) .....	32
<b>Gambar 3.15</b> Kuesioner Pertanyaan Nomor Lima Belas (15) .....	33
<b>Gambar 3.16</b> Kuesioner Pertanyaan Nomor 16 sampai 20 .....	33
<b>Gambar 3.17</b> Kuesioner Pertanyaan Nomor 21 sampai 25 .....	34
<b>Gambar 3.18</b> Hasil Kuesioner Pertanyaan Nomor Dua (2) .....	35
<b>Gambar 3.19</b> Hasil Pertanyaan Kuesioner Nomor Empat (4) .....	35
<b>Gambar 3.20</b> Hasil Kuesioner Pertanyaan Nomor Enam (6) .....	36
<b>Gambar 3.21</b> Hasil Kuesioner Pertanyaan Nomor Tujuh (7) .....	36
<b>Gambar 3.22</b> Hasil Kuesioner Pertanyaan Nomor Delapan (8) .....	37
<b>Gambar 3.23</b> Hasil Kuesioner Pertanyaan Nomor Sembilan (9) .....	37
<b>Gambar 3.24</b> Hasil Kuesioner Pertanyaan Nomor Sepuluh (10) .....	38
<b>Gambar 3.25</b> Hasil Kuesioner Pertanyaan Nomor Sebelas (11) .....	38
<b>Gambar 3.26</b> Hasil Kuesioner Pertanyaan Nomor Dua Belas (12) .....	39
<b>Gambar 3.27</b> Hasil Kuesioner Pertanyaan Nomor Tiga Belas (13) .....	39
<b>Gambar 3.28</b> Hasil Kuesioner Pertanyaan Nomor Empat Belas (14) .....	40
<b>Gambar 3.29</b> Hasil Kuesioner Pertanyaan Nomor Lima Belas (15) .....	40
<b>Gambar 3.30</b> Hasil Kuesioner Pertanyaan Nomor Enam Belas (16) .....	41
<b>Gambar 3.31</b> Hasil Kuesioner Pertanyaan Nomor Tujuh Belas (17) .....	41
<b>Gambar 3.32</b> Hasil Kuesioner Pertanyaan Nomor Delapan Belas (18) .....	42
<b>Gambar 3.33</b> Hasil Kuesioner Pertanyaan Nomor Sembilan Belas (19) .....	42
<b>Gambar 3.34</b> Hasil Kuesioner Pertanyaan Nomor Dua Puluh (20) .....	43
<b>Gambar 3.35</b> Hasil Kuesioner Pertanyaan Nomor Dua Puluh Satu (21) .....	43
<b>Gambar 3.36</b> Hasil Kuesioner Pertanyaan Nomor Dua Puluh Dua (22) .....	44
<b>Gambar 3.37</b> Hasil Kuesioner Pertanyaan Nomor Dua Puluh Tiga (23) .....	44
<b>Gambar 3.38</b> Hasil Kuesioner Pertanyaan Nomor Dua Puluh Empat (24) .....	45
<b>Gambar 3.39</b> Hasil Kuesioner Pertanyaan Nomor Dua Puluh Lima (25) .....	45
<b>Gambar 3.40</b> Analisis Pembuatan Model LSTM .....	48
<b>Gambar 3.41</b> Struktur tag HTML instagram .....	49

<b>Gambar 3.42</b> Flowchart Text Preprocessing.....	50
<b>Gambar 3.43</b> Flowchart Padding Sequence.....	54
<b>Gambar 3.44</b> Flowchart Word2vec.....	56
<b>Gambar 3.45</b> Ilustrasi Representasi Vector .....	58
<b>Gambar 3.46</b> Flowchart Representasi Sequence .....	59
<b>Gambar 3.47</b> Flowchart LSTM Layer .....	60
<b>Gambar 3.48</b> Flowchart Dense Layer .....	68
<b>Gambar 3.49</b> Arsitektur Sistem Secara Keseluruhan.....	71
<b>Gambar 3.50</b> Struktur Menu.....	72
<b>Gambar 3.51</b> Usecase diagram user.....	73
<b>Gambar 3.52</b> Usecase Diagram Admin .....	73
<b>Gambar 3.53</b> Activity Diagram Analisa Komentar .....	74
<b>Gambar 3.54</b> Activity Diagram User Analisa Postingan.....	76
<b>Gambar 3.55</b> Activity Diagram Admin Login.....	77
<b>Gambar 3.56</b> Activity Diagram Admin Home.....	78
<b>Gambar 3.57</b> Activity Diagram Admin Kamus.....	79
<b>Gambar 3.58</b> Activity Diagram Admin Kamus (Tambah Data).....	80
<b>Gambar 3.59</b> Activity Diagram Admin Data.....	81
<b>Gambar 3.60</b> Activity Diagram Admin Statistik .....	81
<b>Gambar 3.61</b> Activity Diagram Admin Logout.....	82
<b>Gambar 3.62</b> Class Diagram.....	83
<b>Gambar 3.63</b> Sequence Diagram User Analisa Komentar.....	84
<b>Gambar 3.64</b> Sequence Diagram User Analisa Postingan.....	85
<b>Gambar 3.65</b> Perancangan Halaman Utama User.....	86
<b>Gambar 3.66</b> Perancangan Halaman User Analisa Komentar .....	87
<b>Gambar 3.67</b> Perancangan Halaman User Analisa Komentar .....	87
<b>Gambar 3.68</b> Perancangan Halaman Admin Login.....	88
<b>Gambar 3.69</b> Perancangan Halaman Utama Admin .....	88
<b>Gambar 3.70</b> Perancangan Halaman Admin Kamus .....	89
<b>Gambar 3.71</b> Perancangan Halaman Admin Riwayat Uji Komentar .....	90
<b>Gambar 3.72</b> Perancangan Halaman Admin Data Training .....	90
<b>Gambar 3.73</b> Perancangan Halaman Admin Statistik .....	91
<b>Gambar 3.74</b> Perancangan Halaman Admin Model .....	91
<b>Gambar 4.1</b> Proses Training .....	117
<b>Gambar 4.2</b> Logger Training.....	117
<b>Gambar 4.3</b> Tampilan Halaman Login.....	123
<b>Gambar 4.4</b> Tampilan Halaman Home Admin .....	123
<b>Gambar 4.5</b> Tampilan Halaman Perbaikan Kata .....	125
<b>Gambar 4.6</b> Tampilan Halaman Data Training .....	127
<b>Gambar 4.7</b> Tampilan Halaman History Analysis Instagram Admin .....	128
<b>Gambar 4.8</b> Tampilan Halaman History Analysis Instagram Admin .....	128
<b>Gambar 4.9</b> Tampilan Halaman History Analysis Text Admin .....	129
<b>Gambar 4.10</b> Tampilan Halaman Statistic Admin .....	130
<b>Gambar 4.11</b> Tampilan Halaman Model Admin .....	132
<b>Gambar 4.12</b> Tampilan Halaman Model Admin .....	132
<b>Gambar 4.13</b> Tampilan Halaman Home User .....	133
<b>Gambar 4.14</b> Tampilan Halaman Analisa Komentar User Admin .....	136
<b>Gambar 4.15</b> Tampilan Halaman Analisa Instagram User .....	140
<b>Gambar 4.16</b> Perbandingan Performa LSTM dan Random Forest Classifier .....	143

## DAFTAR TABEL

<b>Tabel 2.1</b> Penelitian Sebelumnya.....	24
<b>Tabel 2.2</b> Lanjutan Penelitian Sebelumnya .....	25
<b>Tabel 3.1</b> Spesifikasi Kebutuhan Perangkat Keras Pembuatan Model.....	47
<b>Tabel 3.2</b> Spesifikasi Kebutuhan Perangkat Lunak Pembuatan Model.....	47
<b>Tabel 3.3</b> Cleansing .....	50
<b>Tabel 3.4</b> Spelling Correction.....	51
<b>Tabel 3.5</b> Stopword Removal.....	51
<b>Tabel 3.6</b> Stemming.....	51
<b>Tabel 3.7</b> Contoh Dataset.....	52
<b>Tabel 3.8</b> Frekuensi Kemunculan Kata.....	52
<b>Tabel 3.9</b> Text to Sequence.....	53
<b>Tabel 3.10</b> Padding Sequence.....	55
<b>Tabel 3.11</b> Hasil Representasi Vector Word2Vec.....	57
<b>Tabel 3.12</b> Hasil Representasi Sequence Vector .....	60
<b>Tabel 3.13</b> Rekap hasil perhitungan $Ct$ .....	66
<b>Tabel 3.14</b> Rekap hasil perhitungan $St$ .....	66
<b>Tabel 3.15</b> Rekap hasil perhitungan $Ct - 1$ .....	66
<b>Tabel 3.16</b> Rekap hasil perhitungan $St - 1$ .....	66
<b>Tabel 3.17</b> Text to Sequence.....	69
<b>Tabel 3.18</b> Contoh Hasil Prediksi Model.....	69
<b>Tabel 3.19</b> Tabel <i>Confusion Matrix</i> Model Klasifikasi Komentar .....	92
<b>Tabel 3.20</b> Tabel Pengujian <i>K-Fold Cross Validation</i> .....	92
<b>Tabel 3.21</b> Tabel Perbandingan Performa Algoritma.....	92
<b>Tabel 4.1</b> Frekuensi Jawaban Pertanyaan Nomor Tiga (3).....	94
<b>Tabel 4.2</b> Frekuensi Jawaban Pertanyaan Nomor Tiga (3).....	94
<b>Tabel 4.3</b> Lanjutan Frekuensi Jawaban Pertanyaan Nomor Empat (4) .....	95
<b>Tabel 4.4</b> Frekuensi Jawaban Pertanyaan Nomor Enam (6).....	95
<b>Tabel 4.5</b> Frekuensi Jawaban Pertanyaan Nomor Tujuh (7) .....	96
<b>Tabel 4.6</b> Frekuensi Jawaban Pertanyaan Nomor Delapan (8).....	96
<b>Tabel 4.7</b> Frekuensi Jawaban Pertanyaan Nomor Sembilan (9).....	97
<b>Tabel 4.8</b> Frekuensi Jawaban Pertanyaan Nomor Sepuluh (10).....	97
<b>Tabel 4.9</b> Frekuensi Jawaban Pertanyaan Nomor Sebelas (11).....	98
<b>Tabel 4.10</b> Frekuensi Jawaban Pertanyaan Nomor Dua Belas (12).....	98
<b>Tabel 4.11</b> Frekuensi Jawaban Pertanyaan Nomor Tiga Belas (13).....	99
<b>Tabel 4.12</b> Frekuensi Jawaban Pertanyaan Nomor Empat Belas (14).....	99
<b>Tabel 4.13</b> Frekuensi Jawaban Pertanyaan Nomor Lima Belas (15).....	100
<b>Tabel 4.14</b> Frekuensi Jawaban Pertanyaan Nomor Enam Belas (16).....	100
<b>Tabel 4.15</b> Frekuensi Jawaban Pertanyaan Nomor Tujuh Belas (17).....	101
<b>Tabel 4.16</b> Frekuensi Jawaban Pertanyaan Nomor Delapan Belas (18).....	101
<b>Tabel 4.17</b> Frekuensi Jawaban Pertanyaan Nomor Sembilan Belas (19).....	102
<b>Tabel 4.18</b> Frekuensi Jawaban Pertanyaan Nomor Dua Puluh (20).....	102
<b>Tabel 4.19</b> Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Satu (21).....	103
<b>Tabel 4.20</b> Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Dua (22).....	103
<b>Tabel 4.21</b> Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Tiga (23).....	104
<b>Tabel 4.22</b> Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Empat (24).....	104
<b>Tabel 4.23</b> Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Lima (25).....	105
<b>Tabel 4.24</b> Hasil Rekap Kategori Pengalaman.....	105

<b>Tabel 4.25</b> Hasil Rekap Kategori Media Sosial.....	106
<b>Tabel 4.26</b> Hasil Rekap Kategori Korban.....	106
<b>Tabel 4.27</b> <i>Hyperparameter Model</i> .....	115
<b>Tabel 4.28</b> <i>Confusion Matrix K-Fold Cross Validation</i> .....	141
<b>Tabel 4.29</b> Hasil Pengujian <i>K-Fold Cross validation</i> .....	141
<b>Tabel 4.30</b> <i>Confusion Matrix LSTM</i> .....	142
<b>Tabel 4.31</b> <i>Confusion Matrix Random Forest Classifier</i> .....	142
<b>Tabel 4.32</b> Hasil Perbandingan Algortima .....	142

## DAFTAR MODUL

<b>Modul Program 4.1</b> Source Code Koneksi Database.....	107
<b>Modul Program 4.2</b> Source Code Text Preprocessing.....	108
<b>Modul Program 4.3</b> Source Code Remove Newline.....	108
<b>Modul Program 4.4</b> Source Code Remove Username.....	108
<b>Modul Program 4.5</b> Source Code Remove URL.....	109
<b>Modul Program 4.6</b> Source Code Remove Non Alphabet.....	109
<b>Modul Program 4.7</b> Source Code Remove Whitespace.....	109
<b>Modul Program 4.8</b> Source Code Remove Repeating Char.....	109
<b>Modul Program 4.9</b> Source Code Normalisasi Kata.....	109
<b>Modul Program 4.10</b> Source Code Stopword Removal.....	110
<b>Modul Program 4.11</b> Source Code Stemming.....	110
<b>Modul Program 4.12</b> Source Code Tokenisasi.....	111
<b>Modul Program 4.13</b> Source Code Convert Data ke Sequence.....	111
<b>Modul Program 4.14</b> Source Code Pembagian Data Latih dan Data Uji.....	112
<b>Modul Program 4.15</b> Source Code Ekstraksi Corpus Word2Vec .....	112
<b>Modul Program 4.16</b> Source Code Training Word2Vec .....	113
<b>Modul Program 4.17</b> Source Code Update Pretrained Word2Vec.....	113
<b>Modul Program 4.18</b> Source Code Penyimpanan Pretrained Word2Vec .....	114
<b>Modul Program 4.19</b> Source Code Penyimpanan Nilai Embedding.....	114
<b>Modul Program 4.20</b> Source Code Penyimpanan Embedding.....	114
<b>Modul Program 4.21</b> Source Code Pembentukan Model.....	115
<b>Modul Program 4.22</b> Source Code Training Model .....	116
<b>Modul Program 4.23</b> Source Code K-Fold Cross Validation.....	118
<b>Modul Program 4.24</b> Lanjutan Source Code K-Fold Cross Validation .....	119
<b>Modul Program 4.25</b> Source Code Pengujian Random Forest Classifier.....	120
<b>Modul Program 4.26</b> Source Code Pengujian LSTM .....	120
<b>Modul Program 4.27</b> Lanjutan Source Code Pengujian LSTM .....	121
<b>Modul Program 4.28</b> Source Code Grafik Perbandingan Algoritma.....	122
<b>Modul Program 4.29</b> Source Code Halaman Perbaikan Kata .....	124
<b>Modul Program 4.30</b> Source Code Halaman Data Training .....	126
<b>Modul Program 4.31</b> Source Code Halaman History Analisis Instagram .....	127
<b>Modul Program 4.32</b> Lanjutan Source Code Halaman History Analisis Instagram .....	128
<b>Modul Program 4.33</b> Source Code Halaman History Analysis Text .....	129
<b>Modul Program 4.34</b> Source Code Halaman Statistik .....	130
<b>Modul Program 4.35</b> Source Code Halaman Model Admin.....	131
<b>Modul Program 4.36</b> Source Code Halaman Analisis Komentar User.....	133
<b>Modul Program 4.37</b> Lanjutan Source Code Halaman Analisis Komentar User .....	134
<b>Modul Program 4.38</b> Source Code Fungsi Prediksi Komentar .....	135
<b>Modul Program 4.39</b> Source Code Fungsi Cek Validitas .....	135
<b>Modul Program 4.40</b> Source Code Halaman Analisis Postingan User.....	136
<b>Modul Program 4.41</b> Lanjutan Source Code Halaman Analisis Postingan User .....	137
<b>Modul Program 4.42</b> Source Code Fungsi otomatisasi Login .....	137
<b>Modul Program 4.43</b> Source Code Scraping Instagram .....	137
<b>Modul Program 4.44</b> Lanjutan Source Code Scraping Instagram .....	138
<b>Modul Program 4.45</b> Source Code Fungsi Identification .....	138
<b>Modul Program 4.46</b> Source Code Cek Validitas Data .....	139
<b>Modul Program 4.47</b> Source Code Fungsi Prediction .....	139

## DAFTAR PERSAMAAN

<b>Persamaan 2.1</b> <i>Forget Gate</i> .....	16
<b>Persamaan 2.2</b> <i>Input Gate</i> .....	17
<b>Persamaan 2.3</b> <i>Kandidat Konteks</i> .....	17
<b>Persamaan 2.4</b> <i>Cell State</i> .....	19
<b>Persamaan 2.5</b> <i>Output State</i> .....	19
<b>Persamaan 2.6</b> <i>Output LSTM</i> .....	20
<b>Persamaan 2.7</b> Fungsi <i>Sigmoid</i> .....	20
<b>Persamaan 2.8</b> Fungsi <i>Tanh</i> .....	20
<b>Persamaan 2.9</b> Fungsi <i>Softmax</i> .....	20
<b>Persamaan 2.10</b> <i>Accuracy</i> .....	23
<b>Persamaan 2.11</b> <i>Precision</i> .....	23
<b>Persamaan 2.12</b> <i>Recall</i> .....	23
<b>Persamaan 2.13</b> <i>F1-Score</i> .....	23



# **BAB I**

## **PENDAHULUAN**

Bab ini menjelaskan tentang latar belakang, rumusan masalah, tujuan, manfaat, batasan masalah dan metodologi penelitian tugas akhir mengenai identifikasi komentar *cyberbullying* pada instagram menggunakan metode *Long Short Term Memory* (LSTM).

### **1.1. Latar Belakang**

Instagram merupakan salah satu media sosial yang populer beberapa tahun ini. Seperti halnya media sosial lainnya, media sosial ini juga merupakan media yang dapat digunakan untuk mencari informasi, berkomunikasi serta mengembangkan bisnis. Melalui media ini, seseorang dapat membagikan keluh-kesah, foto, video singkat serta aktivitas keseharian melalui akun instagramnya untuk disampaikan kepada khalayak umum. Selain hal tersebut, instagram juga ditunjang dengan beberapa fitur pendamping lainnya seperti *snapgram* dengan berbagai macam efek serta fitur *live* yang memungkinkan seseorang menyiarkan video secara langsung.

Berkembangnya teknologi komunikasi dan informasi telah mengubah pola interaksi individu yang semula dilakukan secara konvensional menjadi serba digital. Internet telah menciptakan sebuah dunia digital baru yang menciptakan ruang kultural, hal ini dapat dirasakan dengan hadirnya media sosial (Sakti et al., 2013). Kehadiran internet dan media sosial membuat masyarakat lebih mudah dalam berkomunikasi serta mendapatkan informasi di manapun dan kapanpun tanpa adanya batasan ruang, jarak maupun waktu.

Terlepas dari sisi positif yang ditawarkan, kehadiran internet dan media sosial juga membawa sisi negatif, salah satunya adalah berkembangnya fenomena *cyberbullying*. *Cyberbullying* sendiri merupakan suatu bentuk penggunaan teknologi untuk mengintimidasi

menjadikan korban, atau mengganggu individu atau sekelompok orang (Bhat, 2008). Fenomena *cyberbullying* sering terjadi dalam kehidupan sehari-hari dan akibat paling fatal dari bentuk kejahatan ini adalah bunuh diri. Akan tetapi *cyberbullying* yang terdapat di Indonesia masih menjadi hal yang sepele (Yana Choria Utami, 2014). Berdasarkan penelitian yang dilakukan oleh Asosiasi Pengguna Jasa Internet Indonesia (APJII) pada tahun 2019 sebanyak 49% pengguna media sosial di Indonesia pernah mengalami tindak *cyberbullying*. Permasalahan *cyberbullying* menjadi hal yang penting karena memiliki dampak yang tidak dapat dikesampingkan.

Secara mental, *cyberbullying* membuat korbannya merasa kesal, malu, bodoh dan bahkan marah. Secara Emosional, *cyberbullying* membuat korbannya merasa malu atau kehilangan minat pada hal yang disukai. Secara Fisik, *cyberbullying* dapat membuat korbannya merasa lelah (kurang tidur) atau mengalami gejala sakit perut dan sakit kepala. Dalam kasus yang paling ekstrem, *cyberbullying* dapat menyebabkan korban untuk mengakhiri nyawanya sendiri (Unicef, 2020).

Untuk mendukung penelitian ini, telah dilakukan *survey* pada tanggal 12 Februari 2020 sampai dengan 17 Februari 2020 yang memperoleh tanggapan dengan jumlah 132 tanggapan. Dari 132 tanggapan yang diterima, sebanyak 42% responden menyatakan sering menemukan tindakan *cyberbullying* di instagram, angka ini merupakan yang tertinggi dibandingkan dengan media sosial lainnya. Kemudian dari 132 tanggapan, 66% responden menyatakan bahwa tindakan *cyberbullying* yang sering dijumpai dalam media sosial berupa komentar jahat atau menyakiti orang lain, angka tersebut merupakan yang tertinggi dibandingkan jenis *cyberbullying* yang lainnya. Permasalahan *cyberbullying* dalam kolom komentar instagram menjadi hal yang penting untuk dikaji sebagai pemrosesan teks.

Pada penelitian sebelumnya mengenai kasus dan objek serupa yang dilakukan oleh Andriansyah (Andriansyah, 2017) proses analisis sentimen dengan metode *support vectore machine* hanya dibagi menjadi dua kelas yaitu positif *cyberbullying* dan negatif *cyberbullying*, selain itu data yang digunakan dalam penelitian ini jumlahnya masih sedikit. Kemudian penelitian yang dilakukan oleh Wanda athira luqyana (Luqyana et al., 2018) memiliki kelemahan dalam proses ekstraksi fitur dengan *lexcion based features* di mana proses pengenalan sentimen hanya berdasarkan kamus *lexcion* yang berisi kata positif dan kata negatif.

Dari permasalahan di atas, solusi yang dapat dikembangkan yaitu perangkat lunak dengan kemampuan melakukan identifikasi komentar secara semantik. Representasi *vector* kata menggunakan model *word2vec* dapat menemukan makna semantik antar kata dan berguna dalam berbagai pemrosesan bahasa alami (Nurrohmat & SN, 2019). Selain itu, teknik *deep learning* dapat digunakan dalam penelitian ini. *Deep learning* merupakan salah satu teknik *machine learning* yang memanfaatkan banyak lapisan pemrosesan informasi non linear untuk melakukan ekstraksi fitur, pengenalan pola dan klasifikasi. Kemudian algoritma yang akan digunakan adalah *Long Short Term Memory* (LSTM), dalam kasus analisis sentimen algoritma ini memiliki hasil yang lebih baik dibandingkan metode konvensional yang menandakan bahwa metode ini cocok digunakan dalam analisis sentimen (Nurrohmat & SN, 2019).

## 1.2. Perumusan Masalah

Berdasarkan latar belakang tersebut, maka perumusan masalah pada penelitian ini adalah sebagai berikut:

1. Bagaimana mengidentifikasi komentar instagram dengan menentukan makna komentar sehingga dapat menghasilkan informasi yang bermanfaat bagi pengguna Instagram dengan menggunakan *Long Short Term Memory* (LSTM) .
2. Bagaimana tingkat performa algoritma *Long Short Term Memory* dalam melakukan identifikasi *cyberbullying* pada komentar instagram.

### **1.3. Batasan Masalah**

Pada penelitian ini, permasalahan dibatasi sebagai berikut:

1. Data penelitian didapatkan dari kolom komentar *influencer* di Indonesia yaitu Samuel Alexander Pieter dan Karin Novilda yang dipilih berdasarkan hasil *survey* yang dilakukan pada tanggal 12 Februari 2020 sampai dengan 17 Februari 2020.
2. Data penelitian yang digunakan dari tahun 2018 sampai 2020.
3. Data penelitian yang digunakan berbahasa Indonesia.
4. Pada penelitian ini akan dilakukan identifikasi komentar termasuk *cyberbullying*, *Irrelevant*, *neutral* dan *non cyberbullying*.

### **1.4. Tujuan Penelitian**

Tujuan pada penelitian ini adalah mengukur tingkat akurasi metode *Long Short Term Memory* (LSTM) dalam mengidentifikasi *cyberbullying* pada komentar di media sosial instagram.

### **1.5. Manfaat Penelitian**

Manfaat penelitian ini adalah memudahkan pengguna instagram untuk mengenali komentar bernada *cyberbullying* serta memudahkan proses identifikasi dari banyaknya

komentar yang ada pada sebuah kiriman instagram sebagai bentuk upaya preventif maupun upaya represif dari fenomena *cyberbullying* yang ada di instagram.

### **1.6. Metodologi Penelitian**

Metode yang digunakan dalam penelitian ini dibagi menjadi dua yaitu metode pengumpulan data dan metode pengembangan *system*. Adapun penjelasan dari kedua metode tersebut sebagai berikut:

#### **1. Metodologi Pengumpulan Data**

Pengumpulan data yang digunakan dalam penelitian ini yaitu dengan observasi, kuesioner dan studi pustaka. Observasi yang dilakukan yaitu dengan menganalisis tingkah laku seseorang atau sekelompok dalam berkomentar di media sosial instagram. Kuesioner dibutuhkan untuk memperoleh data yang sesuai dengan tujuan penelitian dan penjabaran dari hipotesis. Penyebaran kuesioner dilakukan secara *online* melalui *whatsapp* dan *line* dengan bantuan *google form* dalam pembuatan formulir kuesioner dan hasil kuesioner tersebut kemudian diolah dengan *IBM SPSS stastistic 25*. Kuesioner tersebut dibutuhkan untuk mendukung pengambilan data yang akan diolah menjadi informasi. Kemudian Studi pustaka dilakukan dengan cara mencari dan menghimpun data ataupun sumber-sumber pustaka yang dapat mendukung penelitian serta memberikan informasi untuk menyelesaikan permasalahan pada penelitian ini. Studi kepustakaan yang digunakan bersumber dari buku, jurnal, artikel dan *paper* yang berkaitan pada penelitian ini.

#### **2. Metodologi Pengembangan Sistem**

Dalam tahapan ini metode yang akan digunakan adalah metode GRAPPLE (*Guidelines for Rapid APPLication Engineering*). Metode GRAPPLE merupakan metode

pengembangan perangkat lunak yang fleksibel dan memberikan panduan yang jelas dalam pengembangan perangkat lunak. Metode ini terdiri dari lima bagian:

a) *Requirement Gathering*

Tahap pertama yang dilakukan oleh pengembang perangkat lunak adalah mengambil informasi lengkap dari pengguna tentang *system* yang akan dibangun dengan kuesioner. Kuesioner dilakukan secara *online* dengan responden yang berhubungan dengan *system*. Analisis masalah, fungsi dan kebutuhan *system* termasuk dalam tahap *Requirement gathering*.

b) *Analysis*

Di tahap analisis yang dilakukan adalah menggali lebih dalam hasil dari tahap sebelumnya. Tahap ini mengkaji permasalahan pengguna dan melakukan analisis solusinya. Yang termasuk dalam tahapan ini adalah pengembangan data dan informasi dari *requirement gathering* serta pembuatan diagram.

c) *Design*

Tahap *design* dilakukan untuk merancang solusi yang dihasilkan pada tahap analisis dan *design* dapat berjalan dua arah saling menyesuaikan hingga memperoleh rancangan yang sesuai. Yang termasuk dalam tahapan ini adalah implementasi model dan diagram yang telah dianalisis dan dibuat rancangannya.

d) *Development*

Tahap ini ditangani oleh pengembang program untuk membangun kode program dan *user interface*. Pengujian program dan dokumentasi sistem dilakukan pada tahap ini.

e) *Deployment*

Tahap *deployment* adalah tahap pendistribusian produk kepada pengguna. Tahap ini mencakup instalasi dan perencanaan *backup* data bila diminta oleh pengguna sesuai dengan perjanjian sebelumnya.

### **1.7. Sistematika Penulisan**

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

#### **1. BAB I Pendahuluan**

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

#### **2. BAB II Tinjauan Pustaka**

Tinjauan pustaka memuat tentang dasar teori yang digunakan untuk analisis dan perancangan sistem serta implementasi pada penelitian ini. Selain itu juga sebagai bahan referensi dan fondasi untuk memperkuat argumentasi dalam penelitian ini. Teori-teori yang sesuai dengan penelitian ini antara lain identifikasi, *cyberbullying*, *influencer*, Instagram, *word2vec* dan *long short term memory*.

#### **3. BAB III Metodologi Penelitian dan Pengembangan Sistem**

Pada bagian ini akan membahas mengenai langkah-langkah yang dilakukan dalam pelaksanaan penelitian dan pengembangan sistem hingga implementasi sistem dalam mengidentifikasi *cyberbullying* di kolom komentar instagram.

#### **4. BAB IV Hasil, Pengujian dan Pembahasan**

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

#### **5. BAB V Penutup**

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

## **BAB II**

### **TINJAUAN PUSTAKA**

#### **2.1. *Cyberbullying***

*Cyberbullying* merupakan perluasan dari *bullying*, yang mana *bullying* diartikan sebagai kekerasan fisik maupun mental yang dilakukan seseorang atau kelompok orang terhadap kelompok atau orang lainnya sehingga korban merasa teraniaya. Penjelasan mengenai *cyberbullying* dijelaskan sebagai berikut:

##### **2.1.1. Pengertian *Cyberbullying***

Pada tahun 2010, istilah *cyberbullying* ditambahkan ke dalam kamus *Oxford English Dictionary* (OED). Istilah tersebut merujuk pada penggunaan teknologi informasi untuk menggertak orang dengan mengirim atau memposting teks bersifat mengintimidasi atau mengancam. OED menunjukkan penggunaan pertama dari istilah ini pertama kali di Canberra pada tahun 1998, tetapi istilah ini sudah ada sebelumnya di Artikel *New York Times* 1995 di mana banyak sarjana dan penulis Besley seorang kanada yang meluncurkan situs *cyberbullying* tahun 2013 dengan istilah *coining* (Rifauddin, 2016).

Menurut Bhat (Bhat, 2008) *cyberbullying* di definisikan sebagai bentuk penggunaan teknologi untuk mengintimidasi, menjadikan korban atau mengganggu seseorang individu atau sekelompok orang. Selain itu, menurut Smith (Smith dkk, 2008) *cyberbullying* dapat diartikan sebagai perilaku kasar yang dilakukan oleh seseorang atau sekelompok orang dengan menggunakan bantuan alat elektronik yang dilakukan secara berulang-ulang dan terus menerus pada seorang target yang kesulitan membela diri.

*Cyberbullying* merupakan bentuk lain dari *bullying*. Jika *bullying* dilakukan dengan cara melakukan perundungan secara *face to face* dan terjadi di dunia nyata, maka beda halnya dengan *cyberbullying* yang membutuhkan teknologi sebagai media perantara dan

dilakukan di dunia maya. Tindakan *cyberbullying* dapat terjadi di mana saja dan kapan saja dalam berbagai macam ranah kehidupan seperti sosial, politik, budaya, olahraga, pendidikan dan keluarga. Meskipun memiliki pola yang berbeda, tindakan *cyberbullying* dan *bullying* dianggap sebagai sebuah hal yang memiliki dampak negatif, terutama bagi kesehatan mental korban (*victim*) karena dapat menimbulkan depresi.

### **2.1.2. Jenis-Jenis *Cyberbullying***

Menurut Willard (Utami & Baiti, 2018), menyebutkan macam-macam jenis *cyberbullying* sebagai berikut:

- a) *Flaming* (Terbakar), yaitu mengirimkan pesan teks yang isinya merupakan kata-kata penuh amarah dan bersifat frontal. Istilah “*Flame*” merujuk pada kata-kata dipesan yang bersifat berapi-api dan dapat diartikan sebagai penghinaan atau komentar kasar terhadap orang lain. Secara lebih luas, *flaming* dapat diartikan sebagai tindakan yang dilakukan secara *online* dengan mengirim hinaan, sering dicampur dengan kata-kata kotor atau bahasa ofensif di situs jejaring sosial.
- b) *Harrasment* (gangguan), yaitu pesan-pesan yang berisi gangguan pada *e-mail*, sms, maupun pesan teks di jejaring sosial yang dilakukan secara terus menerus. Tindakan *harrasement* sering dilakukan dengan menulis pesan/komentar kepada korban dengan maksud untuk membuat korban merasa tidak nyaman dan mengalami kegelisahan.
- c) *Denigration* (Pencemaran nama baik), yaitu proses mengumbar aib atau keburukan seseorang di internet dengan maksud untuk merusak reputasi dan nama baik orang tersebut.
- d) *Impersonation* (Peniruan) yaitu berpura-pura menjadi orang lain dan mengirimkan pesan-pesan atau status yang tidak baik.
- e) *Outing* yaitu menyebarkan rahasia orang lain, atau foto-foto pribadi orang lain.

- f) *Trickery* (tipu daya) yaitu membujuk seseorang dengan tipu daya agar mendapatkan rahasia atau foto pribadi orang tersebut.
- g) *Exclusion* (Pengeluaran) yaitu secara sengaja dan kejam mengeluarkan seseorang dari *group online*.
- h) *Cyberstalking* yaitu mengganggu dan mencemarkan nama baik seseorang secara intens sehingga membuat ketakutan besar terhadap orang tersebut.

Dari berbagai penjelasan *cyberbullying* tersebut, *cyberbullying* menitik beratkan kepada tindakan verbal atau dilakukan secara tidak langsung yang dapat berdampak kepada kondisi emosional atau psikis dari korbannya.

### **2.1.3. Dampak *Cyberbullying***

Ketika seseorang menjadi korban dari tindakan *cyberbullying*, korban akan merasa bahwa dirinya diserang di mana saja termasuk tempat tinggalnya sehingga korban merasa tidak ada jalan keluar dari tindakan tersebut. Dampak dari *cyberbullying* dapat dikategorikan menjadi tiga (Unicef, 2019), yaitu:

1. Secara mental : Merasa kesal, malu, bodoh dan bahkan marah.
2. Secara Emosional : Merasa malu atau kehilangan terhadap minat-minat yang disukai.
3. Secara Fisik : Lelah yang disebabkan karena kurang tidur, atau mengalami gejala mual-mual dan gejala sakit kepala

Dalam berbagai kasus, dampak paling ekstrem dari tindakan *cyberbullying* adalah korban mengalami depresi yang disebabkan perasaan ditertawakan secara terus menerus yang menyebabkan korban memilih untuk mengakhiri nyawanya sendiri.

### **2.2. *Web Scraping***

*Web Scraping* adalah proses pengambilan sebuah dokumen semi terstruktur dari internet, umumnya berupa laman web yang dibangun dengan bahasa *markup* seperti HTML

atau XHTML yang bertujuan untuk mengambil informasi dari halaman tersebut baik secara keseluruhan atau sebagian untuk digunakan bagi kepentingan lain (Priyanto & Ma'arif, 2018). Aplikasi *web scraping* berfokus kepada cara pengambilan data dan ekstraksi data dengan ukuran yang beraneka ragam. Ada empat tahapan dalam *web scraping* yang dijelaskan sebagai berikut:

1. *Create Scraping Template*

Pada proses ini dilakukan proses observasi terhadap dokumen HTML dari *website* yang akan diambil informasinya untuk *tag* HTML yang mengapit informasi yang akan diambil.

2. *Explore Site Navigation*

Pada proses ini dilakukan proses observasi terhadap Teknik navigasi pada *website* yang akan diambil informasinya yang akan ditiru pada aplikasi *web scraper* yang akan dibuat.

3. *Automate navigation and extraction*

Berdasarkan pada proses 1 dan 2, aplikasi *web scraper* akan mengotomatisasi pengambilan informasi dari *website* yang telah ditentukan sebelumnya.

4. *Extracted data and package history*

Informasi yang diperoleh pada tahap tiga akan disimpan dalam *database*.

### **2.3. Analisis Sentimen**

Analisis sentimen atau *opinion mining* merupakan studi komputasi pendapat, sikap dan emosi orang-orang terhadap sebuah entitas. Entitas dapat mewakili individu, peristiwa, atau sebuah topik. Topik-topik ini kemungkinan besar akan dibahas dalam sebuah opini (Medhat et al., 2014). Penggunaan analisis sentimen saat ini telah menjadi alternatif yang lebih baik dibandingkan dengan melakukan *survey* secara konvensional yang berbiaya mahal. Umumnya analisis sentimen digunakan dalam menilai opini masyarakat terhadap

sebuah *brand*, tokoh dan objek tertentu yang digunakan sebagai dasar untuk menentukan keputusan di masa mendatang. Sentimen analisis dalam penelitian ini dimaksudkan untuk menilai pendapat yang diterima oleh seseorang apakah termasuk komentar *cyberbullying*, *Irrelevant*, netral maupun *non cyberbullying*.

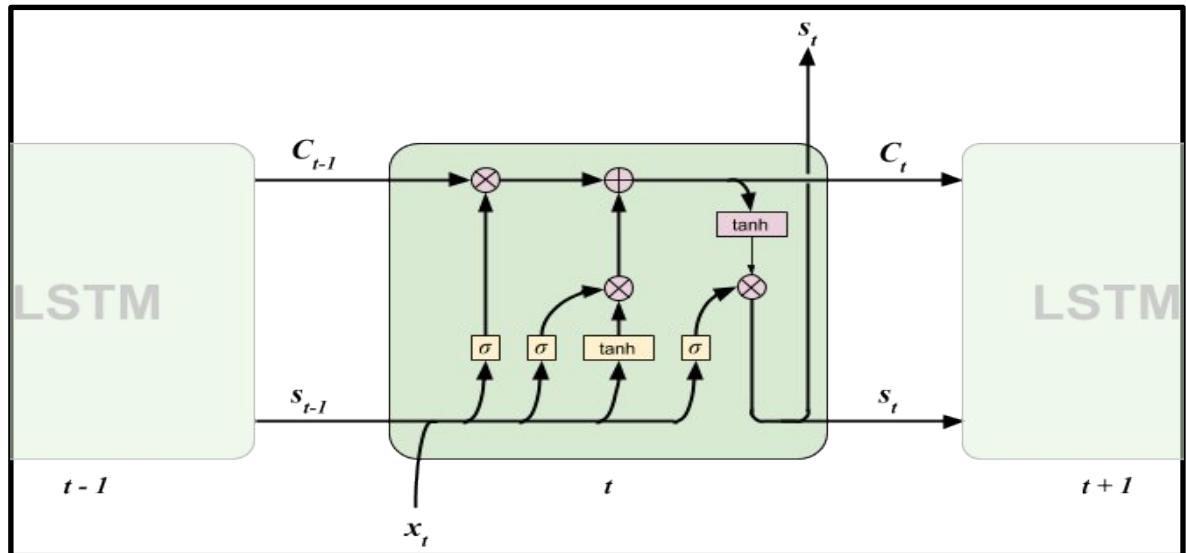
#### **2.4. Deep Learning**

*Deep learning* merupakan salah satu cabang *machine learning* yang memanfaatkan jaringan saraf tiruan untuk implementasi permasalahan dengan *dataset* yang besar. *Deep learning* adalah tentang belajar beberapa tingkat representasi dan abstraksi yang membantu untuk memahami data seperti gambar, suara dan teks (Zulfa & Winarko, 2017). Teknik *deep learning* memiliki arsitektur yang sangat kuat dalam *supervised learning*. Dengan jumlah lapisan yang lebih banyak model ini mampu mewakili data berlabel dengan baik.

#### **2.5. Long Short Term Memory (LSTM)**

*Long Short Term Memory* (LSTM) adalah salah satu algoritma *deep learning* yang dapat digunakan dalam pemrosesan data *sequential*. *Long Short Term Memory* (LSTM) merupakan jenis modul pemrosesan lain untuk *Recurrent Neural Network* (RNN). Seperti RNN, Jaringan LSTM juga terdiri dari modul-modul pemrosesan berulang, yang membedakan adalah modul pembentuk jaringan LSTM adalah modul LSTM sebelumnya. Perbedaan lainnya adalah adanya tambahan sinyal yang diberikan dari satu langkah waktu ke langkah waktu berikutnya, yaitu konteks. LSTM di desain untuk mengatasi *vanishing gradients* menggunakan mekanisme gerbang (*gate*) yang sebelumnya menjadi masalah pada RNN. Keunggulan metode LSTM dibandingkan dengan RNN adalah adanya arsitektur mengingat dan melupakan, serta *output* yang dihasilkan akan di proses kembali menjadi *input*. Selain itu, keunggulan lain yang dimiliki oleh metode ini adalah adanya kemampuan untuk mempertahankan *error* saat melakukan *backpropagation* yang memungkinkan kesalahan yang terjadi tidak akan mengalami peningkatan (Zhang 2016).

Seperti halnya dalam RNN, LSTM juga memiliki struktur jaringan berantai namun dengan struktur yang berbeda-beda. Jika dalam RNN hanya menggunakan satu lapisan tunggal, maka dalam LSTM digunakan empat lapisan yang saling berkaitan seperti pada gambar 2.1.



Gambar 2.1 Struktur Jaringan LSTM

Pada gambar 2.1, terdapat beberapa notasi yang digunakan dalam membangun jaringan LSTM. Notasi tersebut dijelaskan pada gambar 2.2.



Gambar 2.2 Notasi Pada Jaringan LSTM

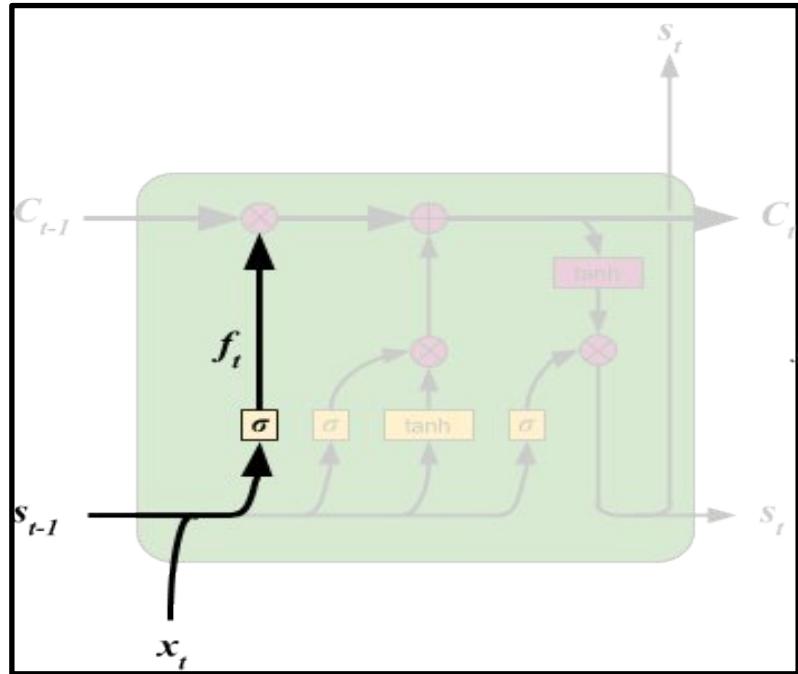
Jaringan LSTM terdiri dari blok memori yang disebut sel. Ada dua jalur yang digunakan untuk mentransfer informasi ke sel berikutnya, yaitu melalui *cell state* dan *hidden state*. *Cell state* merupakan rantai aliran data utama, di mana data yang mengalir tidak mengalami perubahan, hanya saja memungkinkan terjadinya beberapa transformasi linear. Data yang dibawa oleh *cell state* dapat ditambahkan maupun dihapus melalui gerbang

sigmoid. Gerbang ini memiliki kemiripan dengan serangkaian operasi matriks, yang berisi bobot individu yang berbeda. LSTM dirancang untuk menghindari masalah ketergantungan jangka panjang (*long-term*) karena menggunakan gerbang untuk proses mengontrol data yang akan dihafalkan (Le et al., 2019).

Terdapat tiga jenis gerbang yang dimiliki LSTM yaitu *forget gate*, *input gate* dan *output gate*. *Forget gate* merupakan gerbang yang dapat memilih informasi mana yang akan dihapus atau dilupakan pada sel. *Input gate* merupakan gerbang yang dapat menentukan nilai dari *input* yang akan diperbarui pada *state memory*. *Output gate* adalah gerbang yang menentukan hasil *output* yang sesuai dengan *input* dan memori pada sel (Ibnu, 2018).

Kunci utama dalam LSTM adalah *cell state*, yaitu garis horizontal yang berjalan di atas diagram. Dengan beberapa interaksi linear kecil, maka informasi yang ada dengan mudah akan mengalir tanpa mengalami perubahan. LSTM memiliki kemampuan untuk menghapus dan menambahkan informasi ke dalam sel yang dikontrol penuh oleh gerbang.

Langkah pertama dalam membangun jaringan LSTM adalah mengidentifikasi informasi yang tidak perlu dan akan dihapus dari sel. Proses identifikasi dan pengecualian akan ditentukan oleh fungsi sigmoid, yang mengambil dari unit LSTM terakhir yang ditunjukkan oleh ( $S_{t-1}$ ) pada waktu  $t-1$  dan arus *input* yang ditunjukkan oleh  $X_t$  pada waktu  $t$ . Gerbang ini disebut sebagai *forget gate* di mana  $f_t$  adalah *vector* dengan nilai antara 0 hingga 1, sesuai dengan keadaan sel yang ditunjukkan oleh ( $c_{t-1}$ ). *Forget gate layer* diuraikan dalam gambar 2.3.



**Gambar 2.3** *Forget gate*

Dari gambar 2.3, persamaan dari *forget gate layer* dapat diuraikan dalam persamaan 2.1.

Keterangan:

$f_t$  = forget gate

$\sigma$  = fungsi sigmoid

$W_f$  = beban pada *forget gate*

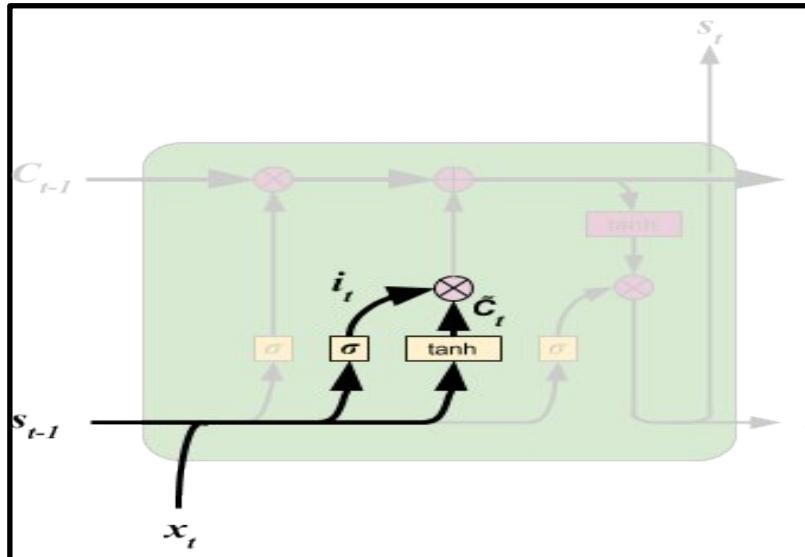
$U_f$  = beban recurrent pada forget gate

$S_{t-1}$  = output blok

$x_t$  = input saat-t

Langkah selanjutnya adalah memutuskan dan menyimpan informasi dari *input* baru yang dilambangkan dengan  $X_t$  dalam sel untuk memperbarui *cell state*. Pada langkah ini ada dua bagian yang dilalui, pertama adalah lapisan sigmoid dan kedua adalah tanh serta untuk memperbarui *cell state*. Lapisan sigmoid akan memutuskan apakah informasi akan diabaikan atau diperbarui dengan mengeluarkan *output* vector antara 0 hingga 1. Kemudian lapisan

tanh akan memberikan bobot pada nilai yang dilewati berdasarkan level kepentingannya dengan memberikan *vector* antara -1 sampai 1. Langkah kedua diuraikan seperti pada gambar 2.4.



**Gambar 2.4** *Input gate* dan kandidat konteks

Terdapat dua operasi dalam gambar 2.4, operasi yang pertama yaitu perhitungan *input gate* dan penentuan kandidat konteks yang ditunjukkan pada persamaan 2.2 dan 2.3.

## Keterangan:

$i_t$  = input gate

$\tilde{C}_t$  = Kandidat konteks

$\sigma$  = fungsi sigmoid

$W_i$  = beban pada *input gate*

$U_i$  = beban recurrent pada forget gate

$b_i$  = bias pada *forget gate*

$W_c$  = beban pada kandidat konteks

$U_c$  = beban *recurrent* pada kandidat konteks

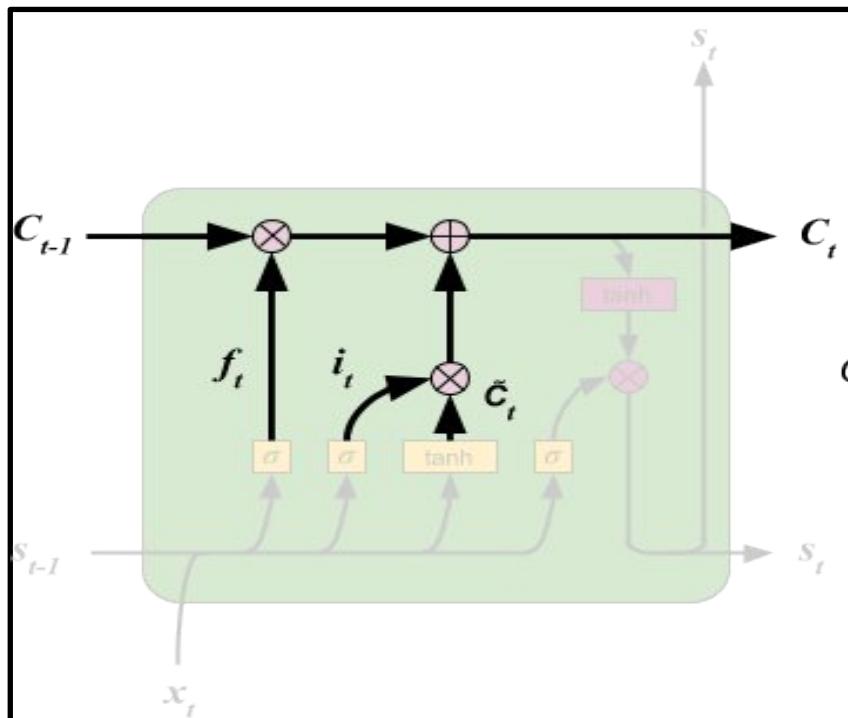
$b_c$  = bias pada kandidat konteks

$S_{t-1}$  = output blok LSTM sebelumnya

$x_t$  = *input* saat-t

Berdasarkan pada gambar 2.4, setelah nilai *input gate* dan kandidat konteks ditemukan, terjadi mekanisme perkalian yang ditunjukkan dengan notasi operasi perkalian elemen.

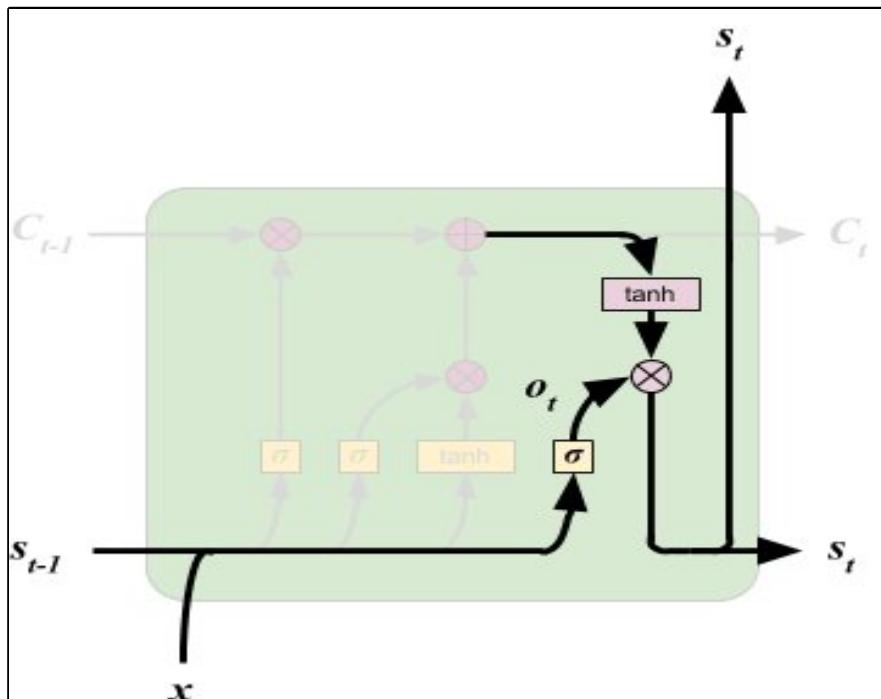
Langkah ketiga adalah memperbaharui nilai *cell state*. Kedua nilai yang telah dihasilkan pada langkah kedua kemudian akan dikalikan untuk memperbarui *cell state*. Memori baru ini kemudian akan ditambahkan ke memori lama yang dilambangkan dengan  $C_{t-1}$  dan akan menghasilkan memori yang baru yang dilambangkan  $C_t$ . Proses tersebut diuraikan dalam gambar 2.5.



**Gambar 2.5** Proses memperbarui memori sel

Berdasarkan gambar 2.5, persamaan yang terjadi dalam proses memperbarui memori sel dapat diuraikan dalam persamaan 2.4.

Pada langkah terakhir, nilai *output* yang dilambangkan  $S_t$  bersumber dari nilai pada sel *output* yang dilambangkan dengan  $O_t$  yang telah mengalami proses *filtering*. Pertama lapisan sigmoid memutuskan bagian sel mana yang menjadi *output*. Selanjutnya, *output* dari gerbang sigmoid ( $O_t$ ) dikalikan dengan nilai baru yang dihasilkan oleh lapisan tanh dari *cell state* ( $C_t$ ), nilai yang dihasilkan berkisar antara -1 sampai 1. Proses ini dijelaskan pada gambar 2.6.



**Gambar 2.6** *Output gate dan Hidden State*

Berdasarkan gambar 2.6, maka proses yang terjadi dapat dijelaskan dalam persamaan 2.5 dan 2.6.

## Keterangan :

$O_t$  = output gate

$\sigma$  = fungsi sigmoid

$W_o$  = beban untuk *output gate*

$U_o$  = beban recurrent pada output gate

$S_{t-1}$  = output blok LSTM sebelumnya

$x_t$  = input saat-t

$b_o$  = bias pada *output gate*

Ada beberapa fungsi aktivasi yang digunakan pada pembentukan jaringan LSTM yaitu *sigmoid*, *tanh* serta *softmax* diuraikan pada persamaan 2.7, 2.8 dan 2.9.

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \quad \dots \dots \dots \quad (2.8)$$

$$softmax(x) = \frac{e^{x_i}}{\sum_j e^{x_j}} \dots \quad (2.9)$$

## Keterangan:

$\sigma$  = fungsi sigmoid

$x$  = data *input*

$e$  = fungsi eksponensial

## **2.6. Natural Language Processing**

*Natural Language Processing* (Pemrograman Bahasa alami) merupakan pembuatan program yang memiliki kemampuan untuk memahami Bahasa manusia. Pada prinsipnya bahasa alami merupakan bahasa yang merepresentasikan pesan yang ingin disampaikan antar manusia (Wangsanegara & Subaeki, 2015).

NLTK merupakan pustaka untuk pemrosesan bahasa alami dalam bahasa pemrograman python. Perkembangannya didorong dengan kebutuhan dalam proses *text pre-processing* seperti *case folding*, *stopword removal*, *tokenizing* dan *filtering*. NLTK terdiri banyak paket pustaka yang dapat digunakan dalam menunjang pemrosesan bahasa alami.

## 2.7. *Influencer*

*Influencer* dapat diartikan sebagai seseorang yang memiliki jumlah pengikut sosial media dengan jumlah yang besar. Dalam hal ini, *influencer* dapat disematkan kepada seorang *youtuber*, *selebgram*, *blogger*, artis maupun profesi sejenisnya yang dipilih berdasarkan popularitas, keahlian ataupun reputasinya.

## 2.8. *Instagram*

Instagram merupakan salah satu media sosial paling populer saat ini. Media sosial ini pertama kali didirikan pada tahun 2010 oleh perusahaan *startup* Burbn, Inc. Pada awalnya, media sosial ini memiliki fokus yang beragam bagi sebuah program ponsel genggam. Hingga pada akhirnya Kevin Systrom dan Mike Rieger yang merupakan CEO dari perusahaan tersebut sepakat untuk menyederhanakan fokus aplikasi tersebut kepada aplikasi berbagi foto yang memungkinkan orang lain untuk menyukai serta memberikan komentar. Pada tahun 2012 aplikasi ini secara resmi telah berpindah kepemilikan kepada Facebook dengan valuasi mencapai angka satu miliar dollar Amerika.

Dengan Instagram, seorang pengguna memiliki akses untuk membagikan foto, video serta aktivitas lainnya ke berbagai situs jejaring sosial lainnya, termasuk pada Instagram sendiri. Salah satu keunggulan Instagram dibanding dengan media sosial lainnya adalah keandalannya dalam melakukan penyuntingan gambar, banyak efek yang disediakan sehingga foto yang akan dibagikan dapat lebih menarik untuk ditampilkan. Dengan canggihnya peralatan pintar saat ini, tentunya menghasilkan foto yang bagus tidak mengharuskan memerlukan peralatan fotografi kelas atas. Hal ini juga menjadi salah satu penyebab dari pesatnya perkembangan Instagram saat ini.

Saat ini, pengguna aktif media sosial Instagram telah mencapai lebih dari satu miliar pengguna. Pada umumnya alasan seorang pengguna menggunakan Instagram adalah karena ingin mendokumentasikan kehidupannya, mengetahui aktivitas orang lain, ingin tampil

keren, tertarik dengan kreativitas orang lain ataupun hanya sekedar untuk mengusir rasa bosan (Arifuddin, 2018.)

### **2.9. Identifikasi**

Menurut JP. Chaplin yang diterjemahkan oleh Kartini Kartono mengemukakan bahwa Identifikasi merupakan proses pengenalan, penempatan objek atau individu ke dalam kelas berdasarkan karakteristik tertentu . Menurut Poerwardaminto, identifikasi merupakan penempatan atau penentuan identitas seseorang atau benda (Hermiyanty, Wandira Ayu Bertin, 2017).

Berdasarkan pendapat para ahli tersebut, identifikasi dapat diartikan sebagai penempatan atau penentuan suatu individu atau benda berdasarkan karakteristik tertentu. Sedangkan identifikasi dalam penelitian ini merupakan proses penempatan atau penentuan objek (komentar) ke dalam kelas yang sesuai berdasarkan karakteristik tertentu.

### **2.10. *K-Fold Cross Validation***

*K-fold Cross validation* merupakan metode *cross validation* yang populer dengan melipat data sejumlah K dan mengulangi iterasi *training* sebanyak K. Metode *validasi silang* bekerja dengan membagi *subset* data *training* dan data uji sejumlah K sehingga seluruh data dalam *dataset* digunakan sebagai pengujian. Tujuan penggunaan *validasi silang* adalah membatasi model dari *overfitting* dan memberikan wawasan bagaimana model akan menyelesaikan masalah di luar *dataset*.

### **2.11. *Confusion Matrix***

*Confusion Matrix* merupakan metode untuk mengukur akurasi, presisi, *recall* dan *F1-Score* dengan membandingkan hasil klasifikasi yang dilakukan oleh sistem dengan klasifikasi sebenarnya. Pada penelitian ini *confusion matrix* digunakan untuk mengukur akurasi, presisi, *recall* dan *F1-Score* yang disajikan dalam persamaan 2.10, 2.11, 2.12 dan 2.13.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad \dots \dots \dots \quad (2.10)$$

$$F1-Score = \frac{2(Recall \times Precision)}{Recall + Precision} \quad \dots \dots \dots \quad (2.13)$$

## Keterangan:

**TP (True Positive)** = Jumlah Kelas *True* yang di prediksi benar pada kelas *True*.

**TN (True Negative)** = Jumlah Kelas *False* yang di prediksi benar pada kelas *False*.

**FP (False Positive)** = Jumlah Kelas *False* yang di prediksi benar pada kelas *False*.

$FN$  (*False Negative*) = Jumlah Kelas *False* yang di prediksi benar pada kelas *True*.

## **2.12. Penelitian Terdahulu**

Penelitian yang berkaitan dengan *cyberbullying* telah banyak dilakukan, seperti penelitian yang dilakukan oleh Andriansyah (Andriansyah, 2017). Dengan data yang diperoleh dari kolom komentar *selebgram* Indonesia, diidentifikasi apakah adanya tindakan *cyberbullying* dengan menggunakan metode *support vectore machine*. Dengan hanya menggunakan tahap *pre processing* pada data, diperoleh hasil penelitian sebesar 0.79 untuk tingkat akurasinya.

Selain itu, terdapat penelitian lain dengan menggunakan metode *random forest classifier* untuk mengidentifikasi *cyberbullying* pada twitter dengan data berbahasa Indonesia, pada penelitian ini data di kelompokan menjadi 50 *group tweet* serta terdapat delapan aturan yang dipakai, dan di setiap penentuan pada pohon keputusan digunakan dua aturan dari total delapan aturan yang ada, dengan metode ini didapatkan hasil *F1-Score* sebesar 0.9 dengan tingkat kesalahan 0.1, kesalahan tersebut terjadi karena adanya inkonsistensi hasil pada beberapa aturan yang ada (Novalita, Herdiani, Lukmana, & Puspandari, 2019).

Selain itu penelitian serupa pernah dilakukan untuk mendeteksi ujaran kebencian pada twitter menggunakan metode *naive bayes*. Data yang diperoleh dari twitter kemudian diolah melalui tahapan *pre-processing* berupa *cleaning*, *case folding*, *tokenisasi*, *stopword removal*, dan *stemming*. Kemudian digunakan metode *N-gram* untuk melakukan pembobotan kata dan seleksi fitur *information gain* untuk mengukur *lexical* teks untuk di klasifikasi. Dari penelitian tersebut kemudian diperoleh hasil akurasi sebesar 0.85, nilai presisi 0.92, nilai *recall* 0.92 dan nilai *f-measure* 0.79 (Hakiem & Fauzi, 2019).

Metode *deep learning* juga pernah digunakan untuk menangani permasalahan serupa, dengan menggunakan metode *convolutional neural networks* dilakukan pengujian terhadap data yang diperoleh dari twitter, data tersebut kemudian di identifikasi menggunakan *word embedding* dan kemudian dilakukan klasifikasi menggunakan metode CNN-CB. Dari penelitian tersebut diperoleh hasil sebesar 0.65, namun mengalami peningkatan menjadi 0.95 setelah 10 *epoch*, hal ini dikarenakan setiap periode yang dilakukan akan meningkatkan hasil prediksi. Selain metode tersebut, terdapat beberapa metode lain dalam *deep learning*, salah satunya adalah *long short term memory*, metode ini pernah digunakan untuk melakukan analisis sentimen pada *review* film, data yang digunakan dalam penelitian ini berasal dari IMDB, hasil penelitian tersebut mendapatkan hasil untuk akurasi sebesar 0.88 (Bodapati, Veeranjaneyulu, & Shaik, 2019).

**Tabel 2.1** Penelitian Sebelumnya

No	Judul	Substansi	Perbedaan
1	<i>Cyberbullying Comments Classification on Indonesian Selebgram Using Support Vectore Machine</i> Penulis: Miftah Andriansyah, Ali Akbar, Afina Ahwan, Ardiono Roma Nugraha, Nico Ariesto gilani, Rizky Nofita Sari dan Remi Sanjaya. 2017.	Model pembelajaran yang digunakan untuk identifikasi <i>cyberbullying</i> menggunakan algoritma <i>Support Vectore Machine</i> (SVM) yang dibagi ke dalam dua kelas, data training yang digunakan berjumlah 1053.	1. Menggunakan algoritma <i>Support Vectore Machine</i> 2. Data yang digunakan

**Tabel 2.2 Lanjutan Penelitian Sebelumnya**

2	Identifikasi <i>Tweet Cyberbullying</i> pada Aplikasi Twitter Menggunakan Metode <i>Support Vectore Machine</i> (SVM) dan <i>Information Gain</i> (IG) Sebagai Seleksi Fitur. Penulis: Ni Made Gita Dwi Purnamasari, M.Ali fauzi, Indriati dan Liana Sinta Dewi. 2018.	Algortima yang digunakan adalah <i>Support Vectore Machine</i> (SVM) dan ekstraksi fitur <i>information gain</i> untuk menghapus fitur yang berlebihan dan tidak relevan, data yang digunakan berjumlah 300 yang dikategorikan menjadi dua kelas dengan pembagian data 240 data <i>training</i> dan 60 data uji.	1. Menggunakan algoritma <i>Support Vectore Machine</i> untuk identifikasi 2. Menggunakan <i>information gain</i> (IG) untuk proses seleksi fitur. 3. Data yang digunakan
3	<i>Cyberbullying Revelation in Twitter Data Using Naïve Baiyes Classifier Algorithm</i> Penulis: Vandana Nandakumar. 2018.	Pada penelitian ini dilakukan komparasi algoritma <i>naïve bayes</i> dan SVM di mana <i>naïve bayes</i> memiliki presisi lebih tinggi dan nilai kompleksitas <i>runtime</i> yang lebih rendah.	1. Menggunakan algoritma <i>naïve bayes</i> clasifier 2. Data yang digunakan
4	<i>Cyberbullying Identification on Twitter using Random Forest Classifier</i> Penulis: N Novalaita, A.Herdiani, I. Lukmana dan D.Puspandari. 2018.	Algoritma yang digunakan adalah <i>random forest classifier</i> yang terdiri dari delapan aturan yang di kategorikan ke dalam dua kelas, data dalam pengujian ini dibagi 80% data <i>training</i> dan 20% data uji.	1. Menggunakan algoritma <i>random forrest Classifier</i> 2. Data yang digunakan.
5	<i>Cyberbullying Intervention Interface Based on Convolutional Neural Network</i> Penulis: Diana Inkpen. 2018.	Identifikasi <i>cyberbullying</i> dengan menggunakan metode CNN dibandingkan dengan SVM memiliki performa yang lebih baik dalam <i>F-score</i> , AUC dan tingkat <i>true positive</i> , fitur LIWC juga meningkatkan <i>F-Score</i> dan akurasi.	1. Menggunakan Algoritma <i>Convolutional Neural Network</i> . 2. Data yang digunakan.
6	<i>Deep Learing Algortihm for Cyberbullying Detection</i> Penulis: Monilah Abdilah Al-Ajlan dan Mouraj Ykhlef. 2018.	Deteksi <i>cyberbullying</i> data twitter menggunakan metode <i>deep learning</i> dengan algoritma CNN mendapatkan hasil kurang bagus para periode pertama yaitu hanya 0.65 untuk akurasinya, namun pada periode ke-10 hasil akurasi meningkat menjadi 0.95.	1. Menggunakan algoritma <i>Convulotional Neural Network-CB</i> . 2. Data yang digunakan.

Perbedaan penelitian ini dengan penelitian sebelumnya adalah metode yang digunakan dan data yang digunakan, Pada penelitian ini menggunakan algoritma *Long Short Term Memory* untuk proses identifikasi komentar dan data yang digunakan berasal dari kolom komentar karin novilda dan samuel alexander pieter yang diperoleh berdasarkan hasil kuesioner pada tanggal 12 februari 2020 sampai 17 februari 2020. Data kemudian akan di klasifikasikan ke dalam empat kelas, *cyberbullying*, *Irrelevant*, *netral* dan *non cyberbullying*.

## **BAB III**

### **METODOLOGI PENELITIAN DAN PENGEMBANGAN SISTEM**

#### **3.1. Metodologi Penelitian**

Metodologi penelitian dibagi menjadi dua yaitu metode pengumpulan data serta metode pengembangan sistem yang dijelaskan sebagai berikut:

##### **3.1.1. Metode Pengumpulan Data**

Dalam melakukan penelitian ini untuk mendapatkan data dan informasi, metode yang digunakan proses pengambilan data dilakukan sebagai berikut:

###### **1. Kuesioner**

Kuesioner merupakan alat riset atau survei yang terdiri atas serangkaian pertanyaan tertulis maupun *online*, yang bertujuan untuk mendapatkan tanggapan dari kelompok orang terpilih. Bentuk pertanyaan pada penelitian ini adalah kuesioner tertutup . Kuesioner tertutup merupakan daftar pertanyaan yang alternatif jawabannya telah disediakan oleh peneliti. Dalam kuesioner ini alat ukur yang digunakan untuk mengetahui pengalaman responden terhadap tindakan *cyberbullying* adalah *Cyberbullying and Online Aggression Survey 2019 Version* dari Sameer Hinduja, Ph.D. dan Justin W. Patchin, Ph.D. yang terdiri dari 20 pertanyaan, yang disesuaikan dengan penelitian ini dengan alternatif jawaban yaitu tidak pernah, sekali, beberapa kali dan sering yang disajikan dalam skala *linkert* dengan nilai nol sampai tiga. Penyebaran kuesioner dilakukan secara *online* melalui media sosial whatsapp dan line dengan bantuan *google form*.

###### **a) Kuesioner Pertanyaan Nomor Satu (1) dan Dua (2)**

Pertanyaan tentang identitas diri dari responden berupa nama dan jenis kelamin dengan tujuan untuk mengetahui jumlah persentase pengalaman responden laki-laki dan

perempuan terhadap fenomena *cyberbullying*. Rancangan pertanyaan dapat dilihat pada gambar 3.1.

The image shows a screenshot of a survey form. At the top, there is a field labeled "Nama \*". Below it is a text input field labeled "Jawaban Anda". A horizontal blue bar separates this from the next section. In the second section, there is a field labeled "Jenis Kelamin \*". Below it is a dropdown menu with the placeholder "Pilih".

**Gambar 3.1** Kuesioner Pertanyaan Nomor 1 dan 2

b) Kuesioner Pertanyaan Nomor Tiga (3)

Pertanyaan kuesioner selanjutnya adalah usia responden. Tujuan dari pertanyaan ini adalah mendapatkan rataan umur responden yang menggunakan internet dan mengetahui tentang *cyberbullying*. Jawaban pada pertanyaan ini diisi secara manual oleh responden dengan tujuan untuk mendapatkan responden dengan umur yang beragam. Rancangan pertanyaan dapat dilihat pada gambar 3.2.

The image shows a screenshot of a survey form. At the top, there is a field labeled "Usia \*". Below it is a text input field with the placeholder "misalnya jika anda berusia 20 tahun, cukup isikan 20". A horizontal blue bar separates this from the next section. In the second section, there is a text input field labeled "Jawaban Anda".

**Gambar 3.2** Kuesioner Pertanyaan Nomor Tiga (3)

c) Kuesioner Pertanyaan Nomor Empat (4)

Pertanyaan kuesioner selanjutnya adalah mengenai profesi atau pekerjaan dari responden. Ada beberapa alternatif jawaban yang ditampilkan di antaranya mahasiswa, pelajar, wirausaha atau wiraswasta, karyawan dan alternatif jawaban yang lain untuk memberikan kesempatan kepada responden menuliskan pekerjaannya jika alternatif jawaban

yang diberikan sebelumnya tidak sesuai. Rancangan pertanyaan ini dapat dilihat pada gambar 3.3.

Pekerjaan \*

- Mahasiswa
- Pelajar
- Wiraswasta atau wirausaha
- Karyawan
- Yang lain:

Gambar 3.3 Kuesioner Pertanyaan Nomor empat (4)

d) Kuesioner Pertanyaan Nomor Lima (5)

Pertanyaan kuesioner selanjutnya adalah mengenai pertanyaan mengenai penggunaan internet dalam satu tahun terakhir. Pertanyaan ini dimaksudkan untuk memvalidasi responden ke pertanyaan selanjutnya agar tidak ada data yang tidak valid. Ada dua alternatif pilihan yang ditampilkan yaitu ya dan tidak. Rancangan pertanyaan dapat dilihat pada pertanyaan 3.4.

Apakah anda telah menggunakan internet dalam satu tahun terakhir? \*

- Ya
- Tidak

Gambar 3.4 Pertanyaan Kuesioner Nomor Lima (5)

e) Pendahuluan

Sebelum kuesioner mengenai *cyberbullying* dilakukan, terlebih dahulu dijelaskan mengenai gambaran secara singkat apa itu *cyberbullying* dan alat ukur yang digunakan dalam skala jawaban di mana nilai nol diartikan tidak pernah, nilai satu diartikan sekali, nilai dua dapat diartikan beberapa kali dan nilai tiga diartikan sering. Rancangan pendahuluan dapat dilihat pada gambar 3.5.

**Pendahuluan**

Cyberbullying adalah ketika seseorang melakukan tindakan untuk melecehkan, menganiaya, atau mengolok-olok orang lain dengan tujuan untuk melukai mereka secara online atau saat menggunakan ponsel atau perangkat elektronik lainnya. Orang lain dalam kuesioner ini dapat diartikan sebagai teman, kenalan, public figure dan lain sebagainya.

dalam penelitian ini, skala yang digunakan bernilai 0-3  
 0 = tidak pernah  
 1 = sekali  
 2 = beberapa kali  
 3 = sering

**Gambar 3.5 Pendahuluan**

## f) Kuesioner Pertanyaan Nomor Enam (6)

Pertanyaan kuesioner selanjutnya adalah mengenai pengalaman responden dalam menemukan komentar jahat atau menyakitkan yang ditujukan kepada orang lain. Pertanyaan ini dimaksudkan untuk mengetahui persentase responden yang menemukan komentar jahat kepada orang lain. Alternatif jawaban yang disediakan dalam pertanyaan ini berupa skala *linkert* dengan *range* nol sampai tiga dengan bobot yang telah dijelaskan di awal. Rancangan pertanyaan ini dapat dilihat pada gambar 3.6.

Saya pernah melihat seseorang berkomentar jahat atau menyakitkan kepada orang lain di internet \*

0	1	2	3	
Tidak pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.6 Kuesioner Pertanyaan Nomor Enam (6)**

## g) Kuesioner Pertanyaan Nomor Tujuh (7)

Pertanyaan nomor enam ditujukan kepada responden yang pernah menemukan tindakan *cyberbullying* berupa kiriman foto orang lain secara kejam atau menyakitkan. Rancangan pertanyaan dapat dilihat pada gambar 3.7.

Saya pernah melihat seseorang memposting foto orang lain secara kejam atau menyakitkan di internet \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.7 Kuesioner Pertanyaan Nomor Tujuh (7)**

h) Kuesioner Pertanyaan Nomor Delapan (8)

Pertanyaan kuesioner selanjutnya diajukan dengan maksud untuk mengetahui pengalaman responden dalam menemukan tindakan *cyberbullying* berupa penyebaran video yang memalukan atau menyakitkan bagi orang lain. Rancangan pertanyaan tersebut dapat dilihat pada gambar 3.8.

Saya pernah melihat seseorang memposting video memalukan atau menyakitkan bagi orang lain di internet *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.8** Kuesioner Pertanyaan Nomor Delapan (8)

i) Kuesioner Pertanyaan Nomor Sembilan (9)

Pertanyaan kuesioner selanjutnya dimaksudkan untuk mengetahui pengalaman responden dalam menemukan tindakan *cyberbullying* berupa penyebaran gosip terhadap orang lain. Rancangan pertanyaan dapat dilihat pada gambar 3.9.

Saya pernah melihat seseorang menyebarkan gosip tentang orang lain di internet *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.9** Kuesioner Pertanyaan Nomor Sembilan (9)

j) Kuesioner Pertanyaan Nomor Sepuluh (10)

Pertanyaan selanjutnya dimaksudkan untuk mengetahui pengalaman responden terhadap tindakan *cyberbullying* berupa tindakan pengancaman atau melukai yang ditujukan kepada orang lain. Rancangan Pertanyaan tersebut dapat dilihat pada gambar 3.10.

Saya pernah melihat seseorang mengancam akan melukai orang lain di internet \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.10** Kuesioner Pertanyaan Nomor Sepuluh (10)

## k) Kuesioner Pertanyaan Nomor Sebelas (11)

Pertanyaan kuesioner selanjutnya adalah mengenai tindakan *cyberbullying* berupa berpura-pura menjadi orang lain dan melakukan tindakan yang merugikan orang yang ditiru. Pertanyaan ini dimaksudkan untuk mengetahui pengalaman responden terhadap tindakan tersebut. Rancangan pertanyaan dapat dilihat pada gambar 3.11.

Saya pernah melihat seseorang berpura-pura menjadi orang lain dan bertindak tidak pantas di internet \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.11** Kuesioner Pertanyaan Nomor Sebelas (11)

## l) Kuesioner Pertanyaan Nomor Dua Belas (12)

Pertanyaan selanjutnya ditampilkan dengan maksud untuk mengetahui pengalaman responden terhadap komentar kasar tentang ras atau warna kulit terhadap orang lain. Rancangan pertanyaan dapat dilihat pada gambar 3.12

Saya pernah melihat seseorang berkomentar kasar di internet tentang ras atau warna kulit orang lain \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.12** Kuesioner Pertanyaan Nomor Dua Belas (12)

m) Kuesioner Pertanyaan Nomor Tiga Belas (13)

Pertanyaan kuesioner selanjutnya diajukan dengan maksud untuk mengetahui pengalaman responden terhadap komentar kasar yang dilakukan seseorang terhadap orang lain dengan makna seksualitas. Rancangan pertanyaan dapat dilihat pada gambar 3.13.

Saya pernah melihat seseorang berkomentar kasar di internet tentang orang lain dengan makna seksualitas *				
	0	1	2	3
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.13** Kuesioner Pertanyaan Nomor Tiga Belas (13)

n) Kuesioner Pertanyaan Nomor Empat Belas (14)

Pertanyaan nomor empat belas merupakan pertanyaan mengenai tindakan *cyberbullying* berupa komentar kasar terhadap agama orang lain. Pertanyaan diajukan untuk mengetahui pengalaman responden terhadap tindakan tersebut. Rancangan pertanyaan dapat dilihat pada gambar 3.14.

Saya pernah melihat seseorang berkomentar kasar di internet tentang agama orang lain *				
	0	1	2	3
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.14** Kuesioner Pertanyaan Nomor Empat Belas (14)

o) Kuesioner Pertanyaan Nomor Lima Belas (15)

Pertanyaan selanjutnya dilakukan untuk mengetahui apakah responden pernah menjadi korban atau mengalami beberapa kejadian dari tindakan *cyberbullying* pada pertanyaan sebelumnya. Rancangan Pertanyaan dapat dilihat pada gambar 3.15.

Saya pernah mengalami beberapa kejadian tersebut \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

**Gambar 3.15** Kuesioner Pertanyaan Nomor Lima Belas (15)

## p) Kuesioner Pertanyaan Nomor Enam Belas (16) sampai Dua Puluh (20)

Pertanyaan kuesioner pada bagian ini diajukan dengan maksud untuk mengetahui media atau tempat manakah yang terdapat tindakan *cyberbullying* berdasarkan pengalaman dari responden. Pada pertanyaan bagian ini disajikan lima pertanyaan dengan alternatif jawaban berupa skala *linkert*. Lima media sosial yang ditanyakan adalah *group chat*, instagram, twitter, facebook dan media sosial lainnya. Dengan pertanyaan ini dapat ditemukan media sosial mana yang sering ditemukan tindakan *cyberbullying*. Rancangan pertanyaan ini dapat dilihat pada gambar 3.16.

Saya menyaksikan tindakan cyberbullying di chat room (ruang obrolan) \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

Saya menyaksikan tindakan cyberbullying di Instagram \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

Saya menyaksikan tindakan cyberbullying di Twitter \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

Saya menyaksikan tindakan cyberbullying di Facebook \*

0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

Saya menyaksikan tindakan cyberbullying di media sosial lainnya \*

0	1	2	3
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Gambar 3.16** Kuesioner Pertanyaan Nomor 16 sampai 20

q) Kuesioner Pertanyaan Nomor Dua Puluh Satu sampai Dua Puluh Lima

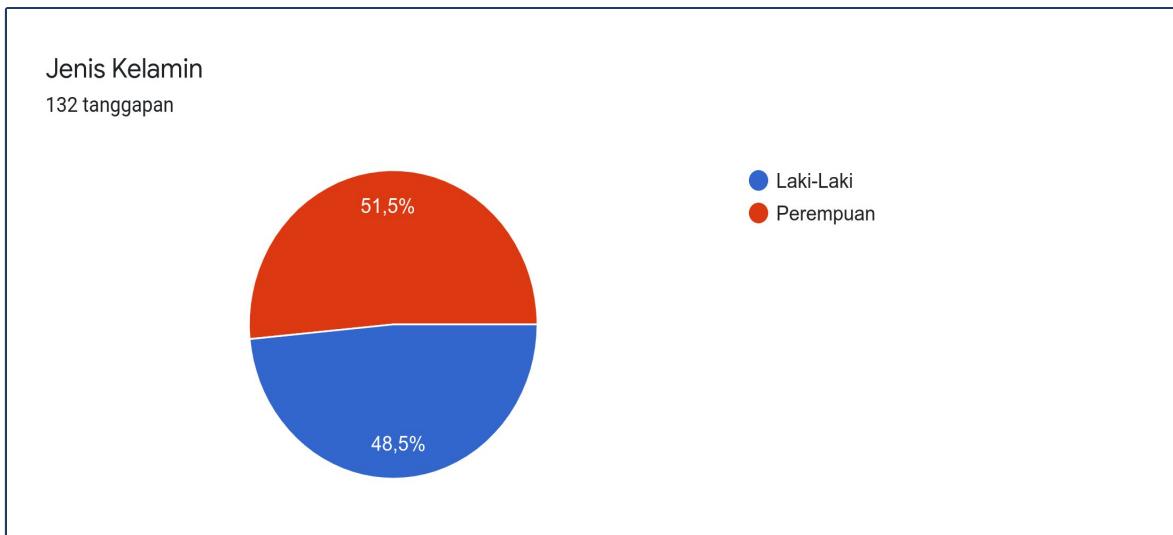
Pertanyaan Kuesioner ini ditanyakan untuk mengetahui pengalaman responden terhadap tindakan *cyberbullying* yang dialami oleh *influencer*. *Influencer* dipilih karena dianggap sebagai profesi yang sering mengalami tindakan *cyberbullying*, untuk membuktikan asumsi tersebut pada pertanyaan ini disajikan lima nama *influencer* di Indonesia dengan alternatif jawaban berupa skala *linkert*. Lima *influencer* tersebut adalah Deddy corbuzier, Via vallen, Atta halilintar, Samuel alexander pieter dan Karin novilda. Hasil dari jawaban pertanyaan ini akan menjadi studi kasus penelitian dalam hal pengambilan data penelitian. Rancangan pertanyaan dapat dilihat pada gambar 3.17.

Saya menyaksikan tindakan cyberbullying kepada Deddy Corbuzier *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering
Saya menyaksikan tindakan cyberbullying kepada Via Vallen *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering
Saya menyaksikan tindakan cyberbullying kepada Atta Halilintar *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering
Saya menyaksikan tindakan cyberbullying kepada Samuel Alexander Pieter (Young Lex) *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering
Saya menyaksikan tindakan cyberbullying kepada Karin Novilda (Awkarin) *				
0	1	2	3	
Tidak Pernah	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/> Sering

Gambar 3.17 Kuesioner Pertanyaan Nomor 21 sampai 25

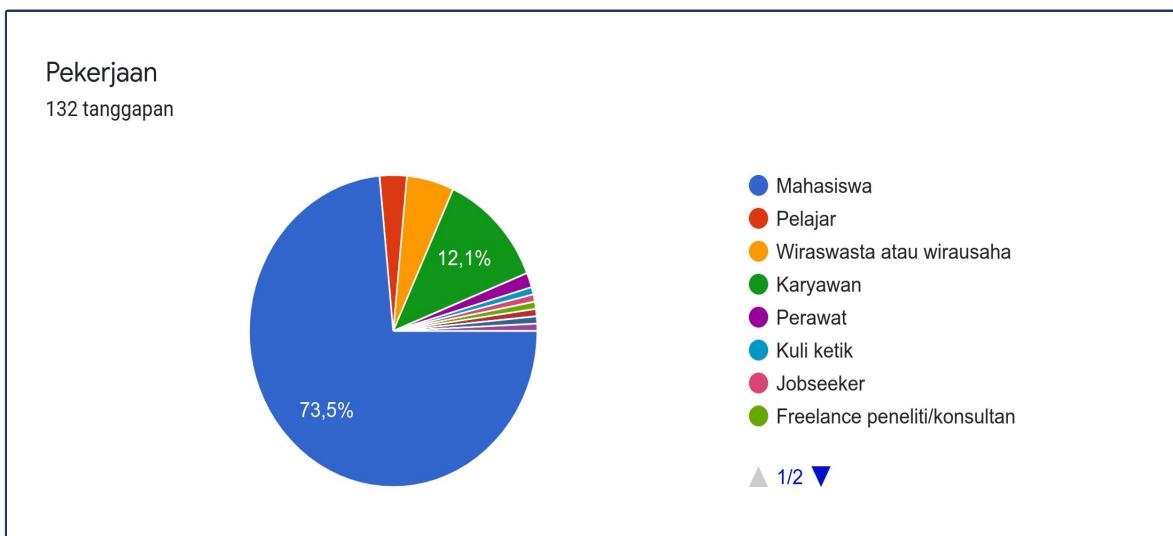
r) Grafik Hasil Kuesioner

Berdasarkan kuesioner yang telah dilakukan, didapatkan hasil sejumlah 132 tanggapan. Tanggapan tersebut kemudian ditampilkan dalam grafik sesuai dengan jenis pertanyaan. Dalam pertanyaan dengan opsi seperti pada pertanyaan nomor dua mengenai jenis kelamin, hasil kuesioner dapat dilihat pada gambar 3.18.



**Gambar 3.18** Hasil Kuesioner Pertanyaan Nomor Dua (2)

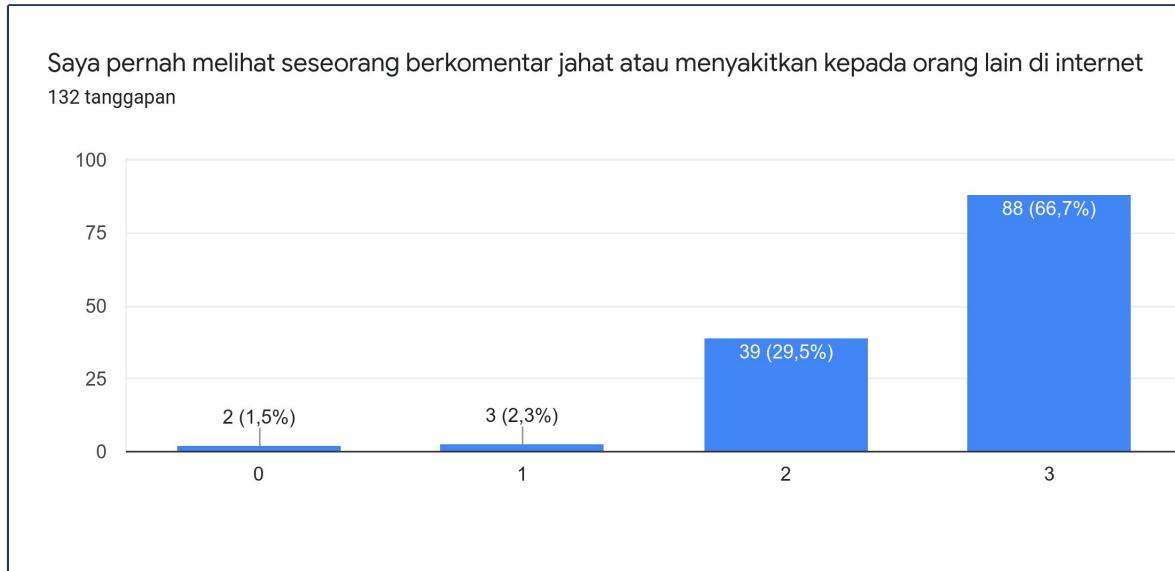
Kemudian untuk hasil pada pertanyaan nomor empat mengenai pekerjaan hasil dapat dilihat pada gambar 3.19.



**Gambar 3.19** Hasil Pertanyaan Kuesioner Nomor Empat (4)

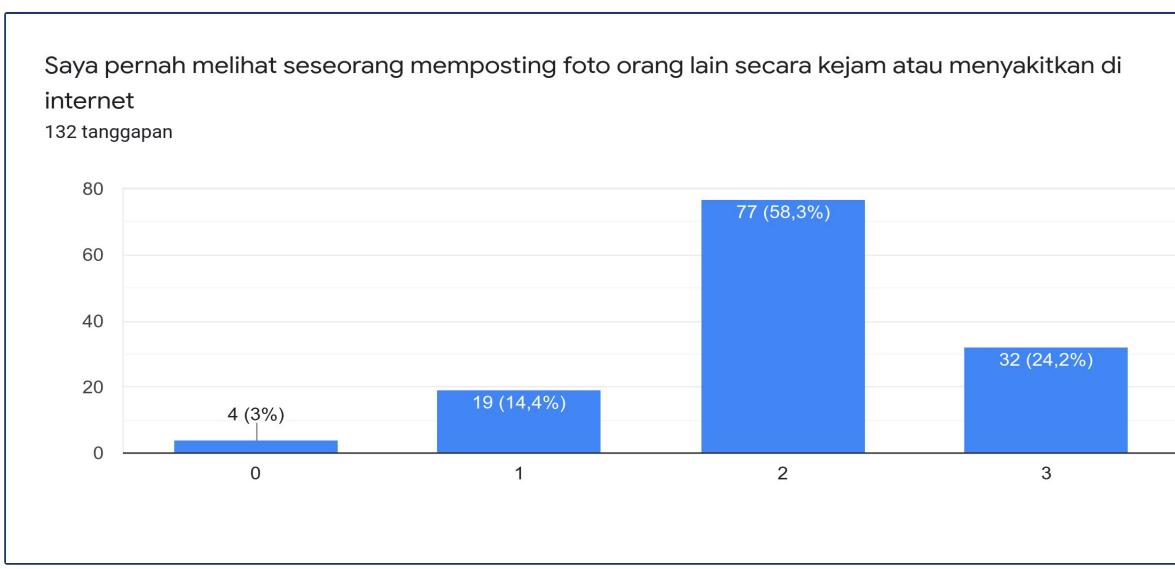
Kemudian untuk pertanyaan dengan jawaban *skala linkert*, hasil jawaban ditampilkan dalam bentuk diagram batang sesuai dengan opsi jawaban. Hasil kuesioner pertanyaan nomor enam

mengenai pengalaman responden terhadap komentar jahat atau menyakitkan terhadap orang lain dapat dilihat pada gambar 3.20.



**Gambar 3.20** Hasil Kuesioner Pertanyaan Nomor Enam (6)

Hasil kuesioner pertanyaan nomor tujuh mengenai pengalaman responden terhadap jenis *cyberbullying* dengan menyebarkan foto orang lain secara kejam dapat dilihat pada gambar 3.21.



**Gambar 3.21** Hasil Kuesioner Pertanyaan Nomor Tujuh (7)

Hasil kuesioner pertanyaan nomor delapan mengenai pengalaman responden terhadap jenis *cyberbullying* dengan menyebarkan video memalukan orang lain dapat dilihat pada gambar 3.22.



**Gambar 3.22** Hasil Kuesioner Pertanyaan Nomor Delapan (8)

Hasil kuesioner pertanyaan nomor sembilan mengenai pengalaman responden terhadap jenis *cyberbullying* dengan menyebarkan gosip orang lain dapat dilihat pada gambar 3.23.



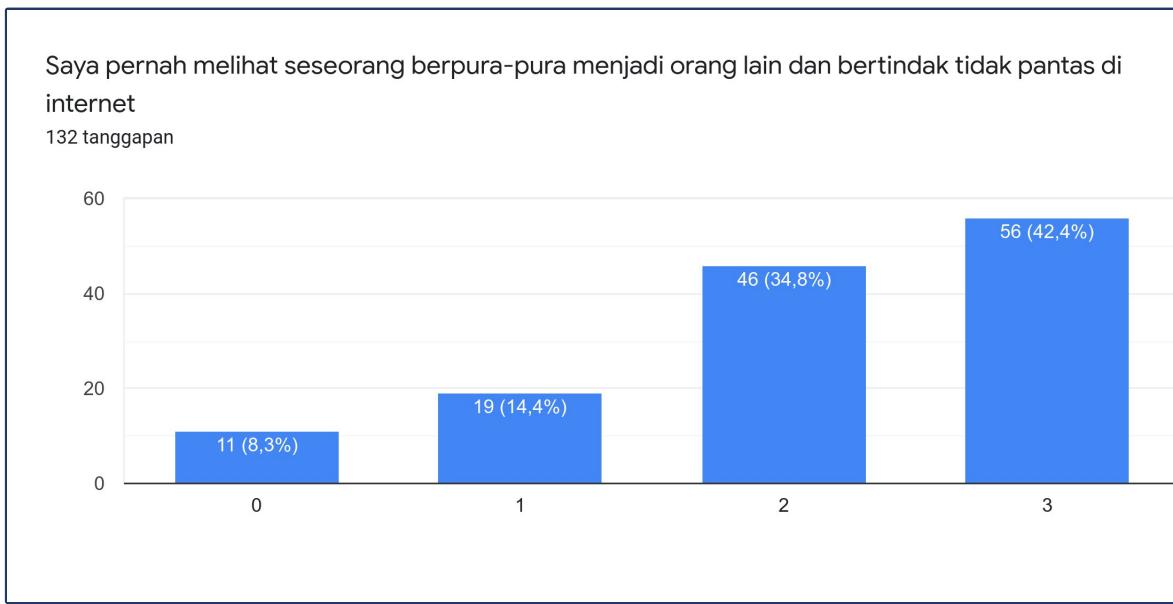
**Gambar 3.23** Hasil Kuesioner Pertanyaan Nomor Sembilan (9)

Hasil kuesioner pertanyaan nomor sepuluh mengenai pengalaman responden terhadap jenis *cyberbullying* dengan mengancam orang lain dapat dilihat pada gambar 3.24.



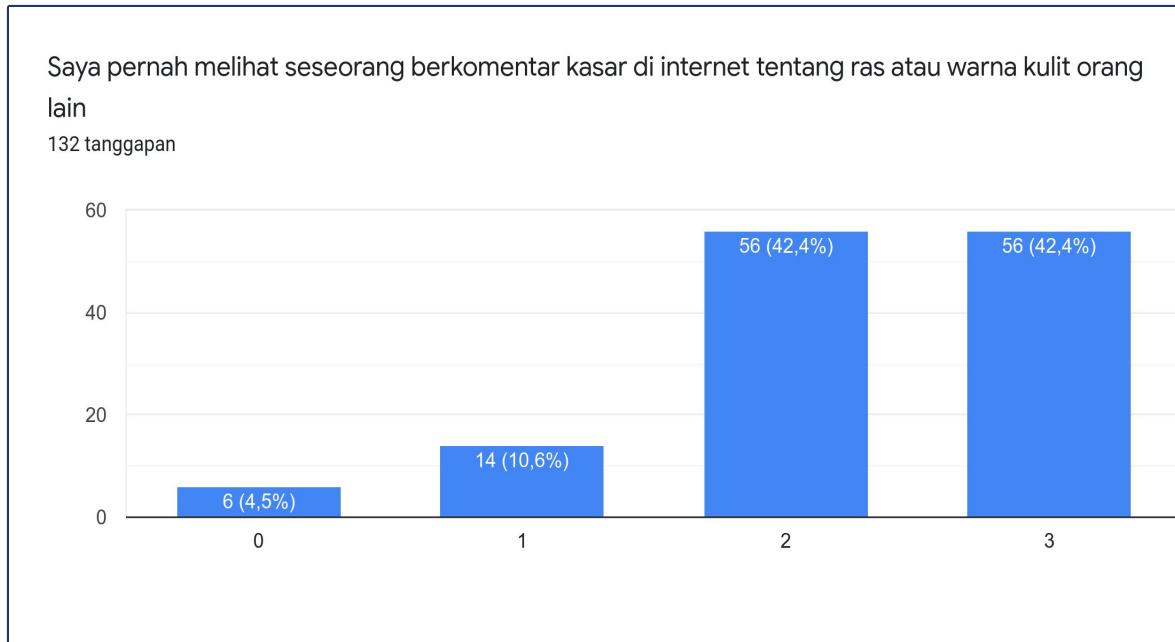
**Gambar 3.24** Hasil Kuesioner Pertanyaan Nomor Sepuluh (10)

Hasil kuesioner pertanyaan nomor sebelas mengenai pengalaman responden terhadap jenis *cyberbullying* berpura-pura menjadi orang lain dan bertindak tidak pantas dapat dilihat pada gambar 3.25.



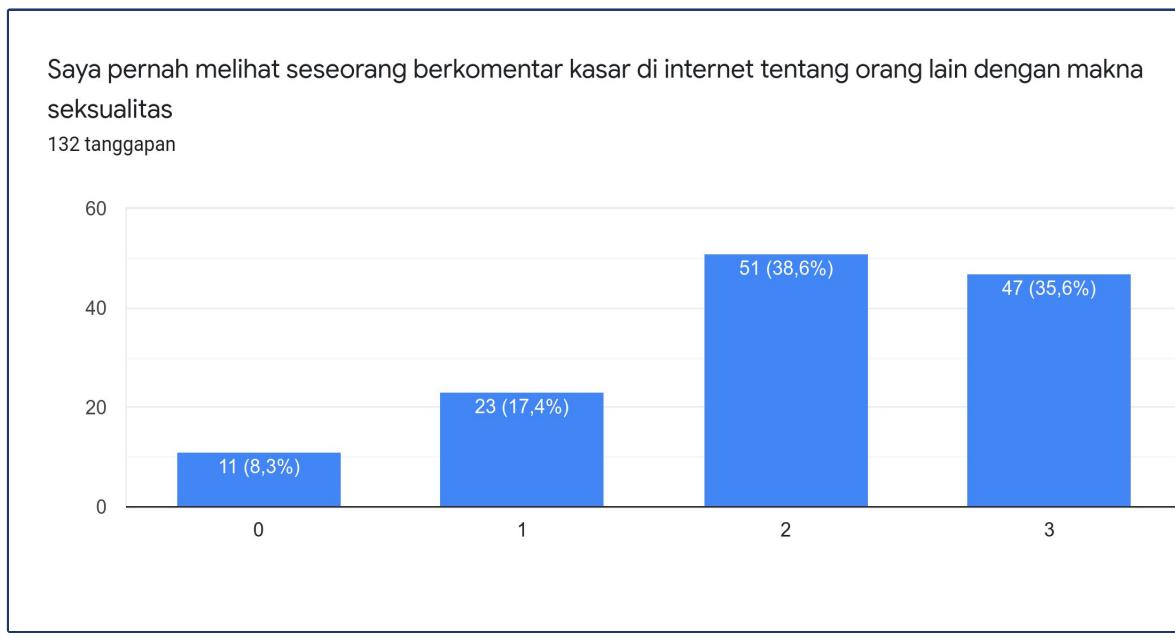
**Gambar 3.25** Hasil Kuesioner Pertanyaan Nomor Sebelas (11)

Hasil kuesioner pertanyaan nomor dua belas mengenai pengalaman responden terhadap jenis *cyberbullying* komentar kasar tentang ras atau warna kulit dapat dilihat pada gambar 3.26.



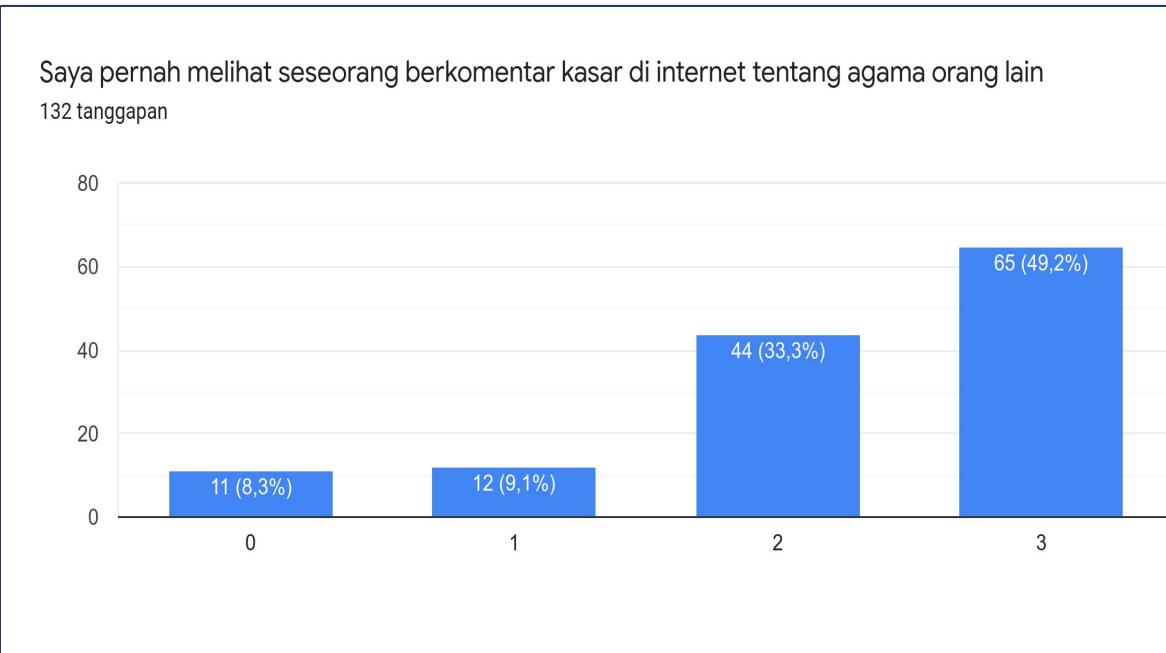
**Gambar 3.26** Hasil Kuesioner Pertanyaan Nomor Dua Belas (12)

Hasil kuesioner pertanyaan nomor tiga belas mengenai pengalaman responden terhadap jenis *cyberbullying* komentar kasar dengan makna seksualitas dapat dilihat pada gambar 3.27.



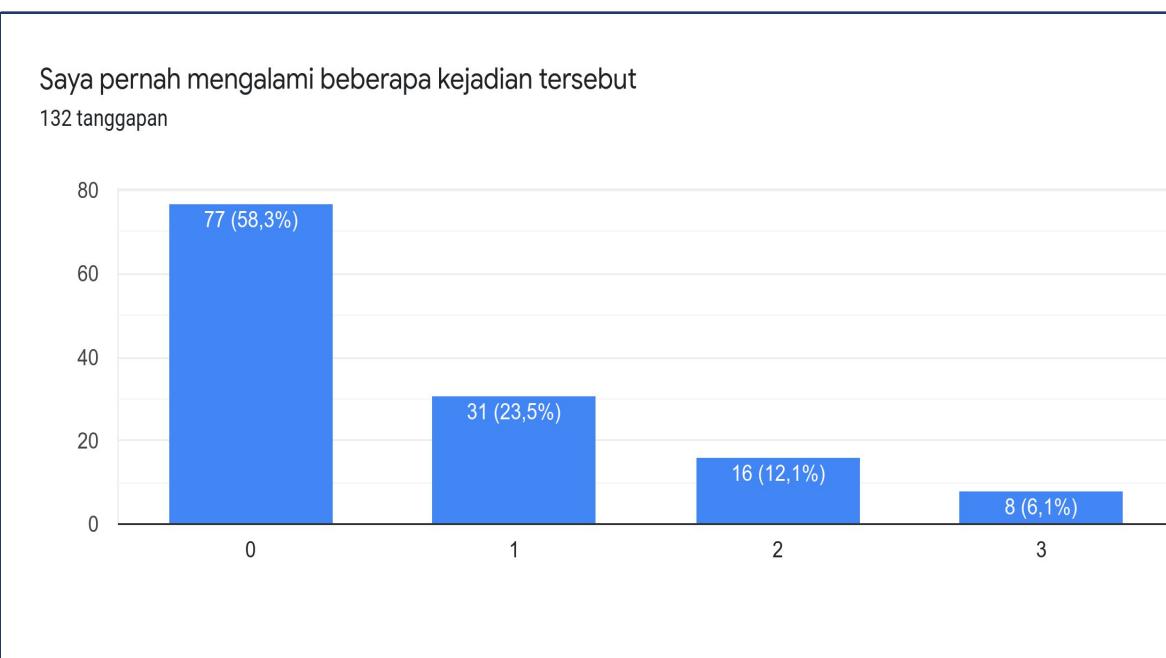
**Gambar 3.27** Hasil Kuesioner Pertanyaan Nomor Tiga Belas (13)

Hasil kuesioner pertanyaan nomor empat belas mengenai pengalaman responden terhadap jenis *cyberbullying* komentar kasar tentang agama dapat dilihat pada gambar 3.28.



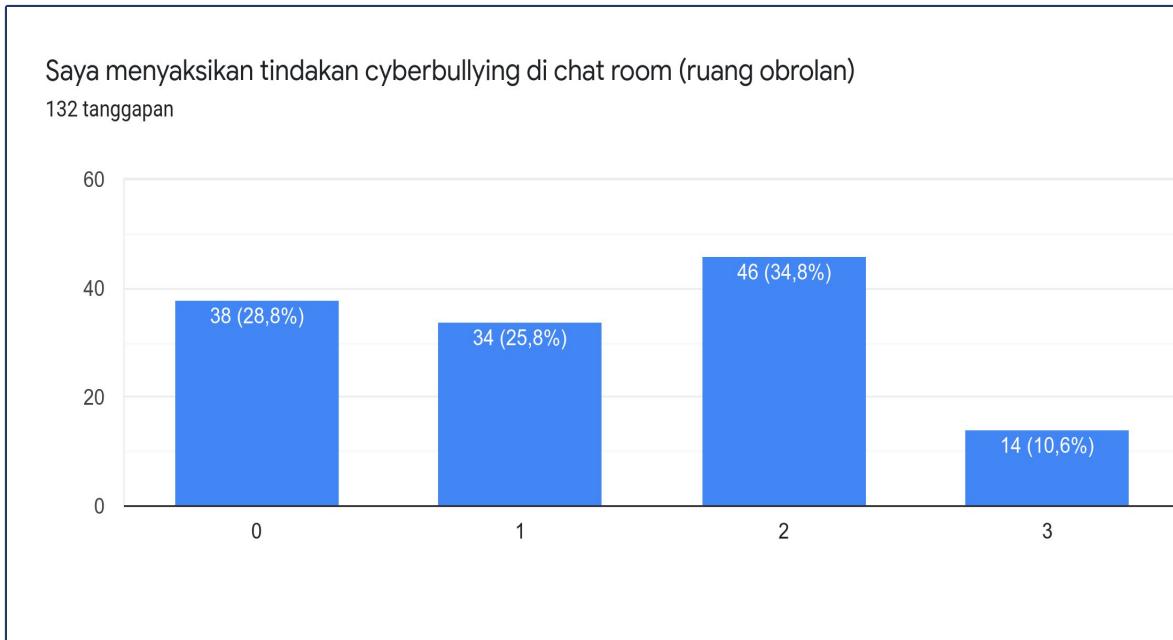
**Gambar 3.28** Hasil Kuesioner Pertanyaan Nomor Empat Belas (14)

Hasil kuesioner pertanyaan nomor lima belas mengenai pengalaman responden terhadap tindakan *cyberbullying* terhadap dirinya sendiri dapat dilihat pada gambar 3.29.



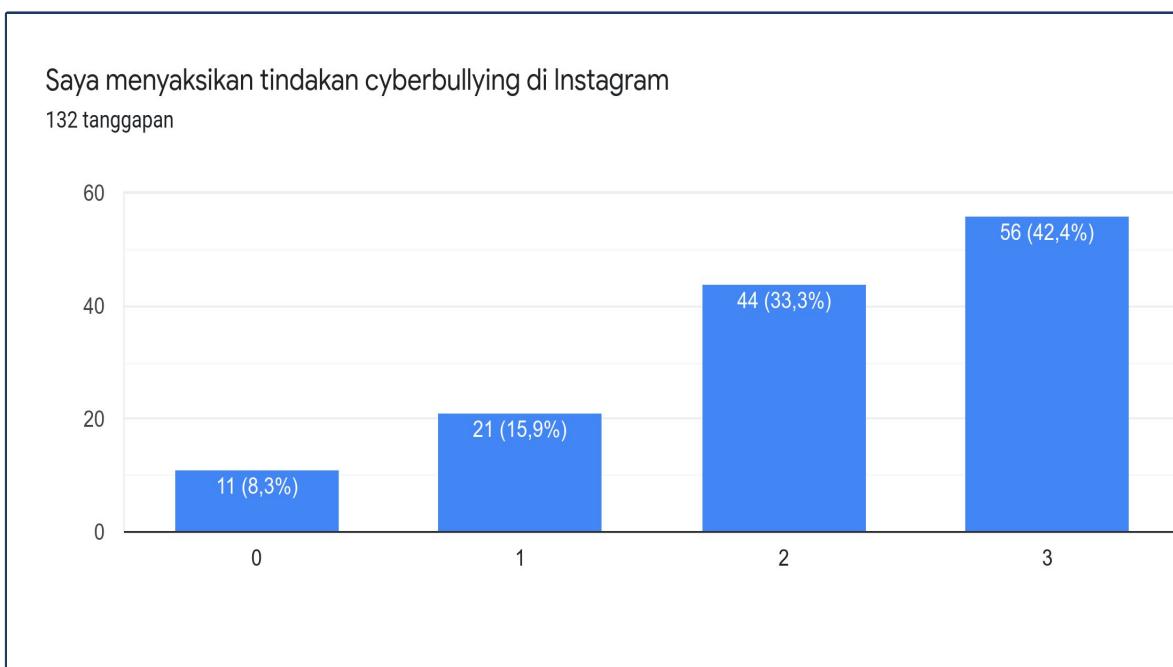
**Gambar 3.29** Hasil Kuesioner Pertanyaan Nomor Lima Belas (15)

Hasil kuesioner pertanyaan nomor enam belas mengenai pengalaman responden terhadap tindakan *cyberbullying* pada media *chat room* dapat dilihat pada gambar 3.30.



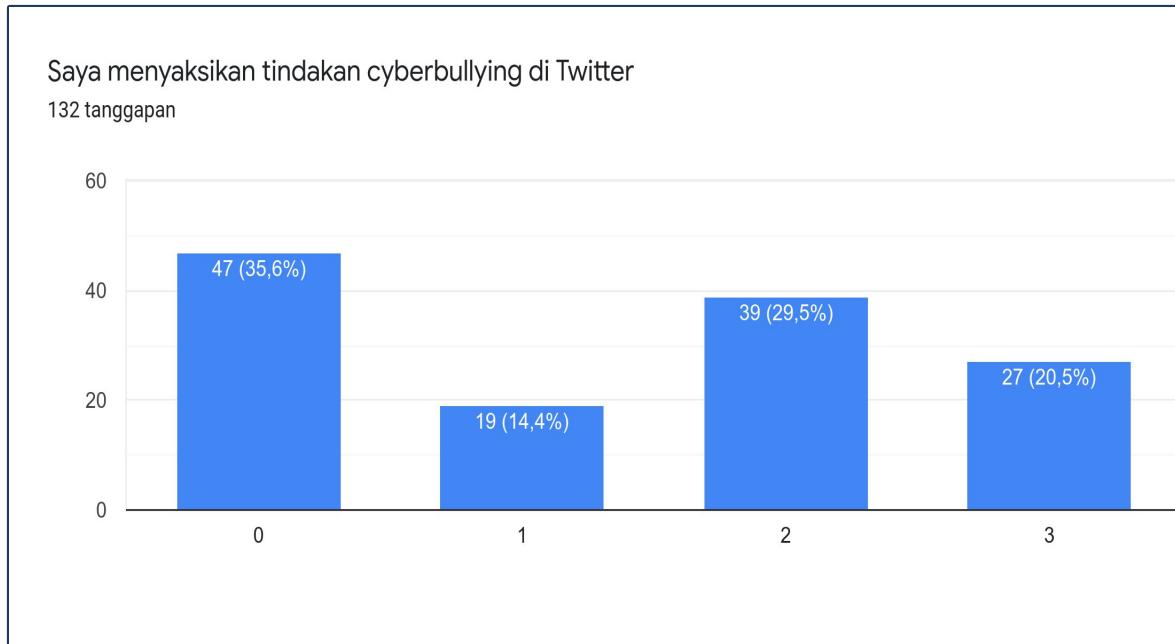
**Gambar 3.30** Hasil Kuesioner Pertanyaan Nomor Enam Belas (16)

Hasil kuesioner pertanyaan nomor tujuh belas mengenai pengalaman responden terhadap tindakan *cyberbullying* pada media *instagram* dapat dilihat pada gambar 3.31.



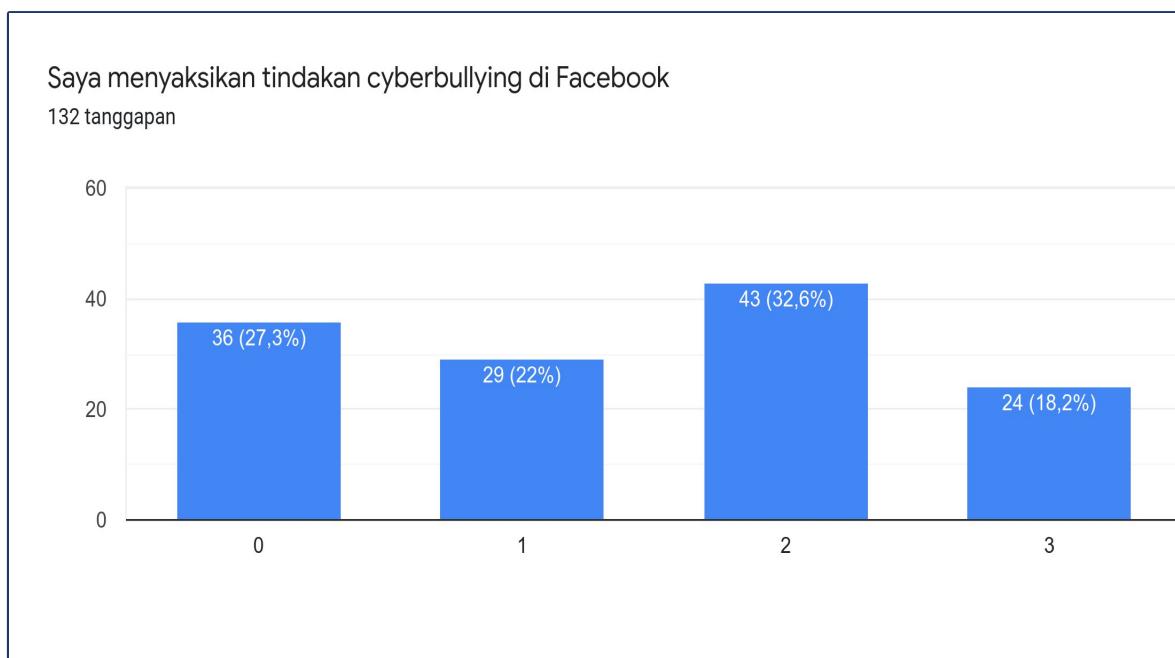
**Gambar 3.31** Hasil Kuesioner Pertanyaan Nomor Tujuh Belas (17)

Hasil kuesioner pertanyaan nomor delapan belas mengenai pengalaman responden terhadap tindakan *cyberbullying* pada media *twitter* dapat dilihat pada gambar 3.32.



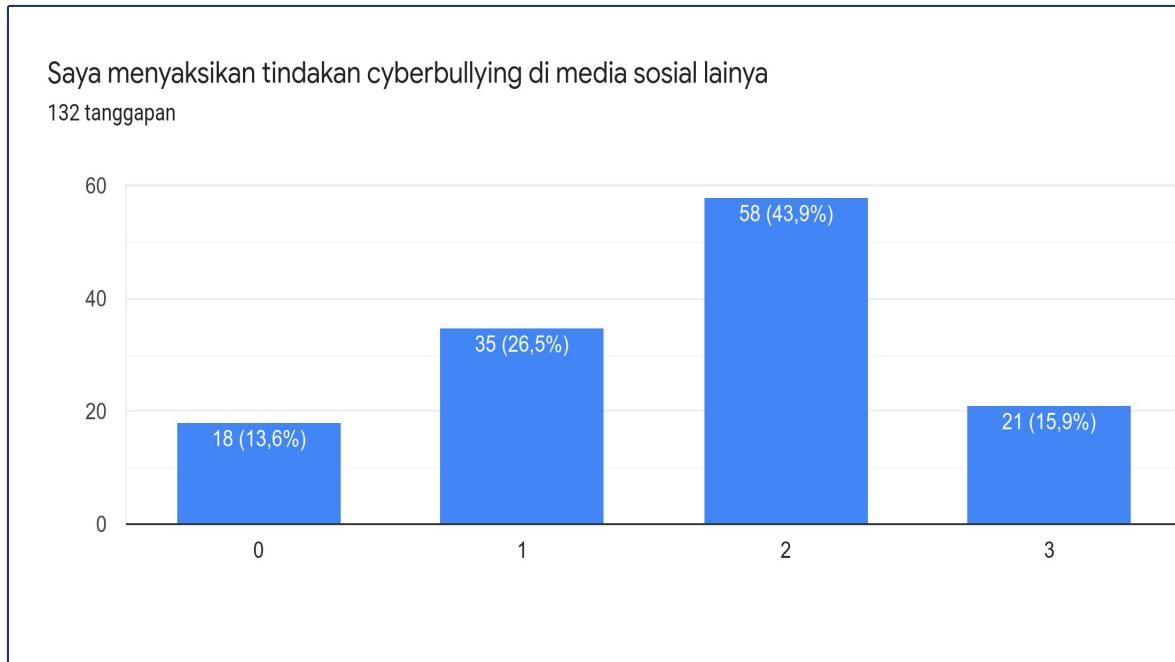
**Gambar 3.32** Hasil Kuesioner Pertanyaan Nomor Delapan Belas (18)

Hasil kuesioner pertanyaan nomor sembilan belas mengenai pengalaman responden terhadap tindakan *cyberbullying* pada media *facebook* dapat dilihat pada gambar 3.33.



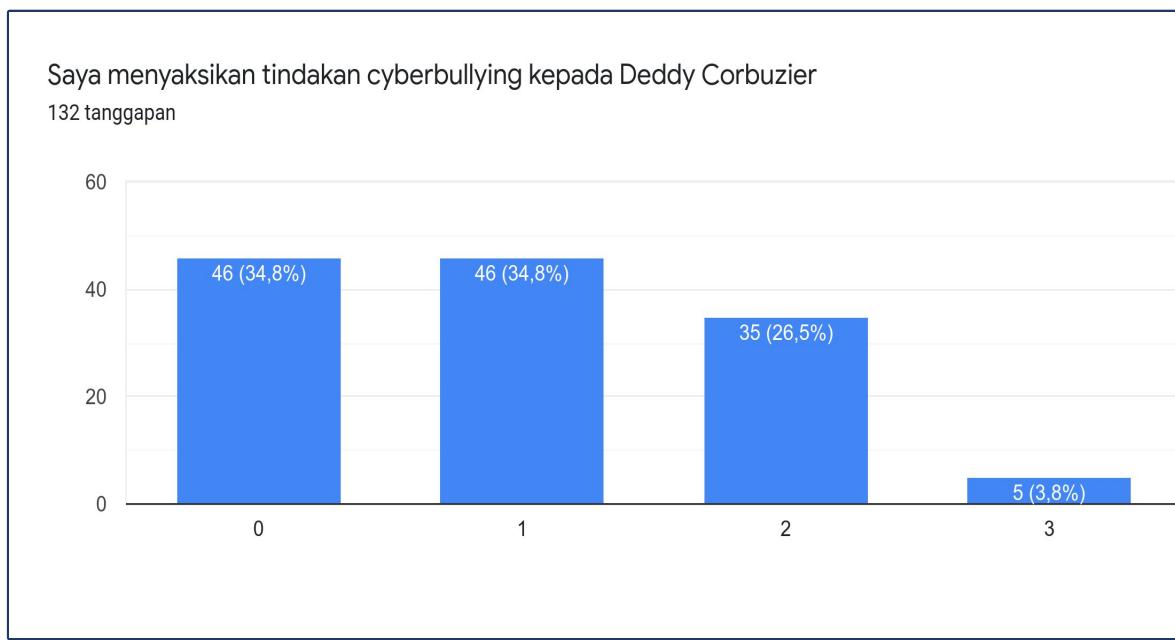
**Gambar 3.33** Hasil Kuesioner Pertanyaan Nomor Sembilan Belas (19)

Hasil kuesioner pertanyaan nomor dua puluh mengenai pengalaman responden terhadap tindakan *cyberbullying* pada media sosial lainnya dapat dilihat pada gambar 3.34.



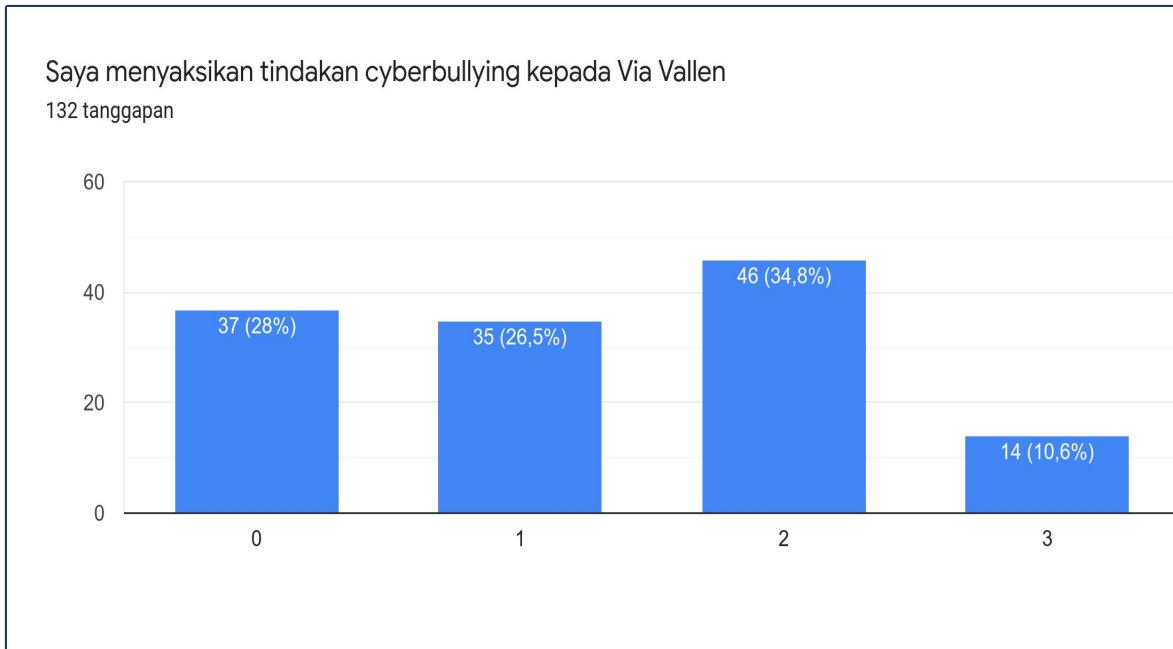
**Gambar 3.34** Hasil Kuesioner Pertanyaan Nomor Dua Puluh (20)

Hasil kuesioner pertanyaan nomor dua puluh satu mengenai pengalaman responden terhadap tindakan *cyberbullying* terhadap Deddy corbuzier dapat dilihat pada gambar 3.35.



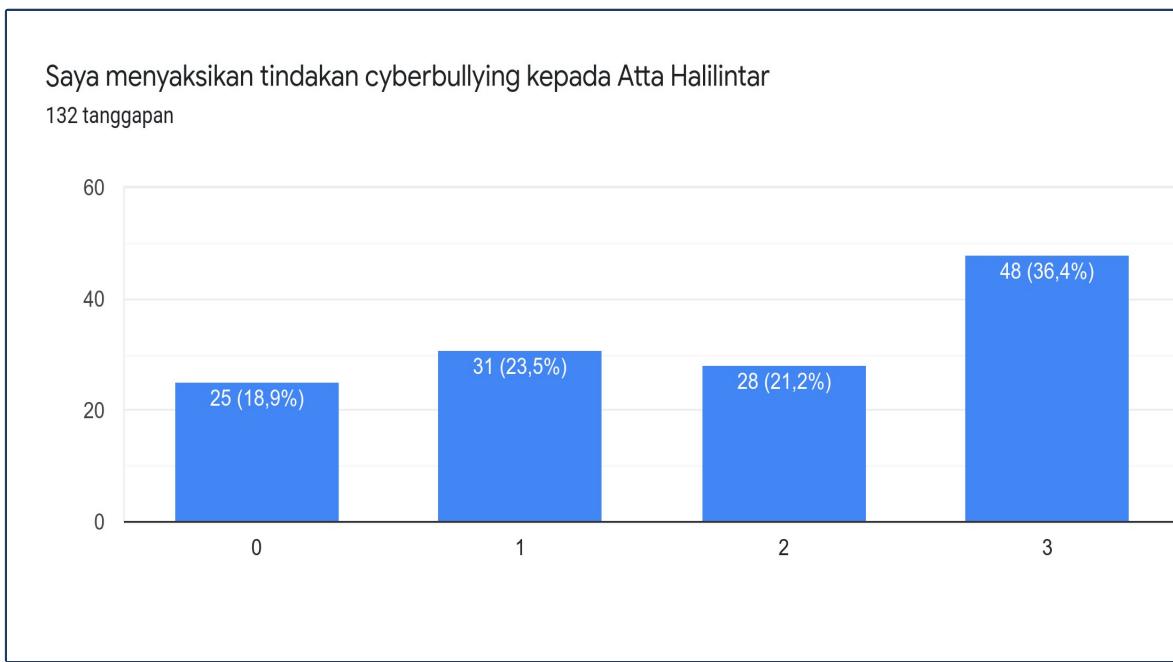
**Gambar 3.35** Hasil Kuesioner Pertanyaan Nomor Dua Puluh Satu (21)

Hasil kuesioner pertanyaan nomor dua puluh dua mengenai pengalaman responden terhadap tindakan *cyberbullying* terhadap Via vallen dapat dilihat pada gambar 3.36.



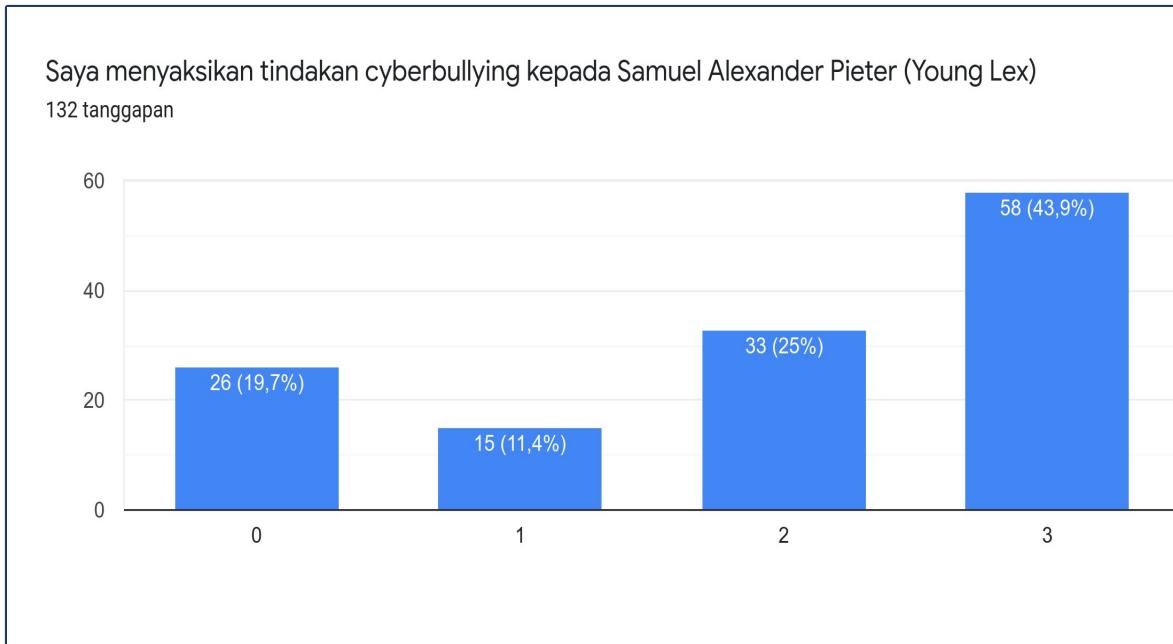
**Gambar 3.36** Hasil Kuesioner Pertanyaan Nomor Dua Puluh Dua (22)

Hasil kuesioner pertanyaan nomor dua puluh tiga mengenai pengalaman responden terhadap tindakan *cyberbullying* terhadap Atta halilintar dapat dilihat pada gambar 3.37.



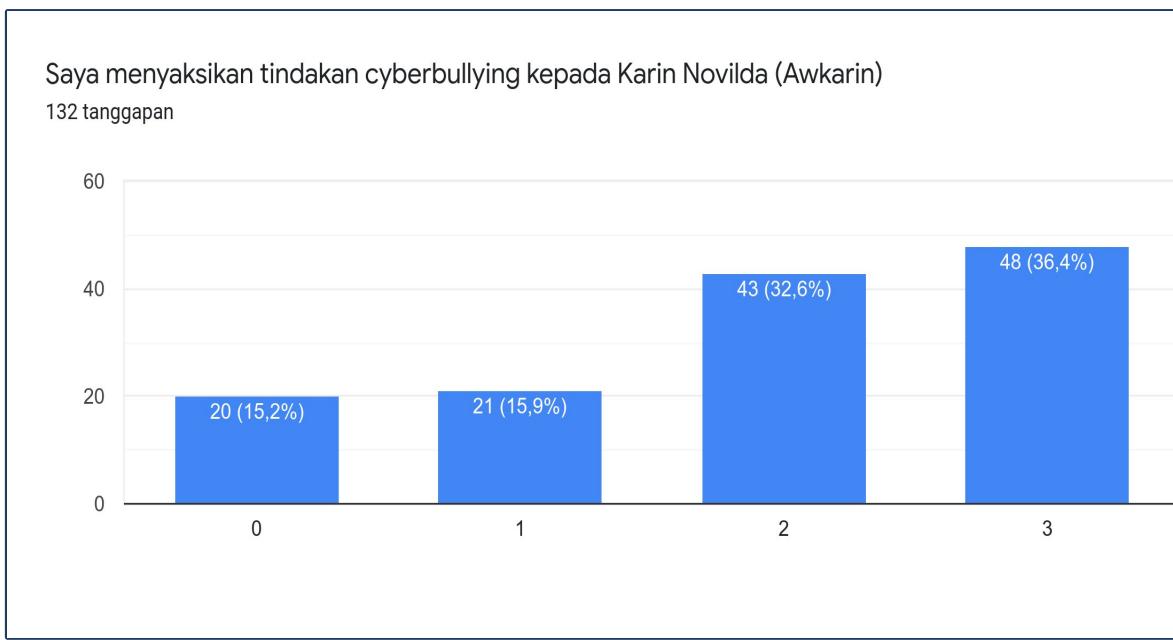
**Gambar 3.37** Hasil Kuesioner Pertanyaan Nomor Dua Puluh Tiga (23)

Hasil kuesioner pertanyaan nomor dua puluh empat mengenai pengalaman responden terhadap tindakan *cyberbullying* terhadap Samuel alexander pieter dapat dilihat pada gambar 3.38.



**Gambar 3.38** Hasil Kuesioner Pertanyaan Nomor Dua Puluh Empat (24)

Hasil kuesioner pertanyaan nomor dua puluh empat mengenai pengalaman responden terhadap tindakan *cyberbullying* terhadap Karin novilda dapat dilihat pada gambar 3.39.



**Gambar 3.39** Hasil Kuesioner Pertanyaan Nomor Dua Puluh Lima (25)

Data hasil kuesioner yang terkumpul kemudian akan diolah menggunakan *IBM SPSS stastistic 25* untuk melakukan rekap serta perhitungan pada tiap jawaban. Hasil dari pengolahan data kuesioner akan digunakan untuk mendukung penelitian.

## 2. Studi Pustaka

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

### 3.1.2. Metodologi Pengembangan Sistem

Dalam tahapan ini metode yang akan digunakan adalah metode GRAPPLE (*Guidelines for Rapid APPLication Engineering*). Metode GRAPPLE merupakan metode pengembangan perangkat lunak yang fleksibel dan memberikan panduan yang jelas dalam pengembangan perangkat lunak. Metode ini terdiri dari lima bagian:

#### 1. *Requirement Gathering*

Tahap pertama yang dilakukan oleh pengembang perangkat lunak adalah mengambil informasi lengkap dari pengguna tentang sistem yang akan dibangun dengan kuesioner. Kuesioner dilakukan secara *online* dengan responden yang berhubungan dengan sistem. Analisis masalah, fungsi dan kebutuhan sistem termasuk dalam tahap *Requirement gathering*.

#### 2. *Analysis*

Di tahap analisis yang dilakukan adalah menggali lebih dalam hasil dari tahap sebelumnya. Tahap ini mengkaji permasalahan pengguna dan melakukan analisis solusinya. Yang termasuk dalam tahapan ini adalah pengembangan data dan informasi dari *requirement gathering* serta pembuatan diagram awal.

#### 3. *Design*

Tahap *design* dilakukan untuk merancang solusi yang dihasilkan pada tahap analisis dan *design* dapat berjalan dua arah saling menyesuaikan hingga memperoleh

rancangan yang sesuai. Yang termasuk dalam tahapan ini adalah implementasi model dan diagram yang telah dianalisis dan dibuat rancangannya.

#### 4. *Development*

Tahap ini ditangani oleh pengembang program untuk membangun kode program dan *user interface*. Pengujian program dan dokumentasi sistem dilakukan pada tahap ini.

#### 5. *Deployment*

Tahap *deployment* adalah tahap pendistribusian produk kepada pengguna. Tahap ini mencakup instalasi dan perencanaan *backup* data bila diminta oleh pengguna sesuai dengan perjanjian sebelumnya.

### **3.2. Analisis kebutuhan Sistem**

Pada penelitian ini melibatkan beberapa kebutuhan perangkat lunak dan perangkat keras. Perangkat keras yang digunakan dalam pembuatan model dan pengkodean disajikan dalam bentuk tabel 3.1.

**Tabel 3.1 Spesifikasi Kebutuhan Perangkat Keras Pembuatan Model**

No	Spesifikasi	Detail Spesifikasi
1	Laptop	Asus X441U.
	Processor	Intel Core i3-6006U,2.0 GHz
	RAM	12 GB
	VGA card	NVIDIA GeForce 920Mx.
	Hard-disk	512 GB
2	Koneksi Internet	-

Perangkat Lunak (*software*) yang digunakan untuk mendukung pembuatan model terdiri dari beberapa komponen, komponen tersebut disajikan dalam tabel 3.2.

**Tabel 3.2 Spesifikasi Kebutuhan Perangkat Lunak Pembuatan Model**

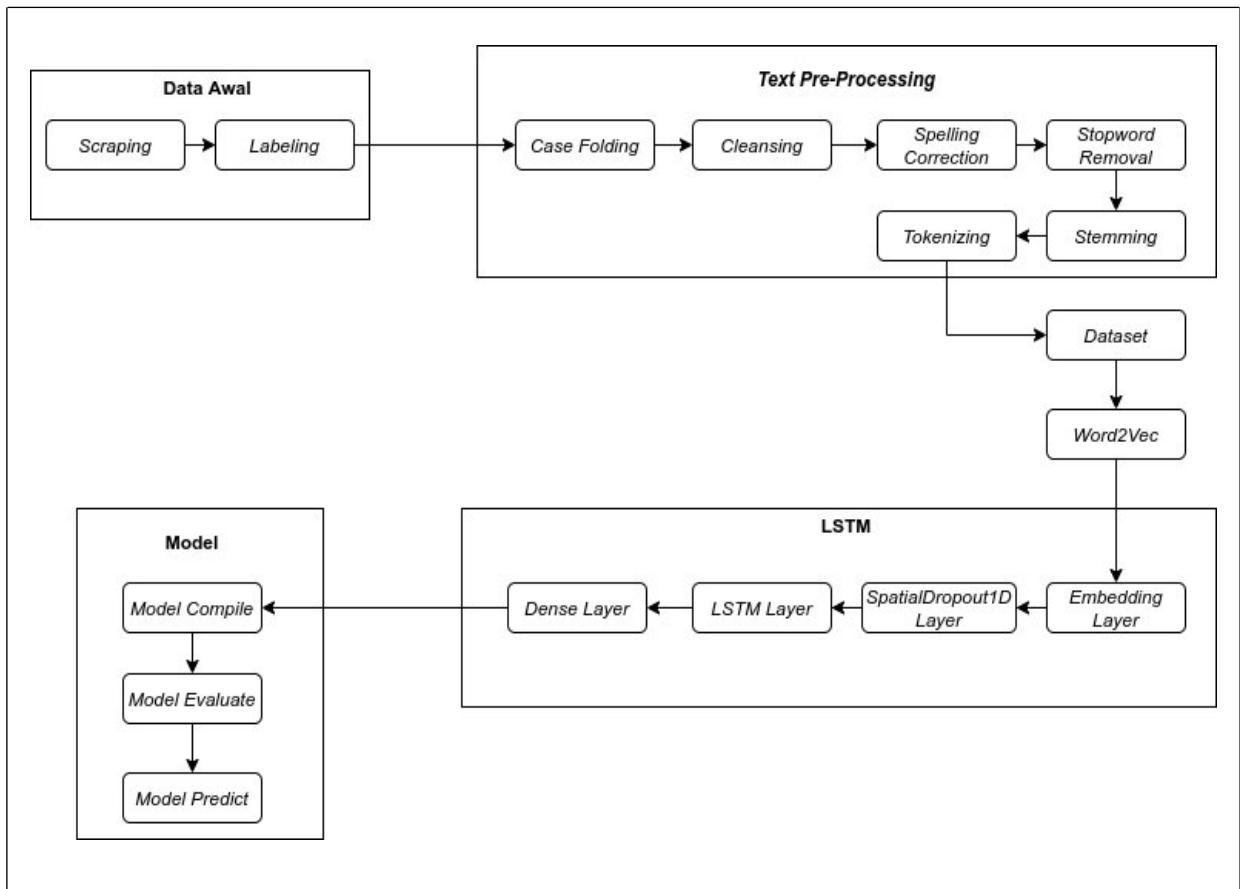
No	Spesifikasi	Detail Spesifikasi
1	Sistem Operasi	Windows 10
2	Bahasa Pemrograman	Python
3	<i>Library Backend</i>	Keras dan Tensorflow
	<i>Library Word2vec</i>	Gensim

### **3.3. Analisis**

Tahapan ini akan membahas mengenai analisis pembuatan model dan analisis pengujian model. analisis ini dijelaskan sebagai berikut:

### 3.3.1. Analisis Pembuatan Model LSTM

Analisis pembuatan model LSTM terdiri dari dua tahapan yaitu pengolahan data awal serta analisis jaringan LSTM. Tahapan analisis jaringan LSTM dijelaskan pada gambar 3.40.



**Gambar 3.40** Analisis Pembuatan Model LSTM

Alur dari pembuatan model LSTM yang dibangun dijelaskan sebagai berikut:

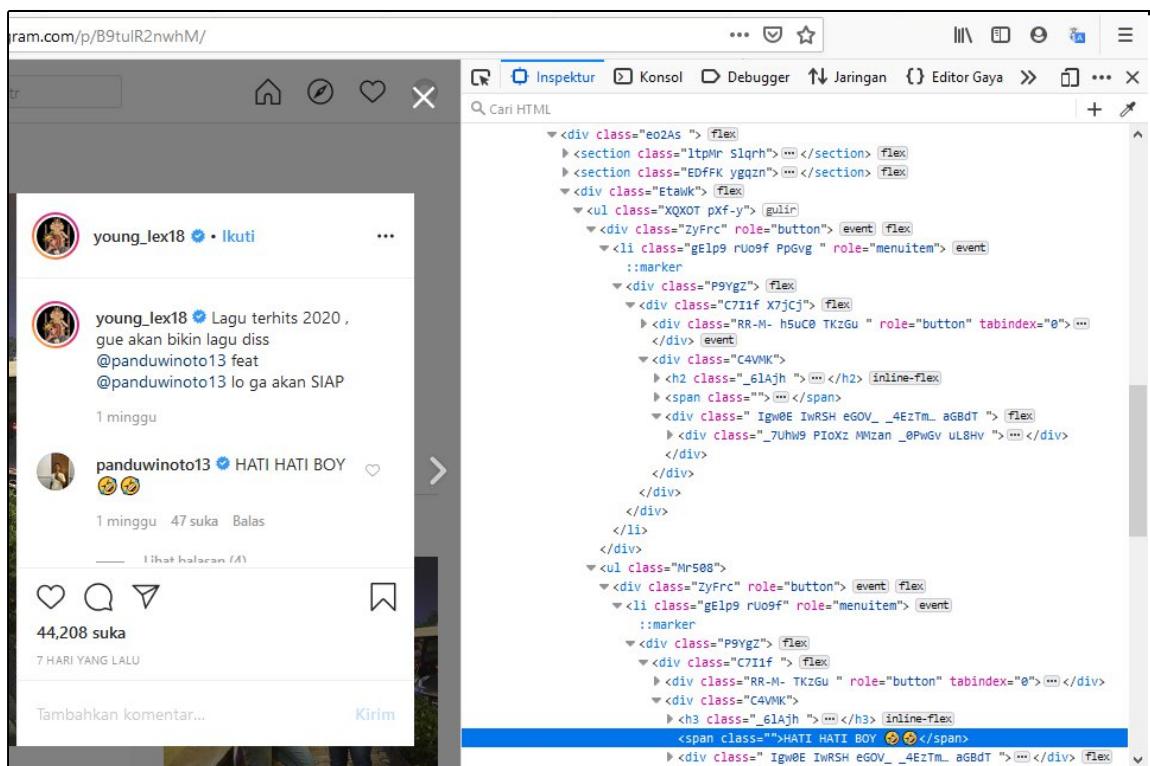
- Pengolahan Data Awal

Pengolahan data awal bertujuan untuk mempersiapkan *dataset* yang akan digunakan dalam proses pembentukan model. Pengolahan ini dilakukan terhadap data yang diperoleh melalui metode *scraping* dari beberapa sumber berdasarkan kuesioner yang telah dilakukan.

Beberapa tahap dalam pengolahan data awal meliputi:

### a) Scraping

Pada proses *scraping* ini, akan dilakukan pengambilan data berupa komentar dari beberapa kiriman *influencer* yang disesuaikan dengan hasil kuesioner yang telah dilakukan. Pengambilan data menggunakan *library selenium* untuk proses automasi. Proses pengambilan data berdasarkan *xpath* dari struktur *html* yang didapatkan dari *web browser* seperti pada gambar 3.41.



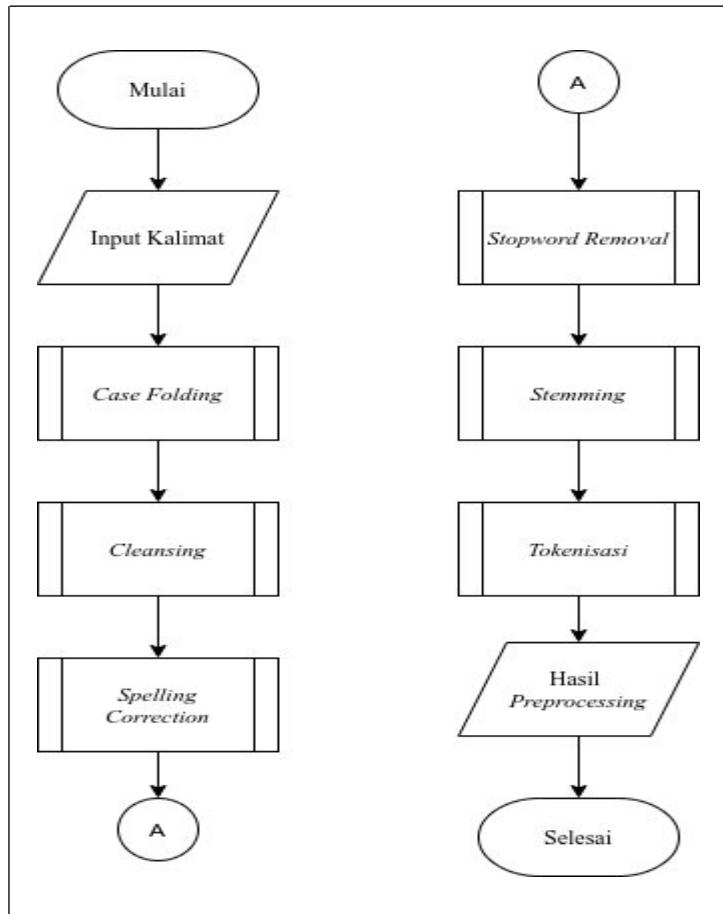
Gambar 3.41 Struktur tag HTML instagram

Data yang telah ditemukan kemudian akan disimpan ke dalam *database* untuk kemudian diteruskan ke proses *pre-processing* untuk mendapatkan *dataset* yang akan digunakan dalam pembentukan model.

### b) Text Pre-processing

Tahap selanjutnya dalam pengolahan data awal adalah *text pre-processing*. Tahapan ini dilakukan dengan tujuan untuk mendapatkan data yang terbebas dari *noise* sehingga tidak mengganggu dalam pembentukan model. Proses *text pre-processing* ini terdiri dari beberapa tahapan seperti *case folding*, *cleansing*, *spelling correction*, *stopword removal*, *stemming*

dan tokenizing. Tahapan *pre-processing* ini disesuaikan dengan data yang didapatkan dari proses *scraping*. Secara umum, tahapan *text pre-processing* disajikan dalam gambar 3.42.



**Gambar 3.42 Flowchart Text Preprocessing**

Tahapan pertama dalam *pre-processing* pada penelitian ini adalah *case folding*. Proses ini merupakan proses perubahan huruf kapital menjadi huruf kecil. Dilanjutkan tahapan *cleansing*, pada tahapan ini terjadi mekanisme seperti penghapusan *whitespace*, penghapusan *newline*, penghapusan *url* dan *username* serta karakter non alphabet. Contoh kalimat yang dilakukan proses *cleansing* di contohkan pada tabel 3.3.

**Tabel 3.3 Cleansing**

Kalimat awal	Hasil Cleansing
gw puas dgn penampilan lo bang. makasih udah jalan2 ke makassar. di tunggu lain waktu bang. @young lex18	gw puas dgn penampilan lo bang makasih udah jalan ke makassar di tunggu lain waktu bang

Kemudian proses selanjutnya adalah *spelling correction*. Proses ini dilakukan dengan tujuan memperbaiki kesalahan penulisan kata, menyamakan kata bermakna sama

sehingga mendapatkan data yang lebih mudah dipahami oleh sistem. Contoh proses yang mengalami *spelling correction* ditunjukkan pada tabel 3.4.

**Tabel 3.4 Spelling Correction**

Kalimat awal	Hasil Spelling Correction
gw puas dgn penampilan lo bang makasih udah jalan ke makassar di tunggu lain waktu bang	saya puas dengan penampilan kamu abang terima kasih sudah jalan ke makassar di tunggu lain waktu abang

Proses selanjutnya adalah *stopword removal*, pada tahap ini kata yang tidak memiliki makna akan dihapus. Kata seperti ke, di, dari dan kata hubung lainnya akan dihilangkan dari data. Contoh proses ini dapat dilihat pada tabel 3.5.

**Tabel 3.5 Stopword Removal**

Kalimat awal	Hasil stopword removal
saya puas dengan penampilan kamu abang Terima kasih sudah jalan ke makassar di tunggu lain waktu abang	saya puas penampilan abang terima kasih jalan makassar tunggu waktu abang

Proses terakhir dari *pre-processing* ini adalah proses *stemming*. Proses ini merupakan proses perubahan sebuah kata ke dalam kata dasar sesuai dengan kamus yang digunakan. Proses ini dapat dilihat pada tabel 3.6.

**Tabel 3.6 Stemming**

Kalimat awal	Hasil stopword removal
saya puas penampilan abang terima kasih jalan makassar tunggu waktu abang	saya puas tampil abang terima kasih jalan makassar tunggu waktu abang

Kemudian setelah data tersebut selesai melalui tahapan *stemming*, kemudian akan dilanjutkan proses *tokenizing* yaitu proses pemecahan kata dalam dokumen, proses ini bertujuan untuk membentuk *vocabulary* yang akan digunakan dalam mekanisme *one hot encoding* atau perubahan kalimat ke dalam urutan angka.

c) Pembentukan *Vocabulary*

*Vocabulary* merupakan sebuah kamus yang berisi fitur-fitur yang ada di dalam *dataset*. *Dataset* yang telah mengalami proses *pre-processing* kemudian akan dilakukan tokenisasi atau proses pemecahan kalimat menjadi kata, dalam pembentukan *vocabulary* kata-kata hasil tokenisasi tersebut kemudian di *group* sesuai kata dan akan diurutkan

berdasarkan kemunculannya. Proses ini akan disajikan contoh dengan menggunakan dataset pada tabel 3.7.

**Tabel 3.7** Contoh Dataset

No	Kalimat
1	Aku suka dengan penampilanmu
2	Penampilanmu sangat keren
3	Aku kurang suka dengan sikapmu
4	Sikapmu seperti binatang
5	Aku sangat suka dengan binatang
6	Penampilanmu terlihat biasa saja
7	Penampilanmu tidak spesial
8	Kelakuanmu sungguh buruk
9	Bagaimana kabarmu
10	Karyamu sangat bagus

Setiap kalimat pada tabel 3.7 akan dilakukan *split* (pemecahan) menjadi kata perkata. Kemudian kata tersebut akan di kelompokan sesuai dengan kata yang sama, jumlah kata dengan frekuensi tertinggi akan mendapatkan indeks yang kecil. Hasil pengelompokan kata dapat dilihat pada tabel 3.8.

**Tabel 3.8** Frekuensi Kemunculan Kata

No	Kata	Kemunculan	Indeks
1	Aku	3	2
2	Suka	3	3
3	Dengan	3	5
4	Penampilanmu	4	1
5	Sangat	3	4
6	Keren	1	8
7	Kurang	1	9
8	Sikapmu	2	6
9	Seperti	1	10
10	Binatang	2	7
11	Terlihat	1	11
12	Biasa	1	12
13	Saja	1	13
14	Tidak	1	14
15	Spesial	1	15
16	Kelakuanmu	1	16
17	Sungguh	1	17
18	Buruk	1	18
19	Bagaimana	1	19
20	Kabarmu	1	20
21	Karyamu	1	21
22	Bagus	1	22

d) *Text to Sequence*

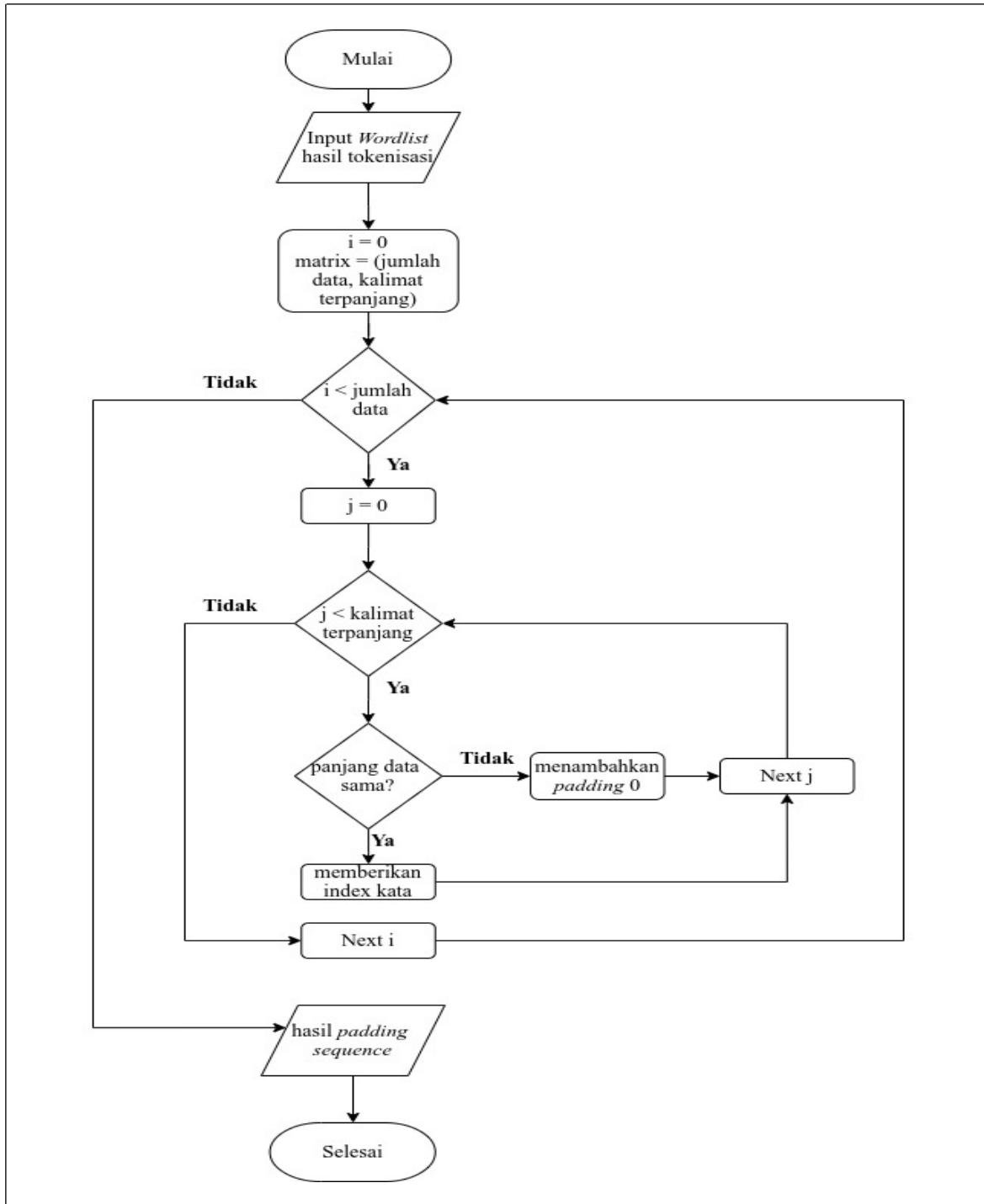
Pada mekanisme ini, setiap kalimat pada tabel 3.7 akan diubah sesuai dengan indeks kata penyusun kalimat sesuai dengan hasil pada tabel 3.8. Perubahan ini dilakukan karena algoritma yang digunakan tidak dapat melakukan pemrosesan terhadap teks secara langsung, sehingga teks tersebut harus diubah menjadi sebuah angka. Hasil dari proses ini dapat dilihat pada tabel 3.9.

**Tabel 3.9** *Text to Sequence*

No	Kalimat	Sequence
1	Aku suka dengan penampilanmu	2 3 5 1
2	Penampilanmu sangat keren	1 4 8
3	Aku kurang suka dengan sikapmu	2 9 3 5 6
4	Sikapmu seperti binatang	6 10 7
5	Aku sangat suka dengan binatang	2 4 3 7
6	Penampilanmu terlihat biasa saja	1 11 12 13
7	Penampilanmu tidak spesial	1 14 15
8	Kelakuanmu sungguh buruk	16 17 18
9	Bagaimana kabarmu	19 20
10	Karyamu sangat bagus	21 4 22

e) *Padding Sequence*

Proses selanjutnya merupakan proses menyamakan ukuran data. Pada tabel 3.9 *sequence* yang terbentuk memiliki ukuran yang berbeda, sehingga diperlukan penambahan *padding* (nilai 0) terhadap *sequence* yang ukurannya kurang dari ukuran *sequence* terpanjang. *Flowchart* proses ini digambarkan pada gambar 3.43.



Gambar 3.43 Flowchart Padding Sequence

Melalui mekanisme tersebut, setiap data akan memiliki representasi *sequence* dengan panjang yang sama. Hasil dari proses ini dapat dilihat pada tabel 3.10.

**Tabel 3.10 Padding Sequence**

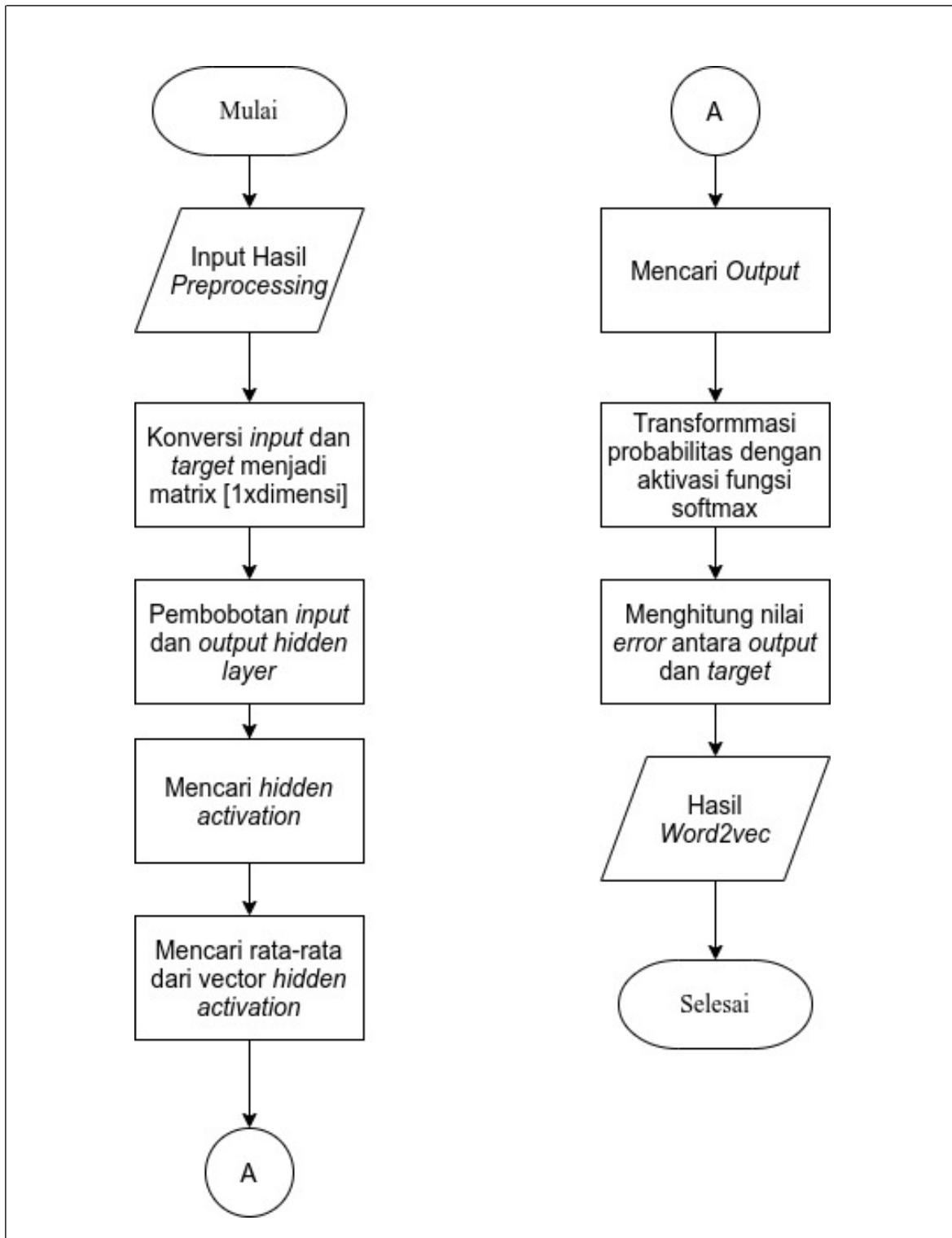
No	Sequence	Padding Sequence
1	2 3 5 1	[ 0 2 3 5 1 ]
2	1 4 8	[ 0 0 1 4 8 ]
3	2 9 3 5 6	[ 2 9 3 5 6 ]
4	6 10 7	[ 0 0 6 10 7 ]
5	2 4 3 7	[ 0 2 4 3 7 ]
6	1 11 12 13	[ 0 1 11 12 13 ]
7	1 14 15	[ 0 0 1 14 15 ]
8	16 17 18	[ 0 0 16 17 18 ]
9	19 20	[ 0 0 0 19 20 ]
10	21 4 22	[ 0 0 21 4 22 ]

## 2. Analisis jaringan LSTM

Pembentukan jaringan LSTM disesuaikan dengan *layer* yang digunakan. Dalam penelitian ini layer yang digunakan adalah *embedding layer*, *LSTM layer* dan *dense layer*. Dalam pembentukan jaringan ini juga digunakan *pre-trained word2vec* sebagai *input* matriks pada *layer embedding*. Proses pembentukan pada jaringan ini dijelaskan sebagai berikut:

### a) Metode Word2vec

*Pre-trained word2vec* digunakan untuk memberikan representasi nilai vector pada tiap katanya. *Pre-trained Word2vec* yang digunakan dapat berasal dari *pretrained* yang sudah disediakan seperti *pre-trained word2vec* wikipedia yang berasal dari ratusan ribu artikel ataupun dapat menggunakan *pre-trained word2vec* yang dibuat berdasarkan data tertentu. Pada penelitian ini digunakan *pre-trained word2vec* yang bersumber dari korpus Wikipedia Bahasa Indonesia dengan menambahkan data hasil *pre-processing* pada pengolahan data awal seta melakukan *training* untuk mendapatkan representasi kata untuk kata yang belum terdapat dalam korpus. Dalam penjelasan ini, data yang digunakan sebagai contoh sama dengan data pada tabel 3.4. Model *word2vec* yang digunakan dalam penelitian ini adalah model jenis CBOW (*Continuous Bag-of-Word Flowchart* pada proses *word2vec* ini dapat dilihat pada gambar 3.44.



Gambar 3.44 Flowchart Word2vec

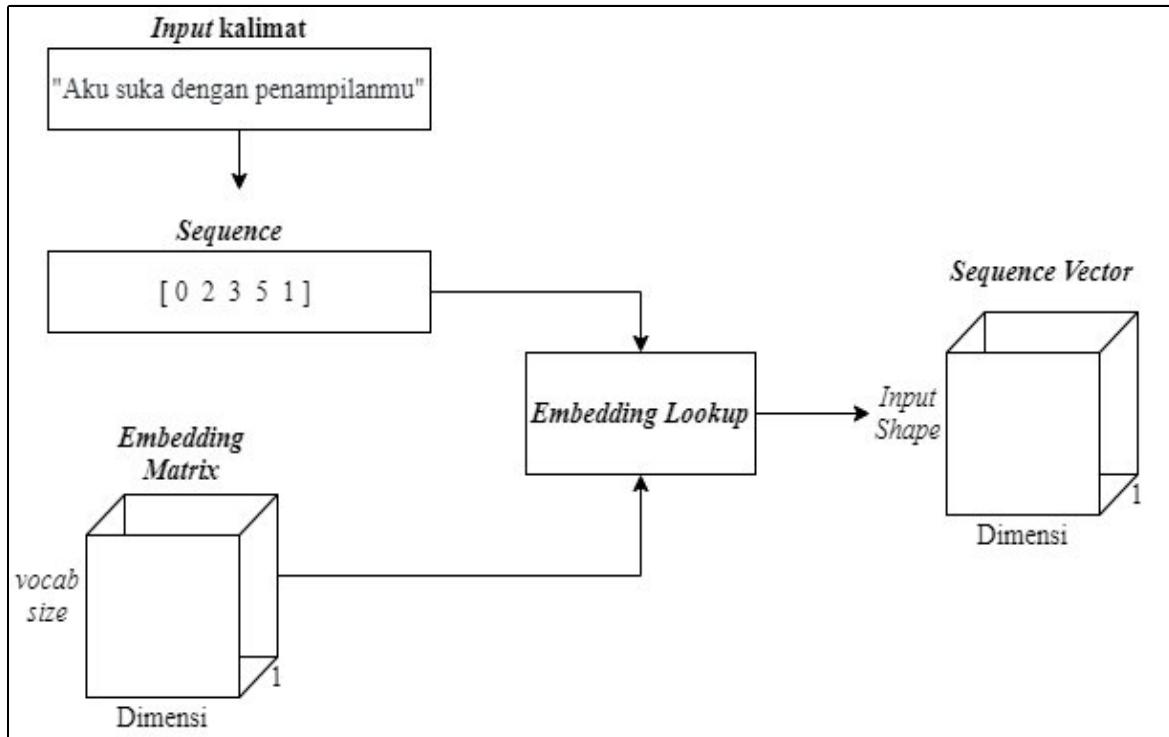
Pada contoh ini, *word2vec* di contohkan dengan vector berdimensi lima. Hasil dari metode *word2vec* ini dapat dilihat pada tabel 3.11.

**Tabel 3.11** Hasil Representasi Vector *Word2Vec*

No	Kata	index	Beban				
			D1	D2	D3	D4	D5
1	<i>Padding</i>	0	0.04158386	0.02959001	-0.04963555	-0.01363942	-0.01900562
2	Penampilanmu	1	-0.05022407	0.01277214	-0.04014155	0.04411147	-0.00672836
3	Aku	2	-0.00354392	0.00176554	0.02657353	-0.02857587	0.03202229
4	Suka	3	-0.03969593	-0.02725925	0.01367543	-0.03058508	-0.04125314
5	Sangat	4	-0.01069643	0.00363313	0.01711509	-0.02284402	0.0267443
6	Dengan	5	0.00069527	0.00546499	0.02828673	0.00658707	-0.03970267
7	Sikapmu	6	-0.03380138	0.00028137	-0.03372325	0.02593099	-0.04310613
8	Binatang	7	-0.01546003	-0.00026823	-0.00263986	-0.0474107	0.01112957
9	Keren	8	0.03178306	0.02756589	0.0210265	0.03450796	0.0197009
10	Kurang	9	0.04166121	0.04887357	-0.00999688	0.02956016	-0.02305247
11	Seperti	10	-0.01409278	0.01992387	-0.02857943	-0.00481863	0.04196896
12	Terlihat	11	0.02798622	0.04240387	-0.01714536	0.03413061	-0.02245562
13	Biasa	12	-0.04453019	-0.01765126	0.01433415	-0.01333065	0.01576451
14	Saja	13	-0.01859797	0.02609836	0.00099853	-0.01058364	0.00520941
15	Tidak	14	-0.0064388	0.0110356	0.04434614	-0.0372705	-0.00396214
16	Spesial	15	-0.04453517	-0.03201138	-0.04232472	0.00244243	-0.04300811
17	Kelakuanmu	16	-0.00926455	-0.0224384	-0.00912403	0.02328083	0.03828733
18	Sungguh	17	0.04285464	0.00909968	-0.01258817	-0.00379785	0.00726926
19	Buruk	18	-0.0210852	-0.01717311	0.02353896	-0.02285265	-0.04629857
20	Bagaimana	19	-0.03200927	0.03136183	0.03540223	-0.04250291	0.0417359
21	Kabarmu	20	0.03681691	-0.03406873	0.0090282	-0.023514	-0.01669268
22	Karyamu	21	-0.02369635	-0.01389248	0.01399858	-0.01260429	0.0297589
23	Bagus	22	-0.03864509	0.03280069	-0.02247358	0.03627903	-0.03464929

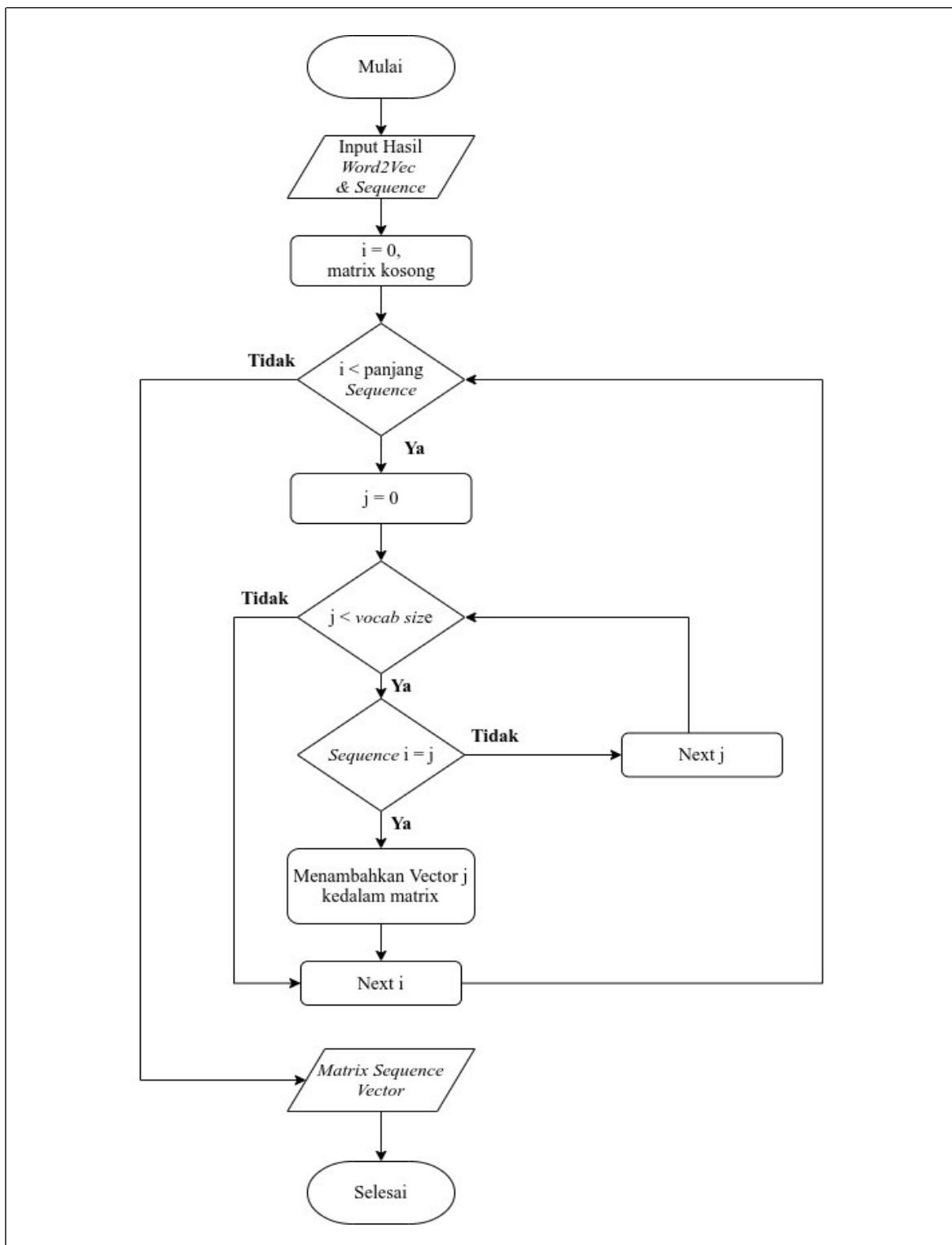
### b) *Embedding Layer*

Pada layer ini terjadi proses perubahan *sequence* menjadi vector. Representasi vector yang digunakan berasal dari metode *word2vec* yang telah dijalankan sebelumnya sesuai dengan tabel 3.11. *Sequence* yang digunakan sebagai masukan akan diubah sesuai indeks yang terdapat pada *sequence* tersebut sehingga masukan yang semula *sequence* kemudian akan berubah menjadi matriks. Representasi matriks yang terbentuk akan sesuai dengan dimensi masukkan serta dimensi keluaran *pretrained word2vec*. Proses ini di ilustrasikan pada gambar 3.45.



Gambar 3.45 Ilustrasi Representasi Vector

Berdasarkan ilustrasi pada gambar 3.45, proses awal dalam mengubah kalimat untuk menjadi *input* layer LSTM dimulai dengan melakukan perubahan kalimat menjadi angka berdasarkan indeks pada *vocabulary* yang terbentuk sesuai dengan tabel 3.8, angka tersebut kemudian akan membentuk sebuah *sequence* angka yang kemudian akan dilakukan *padding* atau proses penyamaan ukuran *sequence* sehingga setiap kalimat dalam data memiliki ukuran panjang yang sama. Angka pada tiap *sequence* tersebut kemudian satu persatu akan dilakukan penyesuaian sesuai dengan matriks *pretrained* yang terbentuk, setiap angka dalam *sequence* merepresentasikan posisi fitur dalam *embedding* matriks, perubahan angka menjadi vector matriks ditunjukan dengan proses *embedding lookup*, hasil yang terbentuk adalah sebuah matriks dengan ukuran panjang sesuai dengan panjang *sequence* dan dimensi sesuai dengan dimensi *pretrained embedding*. Flowchart untuk proses ini dapat dilihat pada gambar 3.46.



Gambar 3.46 Flowchart Representasi Sequence

Hasil representasi pada proses ini dicontohkan berdasarkan *input* yang ditunjukkan pada tabel 3.12.

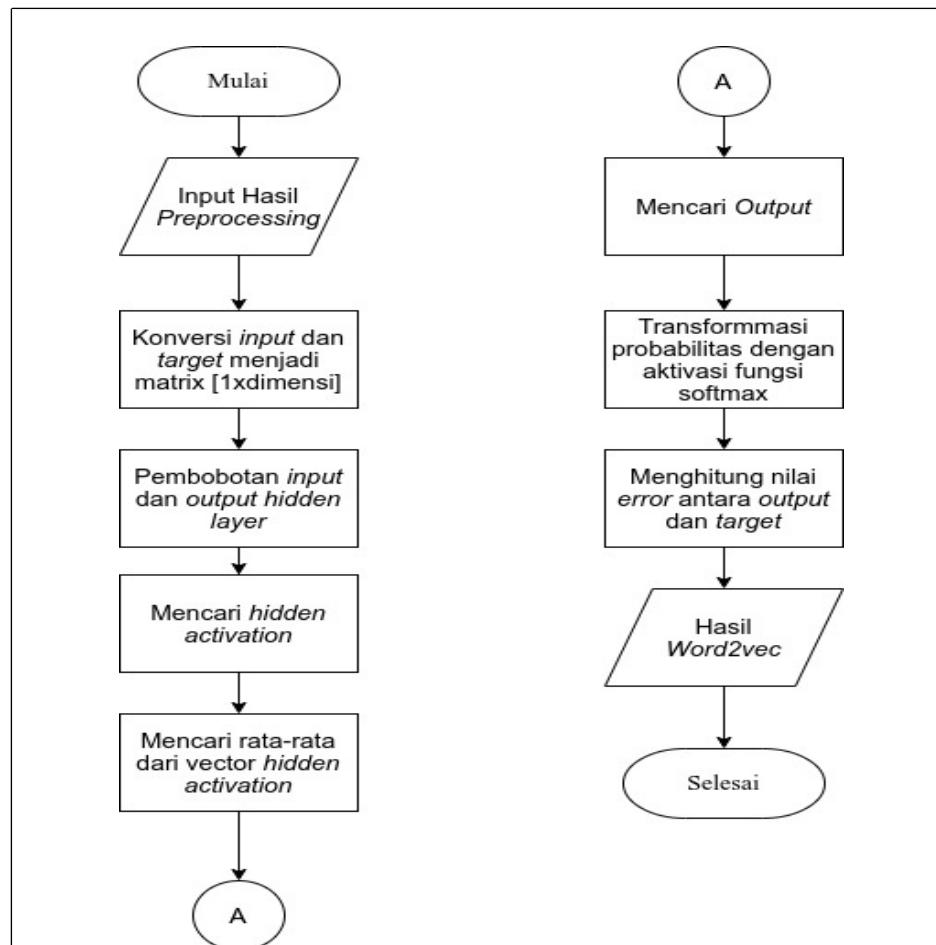
**Tabel 3.12** Hasil Representasi Sequence Vector

<b>Indeks Sequence</b>	<b>Representasi Sequence Vector</b>				
0	0.04158386	0.02959001	-0.04963555	-0.01363942	-0.01900562
2	-0.00354392	0.00176554	0.02657353	-0.02857587	0.03202229
3	-0.03969593	-0.02725925	0.01367543	-0.03058508	-0.04125314
5	0.00069527	0.00546499	0.02828673	0.00658707	-0.03970267
1	-0.05022407	0.01277214	-0.04014155	0.04411147	-0.00672836

Representasi vector tersebut akan digunakan dalam proses pembentukan model sehingga *sequence* yang awalnya berbentuk urutan angka akan berganti menjadi matriks dengan isi sesuai hasil representasi vector.

### c) LSTM Layer

Berdasarkan gambar 2.1, struktur jaringan LSTM terdiri dari beberapa *gate* serta fungsi lainnya. *Gate* pada mekanisme ini di antaranya adalah *forget gate*, *input gate* dan *output gate* serta sebuah kandidat konteks. Kinerja layer ini secara keseluruhan dapat dijelaskan dalam *flowchart* pada gambar 3.47.

**Gambar 3.47** Flowchart LSTM Layer

Untuk menjelaskan *flowchart* pada gambar 3.47 , digunakan nilai *input* pada tabel 3.12 dengan jumlah *hidden unit* sebesar lima. *Hidden unit* digunakan dalam pembentukan matriks pada bobot tiap gerbang ( $W$ ), *recurrent weight* ( $U$ ) serta bias ( $b$ ).

Tahapan pertama merupakan pencarian nilai dari *forget gate* ( $f_t$ ), proses perhitungan ini berdasarkan persamaan 2.1. Terdapat beberapa *variable* yang digunakan dalam perhitungan ini di antaranya *weight forget gate* ( $W_f$ ), *recurrent weight forget gate* ( $U_f$ ), dan *bias forget gate* ( $b_f$ ). *Shape* pada tiap variabel disesuaikan dengan *input dimension* serta *hidden unit* dengan nilai pembentukan secara *random*. contoh bilai variabel pada saat t=0 sebagai berikut:

$$S_{t-1} = [0 \ 0 \ 0 \ 0 \ 0]$$

$$x_t = [0.04158386 \ 0.02959001 \ -0.04963555 \ -0.01363942 \ -0.01900562]$$

$$W_f = \begin{bmatrix} -0.06679793 & 0.07696377 & -0.1955175 & 0.17297111 & -0.21290807 \\ 0.3353057 & 0.21361546 & -0.3104262 & -0.30179897 & -0.20790976 \\ 0.42183167 & -0.28461212 & -0.26954448 & -0.18463889 & -0.27634445 \\ -0.07616175 & -0.13537449 & -0.1974054 & -0.33538216 & 0.13820167 \\ -0.05800783 & -0.36084044 & 0.1130465 & -0.21635813 & 0.1401893 \end{bmatrix}$$

$$U_f = \begin{bmatrix} -0.4968713 & -0.25469545 & -0.07402467 & 0.01189694 & -0.11940969 \\ -0.4944626 & -0.09840754 & -0.04553243 & 0.13584866 & 0.12456008 \\ 0.16532187 & -0.21747749 & -0.16607277 & -0.34502667 & -0.15585357 \\ -0.18400517 & -0.01344236 & 0.18728717 & -0.28626925 & -0.4790093 \\ -0.11064873 & 0.32629865 & 0.34885493 & -0.37731117 & 0.03295562 \end{bmatrix}$$

$$b_f = [1.0006763 \ 1.0009696 \ 1.0009884 \ 1.0009965 \ 0.99900699]$$

Dalam perkalian matriks, syarat dua buah matriks dapat dikalikan adalah jumlah kolom pada matriks A sama dengan jumlah baris pada matriks B sehingga hasil akhirnya adalah matriks C dengan ordo baris matriks A x kolom matriks B (  $A_{mxn} + B_{nxs} = C_{mxs}$  ). Berdasarkan syarat tersebut, maka hasil perkalian berdasarkan persamaan 2.1 dijelaskan sebagai berikut:

$$x_t * W_f =$$

$$[-0.01165259 \ 0.03235264 \ -0.00339292 \ 0.01611366 \ -0.00583846]$$

$$S_{t-1} * U_f = [0 \ 0 \ 0 \ 0 \ 0]$$

Nilai-nilai tersebut kemudian akan dijumlahkan dengan nilai *bias* ( $b_f$ ), syarat penjumlahan dalam matriks adalah jumlah baris dan kolom tiap matriks harus sama. Sehingga hasil penjumlahan pada tahap ini adalah:

$$\begin{aligned} & x_t W_f + S_{t-1} U_f + b_f \\ & = [0.98902371 \quad 1.03332224 \quad 0.99759548 \quad 1.01711016 \quad 0.99316853] \end{aligned}$$

Nilai tersebut kemudian akan menjadi *input* pada fungsi sigmoid sesuai dengan persamaan 2.7, *output* dari fungsi ini adalah vector dengan nilai antara 1 hingga 0.

$$\begin{aligned} f_t &= \sigma(x_t W_f + S_{t-1} U_f + b_f) \\ f_t &= \sigma([0.98902371 \quad 1.03332224 \quad 0.99759548 \quad 1.01711016 \quad 0.99316853]) \\ f_t &= [0.72889504 \quad 0.73755947 \quad 0.73058556 \quad 0.73440931 \quad 0.72971331] \end{aligned}$$

Langkah selanjutnya adalah mencari nilai dari *input gate* ( $i_t$ ). Pada contoh perhitungan ini variabel yang digunakan dalam perhitungan dituliskan sebagai berikut:

$$\begin{aligned} W_i &= \begin{bmatrix} -0.1366528 & -0.1453298 & 0.14726354 & 0.04116447 & -0.22698675 \\ 0.28899997 & -0.334841 & -0.20672792 & -0.28585976 & 0.27622768 \\ 0.32864657 & -0.20383239 & 0.27974045 & -0.3035583 & 0.34484547 \\ -0.13876237 & 0.43230888 & 0.15773965 & -0.25042242 & -0.18264823 \\ -0.42313895 & -0.4679739 & 0.39361656 & 0.13375297 & -0.09634234 \end{bmatrix} \\ U_i &= \begin{bmatrix} -0.2499653 & -0.0829422 & 0.02788328 & -0.11135205 & -0.16843288 \\ -0.01982183 & 0.1994739 & -0.00208896 & -0.3418544 & 0.10404024 \\ -0.12342916 & -0.1927847 & 0.20666523 & 0.08170953 & 0.05257188 \\ 0.15766141 & 0.11748825 & 0.27381027 & -0.03372682 & -0.298041 \\ 0.18291861 & 0.05837192 & 0.02188385 & 0.01012534 & 0.04436669 \end{bmatrix} \\ b_i &= \\ & [0.00098822475 \quad -0.0009966509 \quad 0.000994517 \quad 0.00099469442 \quad 0.00099272723] \end{aligned}$$

Proses perhitungan pada *gate* ini sama dengan proses perhitungan pada *gate* sebelumnya.

Proses perhitungan yang terjadi dijelaskan sebagai berikut:

$$\begin{aligned} x_t * W_i &= [-0.00350894 \quad -0.0028363 \quad -0.02351077 \quad 0.00919403 \quad -0.01405974] \\ S_{t-1} * U_i &= [0 \quad 0 \quad 0 \quad 0 \quad 0] \end{aligned}$$

Kemudian nilai tersebut dijumlahkan dengan menambahkan nilai *bias* ( $b_i$ ). Hasil dari penjumlahan ini adalah sebagai berikut:

$$\begin{aligned} & x_t W_i + S_{t-1} U_i + b_i \\ & = [-0.00252071 \quad -0.00383295 \quad -0.02251625 \quad 0.01018872 \quad -0.01306701] \end{aligned}$$

Kemudian nilai ini akan diproses dengan fungsi sigmoid sesuai persamaan 2.7. Hasil dari perhitungan ini dijelaskan sebagai berikut:

$$\begin{aligned} i_t &= \sigma(x_t W_i + S_{t-1} U_i + b_i) \\ i_t &= \sigma([-0.00252071 \quad -0.00383295 \quad -0.02251625 \quad 0.01018872 \quad -0.01306701]) \\ i_t &= [0.49936982 \quad 0.49904176 \quad 0.49437117 \quad 0.50254716 \quad 0.49673329] \end{aligned}$$

Langkah selanjutnya adalah mencari nilai dari kandidat konteks ( $\tilde{C}_t$ ). Pada contoh perhitungan ini variabel yang digunakan dalam perhitungan dituliskan sebagai berikut:

$$\begin{aligned} W_c &= \begin{bmatrix} 0.29096708 & -0.22230864 & -0.04721818 & -0.14204839 & 0.31201917 \\ 0.24634545 & -0.4805805 & -0.3990263 & 0.22731064 & -0.1041048 \\ -0.35916275 & 0.09782788 & 0.469568 & 0.11810622 & 0.38800818 \\ 0.29614687 & 0.04033977 & 0.20984042 & 0.04260591 & -0.18922086 \\ 0.05765662 & -0.43781337 & -0.28070438 & -0.44990382 & 0.1481089 \end{bmatrix} \\ U_c &= \begin{bmatrix} -0.12083001 & 0.39659318 & -0.0654858 & 0.04401578 & 0.26203072 \\ -0.22963278 & -0.189749 & 0.03406774 & -0.4484896 & 0.01709633 \\ -0.39516687 & 0.0017702 & 0.2092277 & 0.11332781 & -0.0490029 \\ 0.04684084 & -0.25821266 & 0.13414979 & -0.18060955 & -0.00232818 \\ 0.04671448 & 0.24460776 & -0.00362939 & -0.25552842 & -0.34277752 \end{bmatrix} \end{aligned}$$

$$\begin{aligned} b_c &= \\ &= [-0.0009999288 \quad -0.00099996722 \quad 0.00099993369 \quad 0.00099991076 \quad -0.00099989446] \end{aligned}$$

Proses perhitungan pada *gate* ini sama dengan proses perhitungan pada *gate* sebelumnya.

Proses perhitungan yang terjadi dijelaskan sebagai berikut:

$$\begin{aligned} x_t * W_c &= [0.03208107 \quad -0.02054987 \quad -0.03460511 \quad 0.00292652 \quad -0.00959854] \\ S_{t-1} * U_c &= [0 \quad 0 \quad 0 \quad 0 \quad 0] \end{aligned}$$

Kemudian nilai tersebut dijumlahkan dengan menambahkan nilai *bias* ( $b_c$ ). Hasil dari penjumlahan ini adalah sebagai berikut:

$$\begin{aligned} & x_t W_c + S_{t-1} U_c + b_c \\ & = [0.03108114 \quad -0.02154984 \quad -0.03360518 \quad 0.00392643 \quad -0.01059843] \end{aligned}$$

Kemudian nilai ini akan diproses dengan fungsi *tanh* sesuai persamaan 2.8. Hasil dari perhitungan ini dijelaskan sebagai berikut:

$$\begin{aligned} \tilde{C}_t &= \tanh(x_t W_c + S_{t-1} U_c + b_c) \\ \tilde{C}_t &= \tanh([0.03108114 \quad -0.02154984 \quad -0.03360518 \quad 0.00392643 \quad -0.01059843]) \\ \tilde{C}_t &= [0.03107113 \quad -0.0215465 \quad -0.03359254 \quad 0.00392641 \quad -0.01059804] \end{aligned}$$

Langkah selanjutnya adalah mencari nilai *cell state* ( $C_t$ ). Nilai *cell state* didapatkan dari hasil perkalian *forget gate* ( $f_t$ ) dengan *cell state* sebelumnya ( $C_{t-1}$ ) dijumlahkan dengan *input gate* ( $i_t$ ) dikalikan dengan nilai kandidat ( $\tilde{C}_t$ ). Berdasarkan persamaan 2.4, perhitungan ini dijelaskan sebagai berikut:

$$C_{t-1} = [0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Karena dalam pengujian ini menggunakan  $t=0$ , maka nilai *cell state* sebelumnya ( $C_{t-1}$ ) masih bernilai 0.

$$\begin{aligned} C_{t-1} * f_t &= [0 \quad 0 \quad 0 \quad 0 \quad 0] \\ i_t * \tilde{C}_t &= [0.01551599 \quad -0.0107526 \quad -0.01660718 \quad 0.00197321 \quad -0.0052644] \\ C_t &= C_{t-1} * f_t + i_t * \tilde{C}_t \\ C_t &= [0.01551599 \quad -0.0107526 \quad -0.01660718 \quad 0.00197321 \quad -0.0052644] \end{aligned}$$

Kemudian selanjutnya adalah mencari nilai *output gate* ( $o_t$ ). Berdasarkan persamaan 2.5. dalam menentukan nilai *output gate* diperlukan beberapa variabel sebagai berikut:

$$W_o = \begin{bmatrix} 0.019035183 & -0.34337175 & 0.04184407 & 0.18197168 & -0.398562 \\ 0.3628862 & -0.04002702 & -0.1929804 & -0.12946755 & 0.30658054 \\ -0.2654166 & 0.13074106 & -0.2759772 & 0.20468484 & -0.43865386 \\ -0.18691759 & -0.29926628 & 0.31149325 & -0.35591918 & -0.26953852 \\ -0.19566391 & 0.29882258 & 0.3616977 & 0.23587628 & 0.40475217 \end{bmatrix}$$

$$U_o = \begin{bmatrix} 0.27834892 & -0.07738667 & -0.30355817 & -0.24236387 & -0.27790406 \\ -0.07348764 & -0.06738382 & 0.3287794 & 0.3633701 & 0.00691418 \\ 0.09444071 & 0.42943755 & 0.02961049 & 0.442206 & -0.21621531 \\ -0.11019727 & -0.00283894 & -0.3779591 & 0.00822705 & 0.375038 \\ -0.15103883 & 0.00998165 & -0.00746319 & -0.06825706 & -0.5506425 \end{bmatrix}$$

$$b_o =$$

$$[0.00098884047 \quad -0.00099684286 \quad 0.00099455914 \quad 0.00099475321 \quad 0.00099280605]$$

Berdasarkan persamaan 2.5, perhitungan mengenai proses ini dijelaskan sebagai berikut:

$$x_t * W_o =$$

$$[0.03809563 \quad -0.02355002 \quad -0.00139485 \quad -0.00605195 \quad 0.01025458]$$

$$S_{t-1} * U_o = [0 \quad 0 \quad 0 \quad 0 \quad 0]$$

Kemudian nilai tersebut dijumlahkan dengan menambahkan nilai *bias* ( $b_o$ ). Hasil dari penjumlahan ini adalah sebagai berikut:

$$x_t W_o + S_{t-1} U_o + b_o =$$

$$[0.03908447 \quad -0.02454686 \quad -0.00040029 \quad -0.0050572 \quad 0.01124739]$$

Kemudian nilai ini akan diproses dengan fungsi *tanh* sesuai persamaan 2.8. Hasil dari perhitungan ini dijelaskan sebagai berikut:

$$O_t = \sigma(x_t W_o + S_{t-1} U_o + b_o)$$

$$O_t = \sigma([0.03908447 \quad -0.02454686 \quad -0.00040029 \quad -0.0050572 \quad 0.01124739])$$

$$O_t = [0.50976987 \quad 0.49386359 \quad 0.49989993 \quad 0.4987357 \quad 0.50281182]$$

Langkah terakhir dalam jaringan LSTM adalah menentukan *output* ( $S_t$ ). Nilai ini didapatkan dari nilai *output gate* ( $O_t$ ) dikalikan dengan nilai tanh dari *cell state* ( $C_t$ ). Hasil perhitungan dijelaskan sebagai berikut:

$$\tanh(C_t) =$$

$$[0.01551474 \quad -0.01075219 \quad -0.01660565 \quad 0.0019732 \quad -0.00526435]$$

$$S_t = O_t \tanh(C_t)$$

$$S_t = [0.00790895 \quad -0.00531012 \quad -0.00830117 \quad 0.00098411 \quad -0.00264698]$$

Nilai *output* ini akan digunakan dalam perhitungan pada saat t selanjutnya. Berikut merupakan hasil rekap perhitungan pada saat t satu sampai dengan lima berdasarkan *input* pada tabel 3.13.

**Tabel 3.13** Rekap hasil perhitungan  $C_t$

NO	Peroide t	Ct				
1	0	0.01551599	-0.0107526	-0.01660718	0.00197321	-0.0052644
2	1	0.00423049	-0.01294372	-0.01433707	-0.00271216	0.00694889
3	2	-0.01282369	0.01186155	0.00100534	0.00847993	0.00207198
4	3	-0.01697702	0.00083259	-0.00073215	0.00040396	0.00566259
5	4	-0.00447466	0.00164685	-0.00496093	0.00520149	-0.01881754

**Tabel 3.14** Rekap hasil perhitungan  $S_t$

NO	Peroide t	St				
1	0	0.00790895	-0.00531012	-0.00830117	0.00098411	-0.00264698
2	1	0.00211003	-0.0065294	-0.00713618	-0.0013669	0.00349756
3	2	-0.00639135	0.00595893	0.0004961	0.00422919	0.00103209
4	3	-0.00845363	0.00042044	-0.00036607	0.00020404	0.00285756
5	4	-0.00223387	0.00082168	-0.00250699	0.00255722	-0.00954338

**Tabel 3.15** Rekap hasil perhitungan  $C_{t-1}$

NO	Peroide t	Ct-1				
1	0	0	0	0	0	0
2	1	0.01551599	-0.0107526	-0.01660718	0.00197321	-0.0052644
3	2	0.00423049	-0.01294372	-0.01433707	-0.00271216	0.00694889
4	3	-0.01282369	0.01186155	0.00100534	0.00847993	0.00207198
5	4	-0.01697702	0.00083259	-0.00073215	0.00040396	0.00566259

**Tabel 3.16** Rekap hasil perhitungan  $S_{t-1}$

NO	Peroide t	St-1				
1	0	0	0	0	0	0
2	1	0.00790895	-0.00531012	-0.00830117	0.00098411	-0.00264698
3	2	0.00211003	-0.0065294	-0.00713618	-0.0013669	0.00349756
4	3	-0.00639135	0.00595893	0.0004961	0.00422919	0.00103209
5	4	-0.00845363	0.00042044	-0.00036607	0.00020404	0.00285756

#### d) Dense Layer

Dalam layer ini, terjadi proses perhitungan probabilitas sesuai dengan jumlah label.

*Output* dari layer LSTM ( $S_t$ ) pada periode terakhir akan digunakan sebagai *input* yang

kemudian akan dihitung dengan bobot pada *dense layer* serta bias pada *dense layer*. Hasil

pada perhitungan tersebut kemudian akan menjadi *input* pada fungsi softmax sesuai pada persamaan 2.9.

Pada layer ini bobot diinisiasi secara *random*, contoh bobot ( $W_y$ ) yang akan digunakan adalah sebagai berikut:

1.5251895306968255	0.03196444769188688	0.2625610645449784	- 2.11126763595446
- 0.11843542877489456	0.6394260325950156	- 0.3848575729419647	0.621739078512818
0.01736577202261132	1.1674852451129927	0.28808056082470995	0.7534497253823937
- 1.338263930739682	- 0.6936643320066209	2.1224731083231743	1.4131338406975564
0.27440716504223805	- 1.231741534740754	- 0.16453357812783548	- 0.09936133901927596

*Shape* yang terbentuk pada beban memiliki formula (*hidden state*, jumlah *unique label*), dalam percobaan ini *hidden state* sesuai pada layer LSTM yaitu lima dan jumlah *unique label* yang digunakan adalah empat sehingga terbentuk matriks  $W_{5 \times 4}$ . Kemudian untuk bias *shape* yang terbentuk sesuai dengan jumlah *unique label*, sehingga nilai bias ( $b_y$ ) adalah sebagai berikut:

$[-0.8472062988588057 \quad 1.2346212693109095 \quad 0.5332725256373476 \quad 1.8198945995558193]$

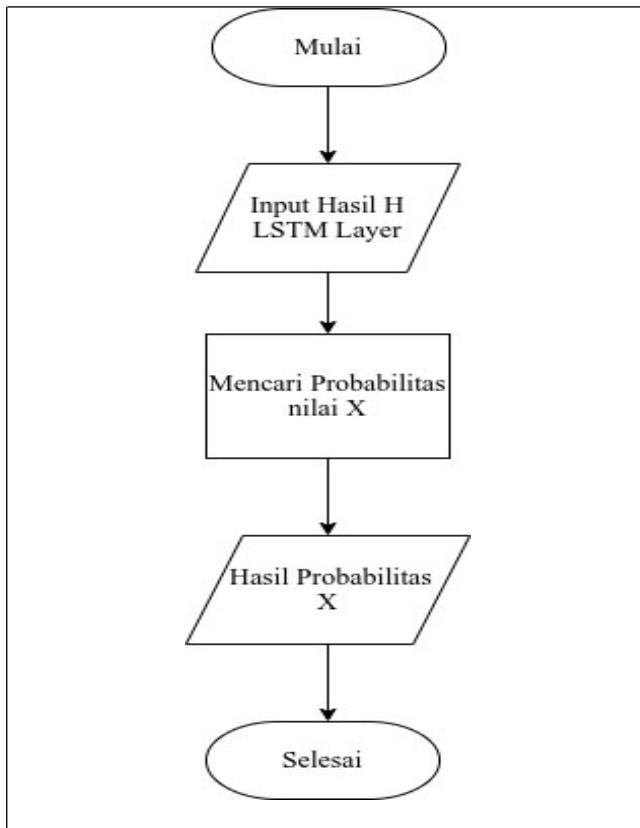
Proses perhitungan dijelaskan sebagai berikut:

$$S_t = [-0.00223387 \quad 0.00082168 \quad -0.00250699 \quad 0.00255722 \quad -0.00954338]$$

$$Y = \text{Softmax}(S_t W_y + b_y) \dots \quad (3.1)$$

$$Y = [0.29378845 \quad 0.40725321 \quad 0.05800649 \quad 0.24095185]$$

Berdasarkan hasil dari Y tersebut, probabilitas tertinggi berada pada kolom ke dua. Sehingga *input* pada tabel 3.8 memiliki hasil probabilitas tertinggi pada label ke dua. Proses ini kemudian akan berjalan terhadap semua *input* x pada dataset yang kemudian akan dilanjutkan dengan proses *training* dalam pembentukan model. Secara keseluruhan proses yang terjadi pada *dense layer* disajikan dalam gambar 3.48.



**Gambar 3.48 Flowchart Dense Layer**

### 3.3.2. Analisis Pengujian Model LSTM

Tahap analisis pengujian model LSTM membahas mengenai proses yang akan dijalankan dalam tahap pengujian melalui *web browser*. Tahapan tersebut dijelaskan sebagai berikut:

1. *Pre-processing*

Tahap *pre-processing* yang digunakan dalam pengujian model memiliki tahapan yang serupa dengan *pre-processing* pada pembentukan model. Tahapan ini terdiri dari beberapa tahapan seperti *case folding*, *punctuation removal*, *spelling correction*, *stemming* dan *tokenizing*.

2. *Load Tokenizer*

Pada tahapan ini data *tokenizer* yang sudah terbentuk dalam pembentukan model di *load* untuk pengujian model. Data yang berisi *vocabulary* dalam *dataset* tersebut akan digunakan dalam pengujian model pada tahapan *one hot encoding*.

### 3. *Text to Sequence*

Pada tahapan ini teks yang akan diuji diubah terlebih dahulu ke dalam sebuah *sequence* yang panjangnya sesuai dengan jumlah kalimat terpanjang dalam dataset, hal ini dikarenakan model LSTM tidak mampu membaca teks secara langsung sehingga teks tersebut harus terlebih dahulu diubah ke dalam bentuk *sequence*. Contoh pada tahapan ini dapat dilihat pada tabel 3.17.

**Tabel 3.17** *Text to Sequence*

#### 4. Prediksi Model

Pada proses ini, model yang telah terbentuk pada proses sebelumnya akan di-*load* untuk melakukan prediksi. Dengan fungsi *predict*, *sequence* yang sudah terbentuk sebelumnya akan dijadikan masukan fungsi tersebut sehingga akan menghasilkan probabilitas pada tiap kelas, probabilitas tertinggi pada hasil prediksi akan menjadi hasil klasifikasi. Berikut ini merupakan contoh hasil prediksi yang ditampilkan pada tabel 3.18.

**Tabel 3.18** Contoh Hasil Prediksi Model

Kalimat	Kelas	Probabilitas	Hasil Klasifikasi
terima kasih sudah jadi inspirasi saya mog sehat selalu	0	8.5452973e-07	<i>Cyberbullying</i>
	1	3.7949176e-06	<i>Irrelevant</i>
	2	4.6749008e-04	Netral
	3	<b>9.9952793e-01 (Tertinggi)</b>	<b><i>Non Cyberbullying</i></b>

### **3.4. Perancangan Sistem**

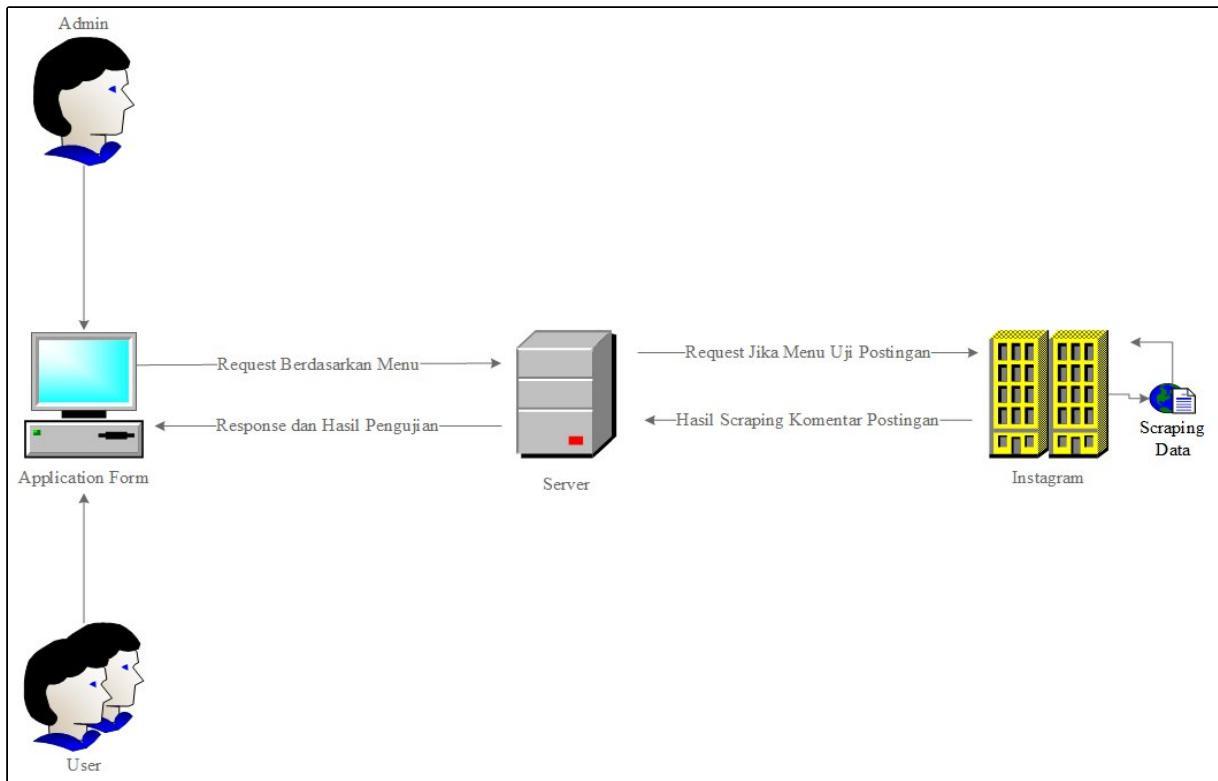
Perancangan sistem merupakan sebuah tahapan analisis untuk menjelaskan spesifikasi aliran proses yang akan dijalankan oleh sistem.

### **3.4.1. Perancangan Arsitektur Sistem Secara Keseluruhan**

Proses perancangan arsitektur sistem secara keseluruhan pada penelitian ini berupa *user* dan *admin* yang memiliki akses yang berbeda. Akses yang dimiliki oleh *user* ialah *user*

dapat melakukan uji kalimat serta melakukan identifikasi sebuah kiriman sesuai dengan *url* yang diisikan. Sedangkan *admin* memiliki akses secara menyeluruh mengenai sistem pada penelitian ini seperti menambah dan mengedit data, melihat statistic pengujian model, menampilkan grafik dan melihat *history*.

Alur dari sistem ini ialah *user* dapat mengisi kalimat di menu Analisa komentar pada *application form*, kalimat yang di *input*-kan kemudian akan mengalami tahapan *pre-processing* yang kemudian akan diuji dengan model yang telah dibuat. Hasil pada alur ini ialah akan ditampilkan hasil identifikasi berupa probabilitas pada setiap kelas, probabilitas tertinggi menandakan bahwa kelas tersebut merupakan hasil identifikasi. Untuk alur analisa postingan instagram, *User* akan diminta untuk memasukkan alamat *url* post instagram, kemudian sistem akan melakukan proses *scraping* yang kemudian hasil dari *scraping* tersebut akan mengalai proses *text preprocessing* dan kemudian akan dilakukan pengujian model, data hasil pengujian akan ditampilkan ke dalam tabel dengan hasil username, komentar, hasil identifikasi serta validitas pengujian. Pemilihan media sosial instagram berdasarkan hasil kuesioner yang telah dilakukan di mana instagram merupakan media sosial yang mendapat rataan hasil tertinggi. Arsitektur sistem secara keseluruhan dapat dilihat pada gambar 3.49.



**Gambar 3.49** Arsitektur Sistem Secara Keseluruhan

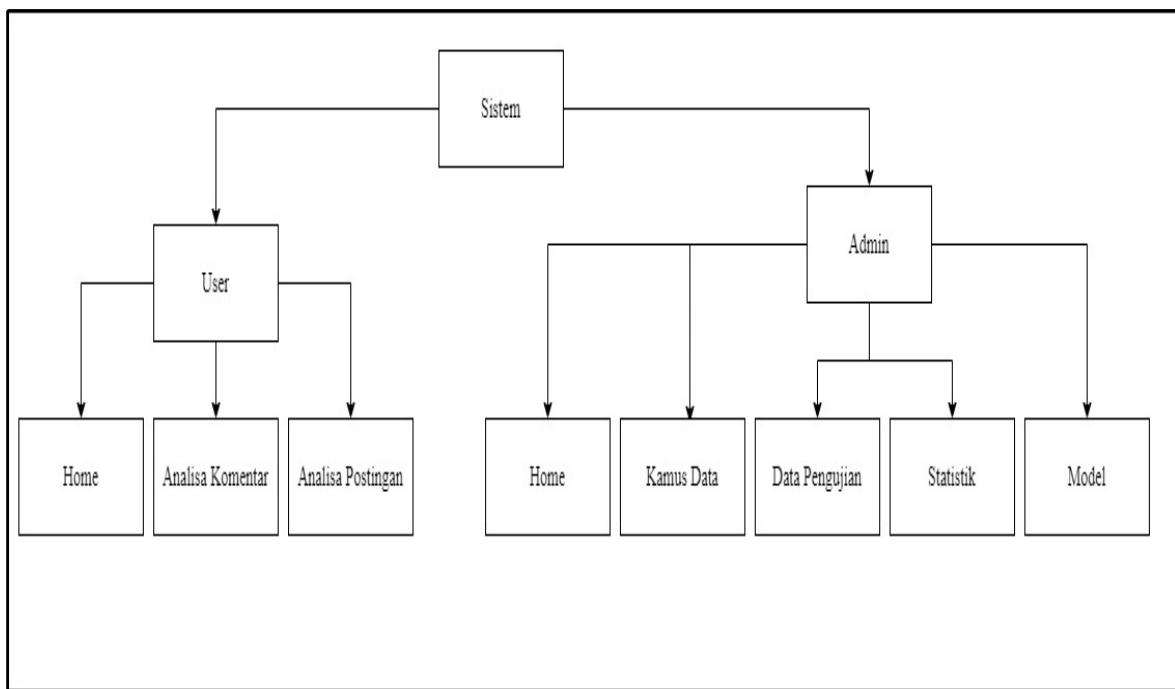
### 3.4.2. Struktur Menu

Struktur menu pada sistem ini terdiri dari struktur menu untuk *user* dan untuk *admin*.

Pada sistem ini *user* memiliki akses yang lebih terbatas dibandingkan dengan *admin*. Menu yang terdapat pada *user* adalah *Home*, Analisa komentar dan Analisa Postingan. Pada menu *home*, *user* diberikan informasi mengenai *cyberbullying*, dalam menu Analisa komentar *user* dapat melakukan pengujian terhadap komentar yang di masukan memalui *form* yang kemudian akan di proses oleh sistem, sedangkan pada menu analisa postingan *user* dapat melakukan analisa terhadap komentar yang ada dalam postingan instagram.

Sedangkan pada pihak *admin*, tersedia lima menu utama yaitu *Home*, kamus, data, *statistic* dan model. Halaman *home* berisi tentang informasi umum seperti informasi jumlah data yang digunakan tiap kelas dan grafik lainnya, pada menu kamus berisi data perbaikan yang digunakan dalam proses *preprocessing*, pada menu data berisi informasi mengenai *dataset* yang digunakan, dan pengujian yang telah dilakukan oleh *user*. Pada menu statistik

berisi informasi penjabaran tentang efektifitas model dan kemudian secara rinci akan dijelaskan dalam menu model yang berisi pengujian data, hasil *confusion matrix* dan *classification report*. Struktur menu dapat dilihat pada gambar 3.50.

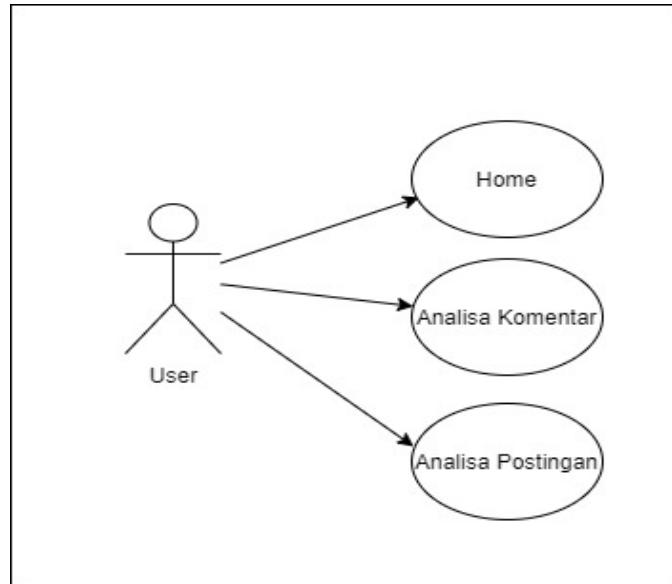


**Gambar 3.50** Struktur Menu

### 3.4.3. Perancangan Model

#### a. Usecase diagram user

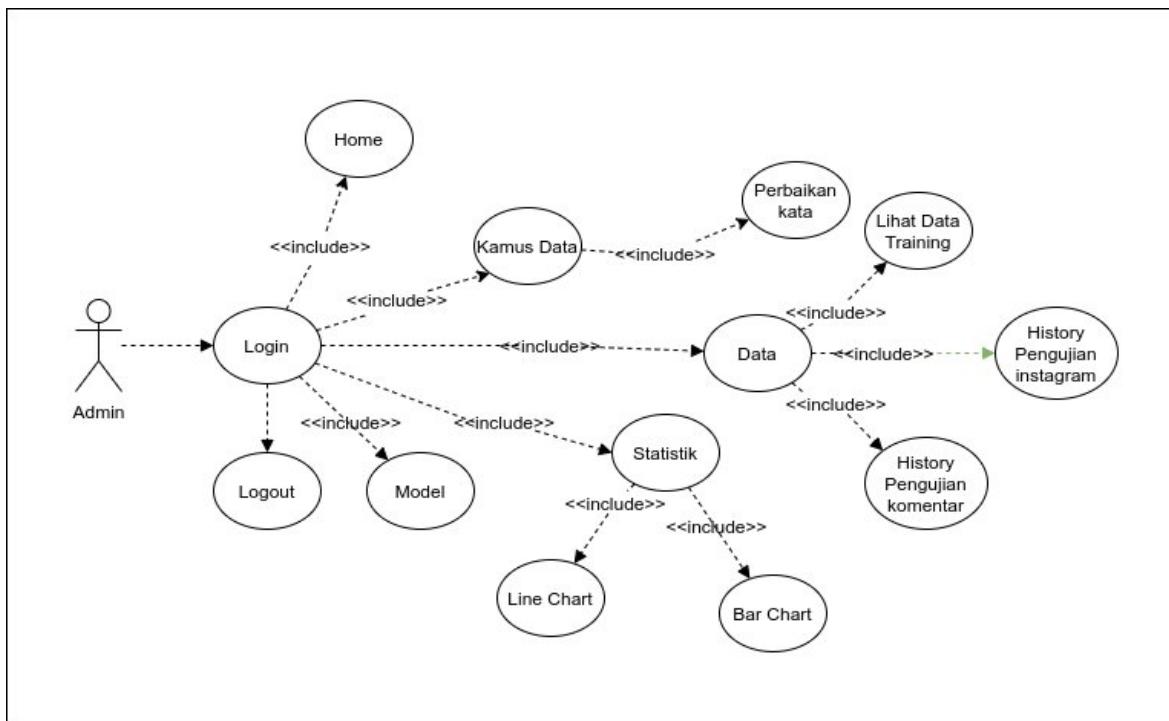
*Use-case diagram user* terdiri dari uji kalimat dan postingan instagram serta menu tambahan tentang yang berisikan informasi mengenai *cyberbullying*. Pada analisa komentar, *user* akan mengisi sebuah komentar kepada *application form*, kemudian *application form* akan mengirimkan *request* kepada *server* untuk dianalisis, *server* akan memberikan umpan balik berupa hasil dari analisis tersebut beserta dengan probabilitas kelas lainnya. Kemudian untuk menu analisa postingan, *user* dapat mengisikan alamat kiriman instagram kemudian sistem akan melakukan *scraping* sesuai dengan alamat yang diisikan menggunakan proses automasi, hasil dari identifikasi akan ditampilkan kepada *user*. *Use-case diagram* dapat dilihat pada gambar 3.51.



**Gambar 3.51** Usecase diagram user

b. *Usecase diagram admin*

Secara keseluruhan, *usecase diagram admin* disajikan pada gambar 3.52.



**Gambar 3.52** Usecase Diagram Admin

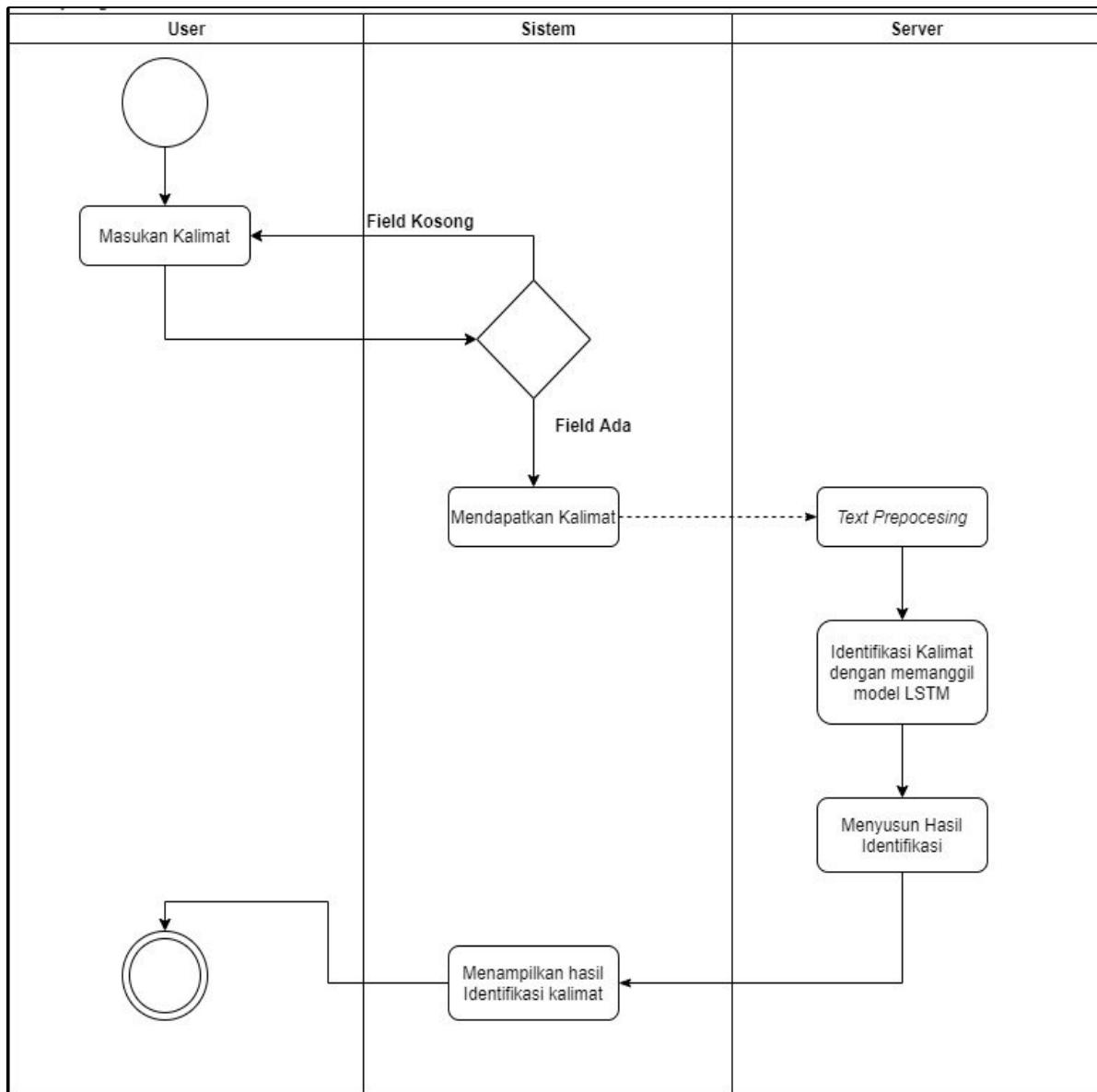
Berdasarkan gambar 3.52, untuk masuk ke dalam menu *admin* yang berisikan bermacam-macam menu mengenai sistem, diperlukan langkah *login* terlebih dahulu. *Include* yaitu *user* terlebih dahulu harus melalui proses lain untuk menjalankannya, sedangkan *extend* tidak

memerlukan proses lain untuk menjalankannya. Terdapat lima menu utama dalam admin yang memerlukan proses *login* yaitu *home*, kamus, data, *statistic* dan model.

c. *Activity Diagram User Analisa Komentar*

Diagram ini menggambarkan aktivitas *user* untuk melakukan pengujian kalimat.

*Activity diagram* analisa komentar disajikan dalam gambar 3.53.



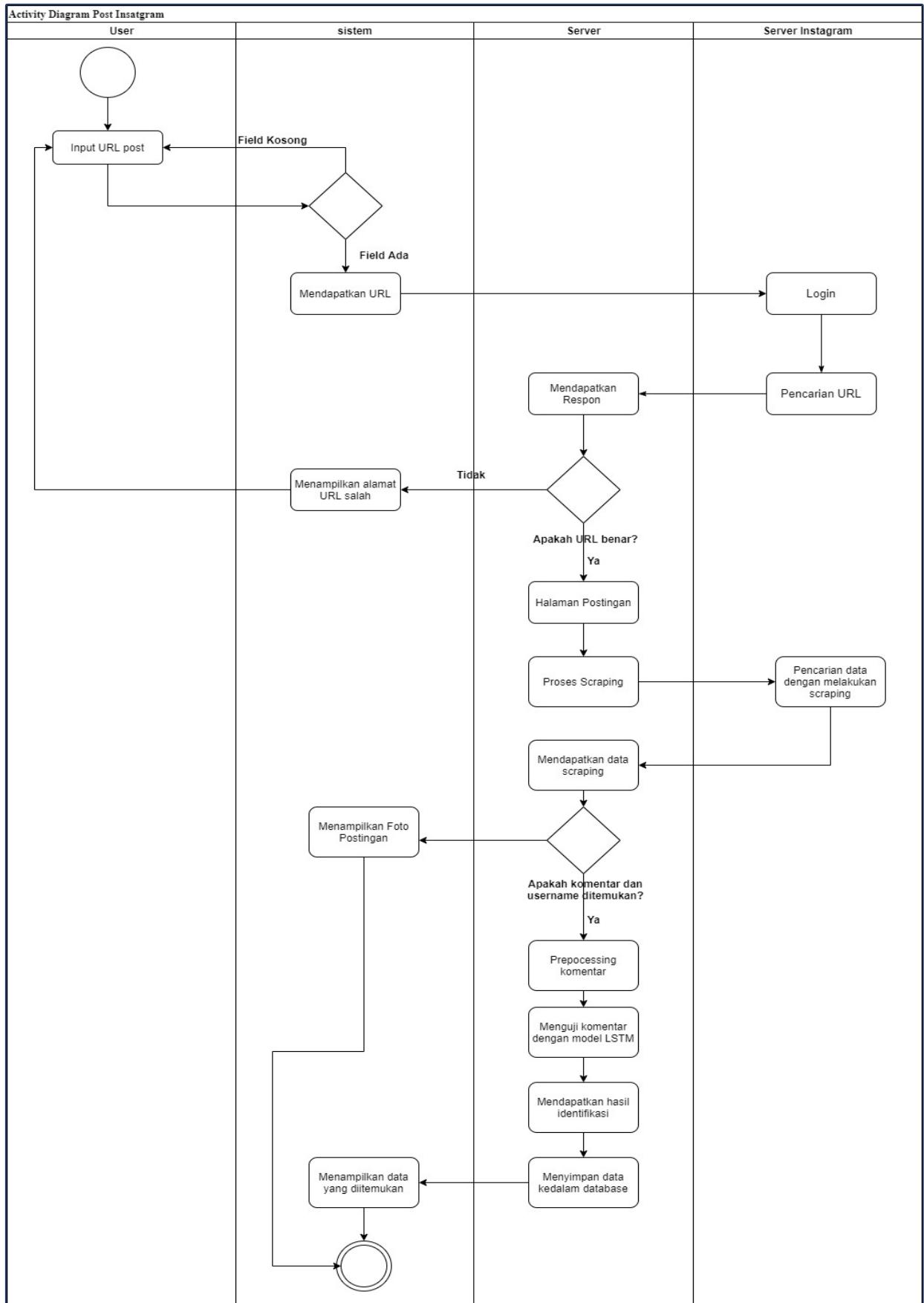
Gambar 3.53 *Activity Diagram* Analisa Komentar

Berdasarkan gambar 3.53, *user* memasukkan kalimat dan kemudian sistem akan melakukan *pre-processing*, dilanjutkan dengan pengujian kalimat menggunakan model LSTM yang

telah terbentuk, hasil akhir komentar akan di klasifikasikan ke dalam salah satu kelas *cyberbullying*, *Irrelevant*, netral ataupun bukan *cyberbullying*.

d. *Activity Diagram User Analisa Postingan*

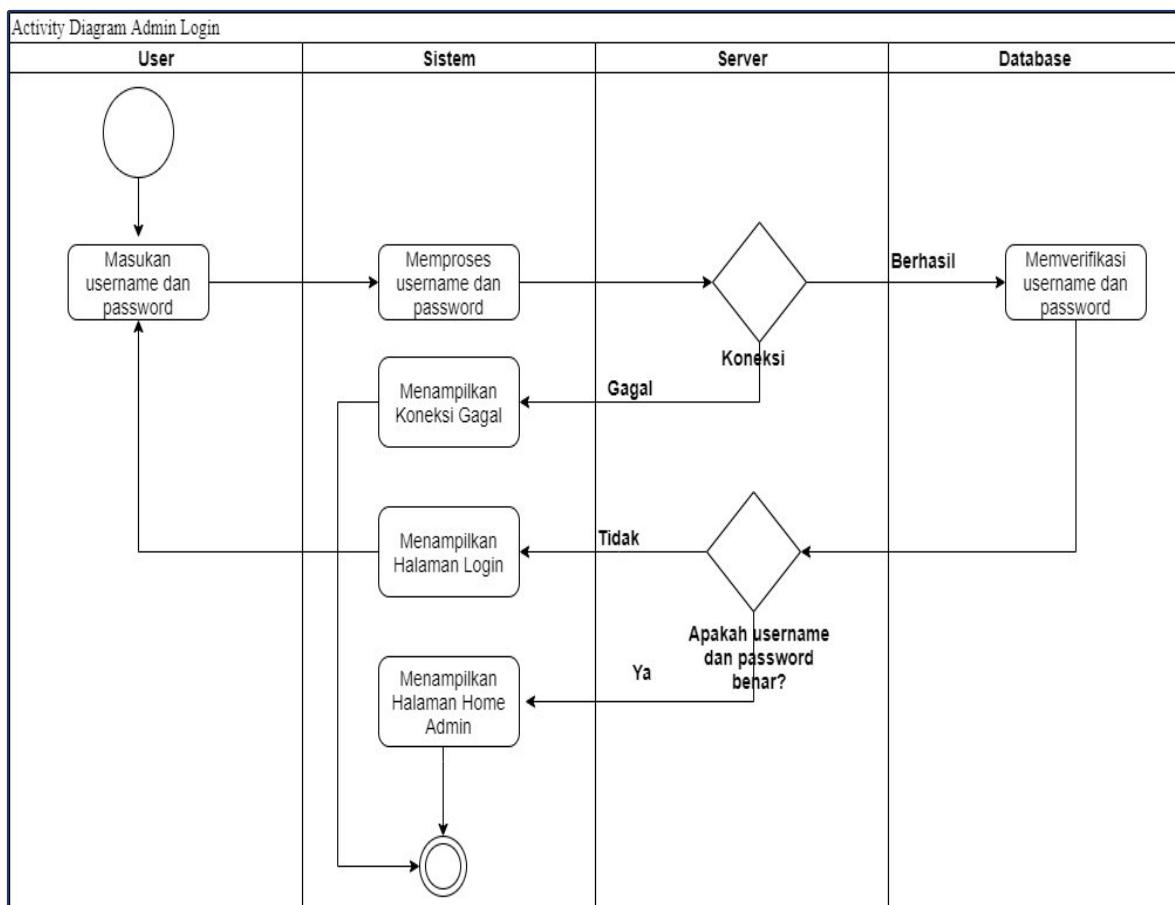
Diagram ini menggambarkan aktivitas *user* untuk melakukan pengujian terhadap komentar yang didapatkan dari sebuah halaman kiriman instagram. *User* menginputkan *url* yang berisikan alamat kiriman instagram, kemudian sistem secara otomatis akan melakukan proses *login* dan kemudian akan menuju alamat kiriman tersebut dan melakukan proses *scraping*. Data dari proses *scraping* tersebut kemudian akan dilanjutkan dengan *preprocessing* dan pengujian ke dalam model yang telah dibuat, kemudian hasil dari pengujian tersebut akan disimpan ke dalam *database*. komentar beserta hasil identifikasi akan ditampilkan kepada *user*. *Activity diagram* analisa postingan dapat dilihat pada gambar 3.54.



Gambar 3.54 Activity Diagram User Analisa Postingan

e. *Activity Diagram Admin Login*

Diagram ini menggambarkan alur aktivitas *admin* dalam melakukan proses *login*. *Admin* menuju halaman *login* yang berisi *login form* berupa *username* dan *password*. *Admin* harus sudah terdaftar sebelumnya sehingga sudah memiliki *username* dan *password*. Setelah mengisi *username* dan *password* kemudian *server* akan menghubungkan dengan *database* untuk mengecek apakah *username* dan *password* yang diisikan benar, jika benar maka akan dilanjutkan ke halaman home *admin* sedangkan jika *login* gagal maka akan kembali ke *login form*. *Activity diagram* untuk proses ini dapat dilihat pada gambar 3.55.

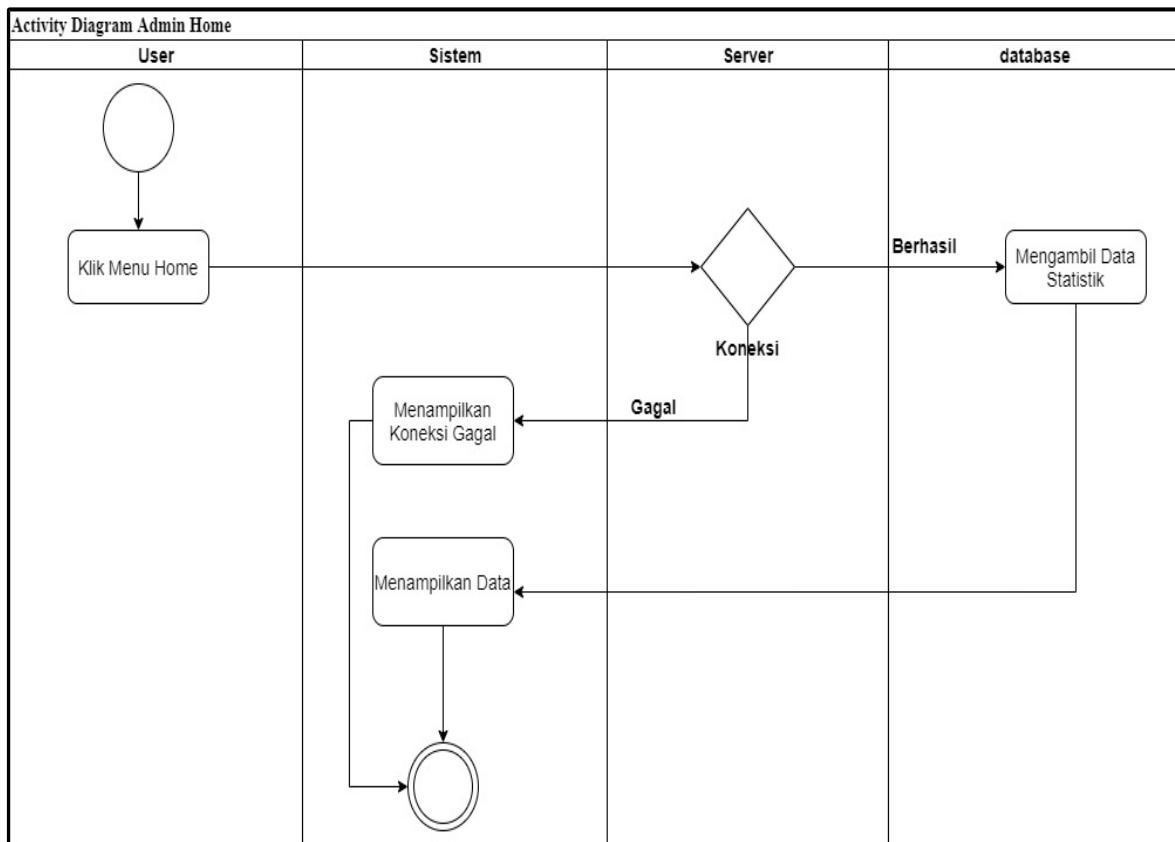


Gambar 3.55 *Activity Diagram Admin Login*

f. *Activity Diagram Admin Home*

Diagram ini menjelaskan alur *admin* dalam menampilkan data. Setelah melalui proses *login*, admin akan diarahkan ke halaman *home* dengan proses penampilan *statistic* pengujian serta *statistic* data yang digunakan, selain itu untuk Kembali ke halaman ini admin

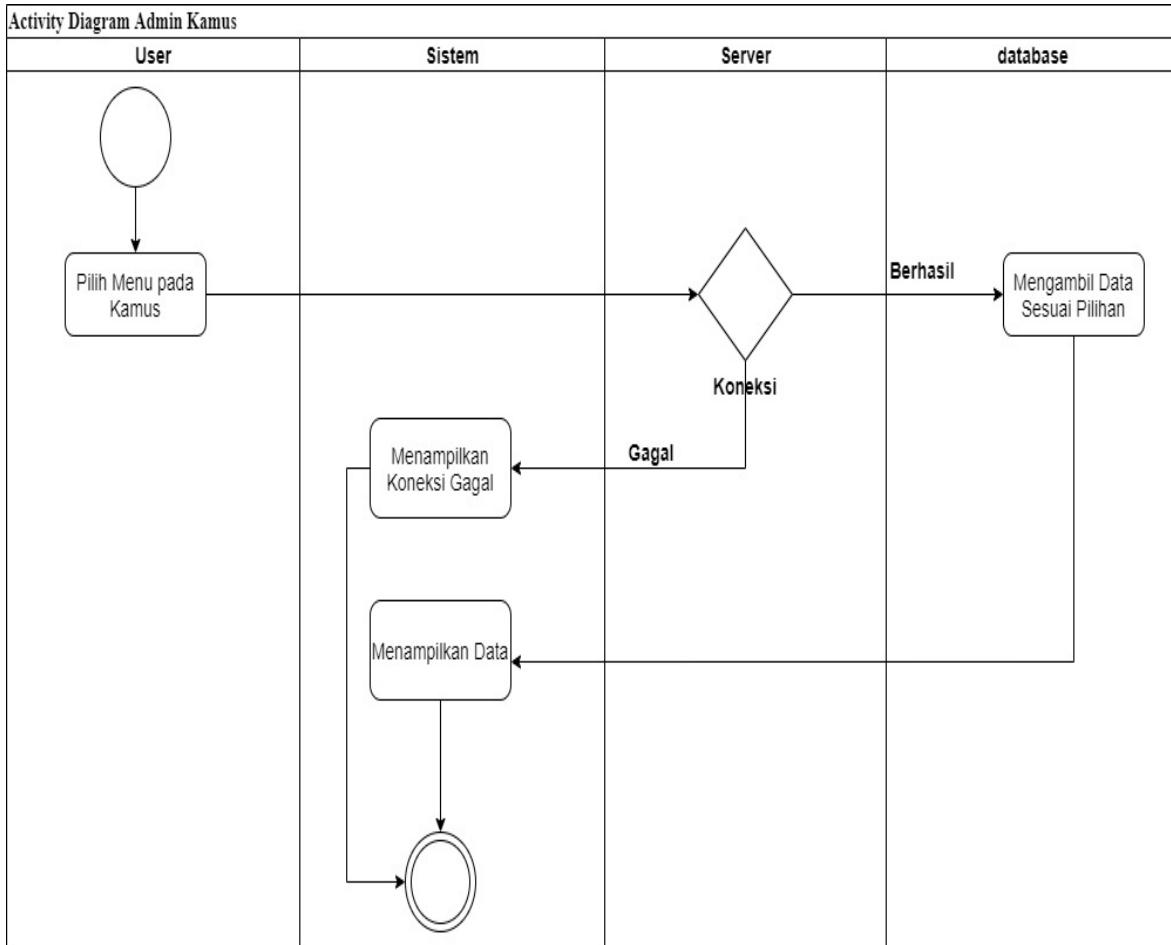
dapat memilih menu *home* dalam menu. Setelah itu *server* akan mencari *dataset* yang digunakan dalam pembuatan model pada *database*, *dataset* tersebut kemudian akan ditampilkan dalam bentuk tabel. *Activity Diagram* pada alur ini dapat dilihat pada gambar 3.56.



**Gambar 3.56** Activity Diagram Admin Home

g. *Activity Diagram Admin Kamus*

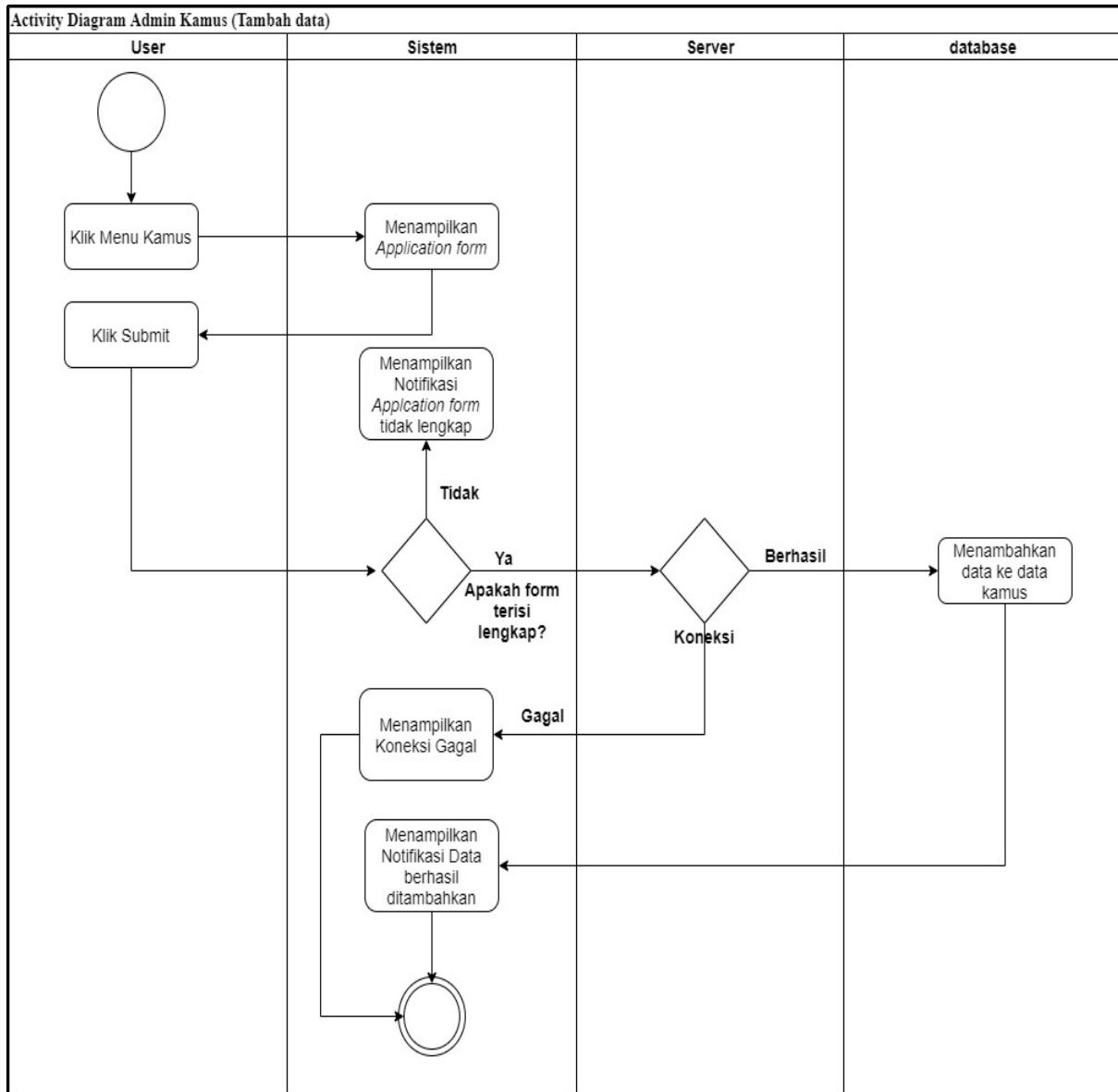
Diagram ini menggambarkan alur *admin* menampilkan kamus data yang digunakan, di dalam menu kamus berisi data perbaikan kata. Dalam menampilkan data, *user* melakukan *request* kepada sistem dengan cara memilih salah menu kamus dan kemudian *server* akan melakukan koneksi ke *database* untuk mencari data yang sesuai dengan menu tersebut. *Diagram* untuk alur ini dapat dilihat pada gambar 3.57.



**Gambar 3.57 Activity Diagram Admin Kamus**

h. *Activity Diagram Admin Kamus Tambah Data*

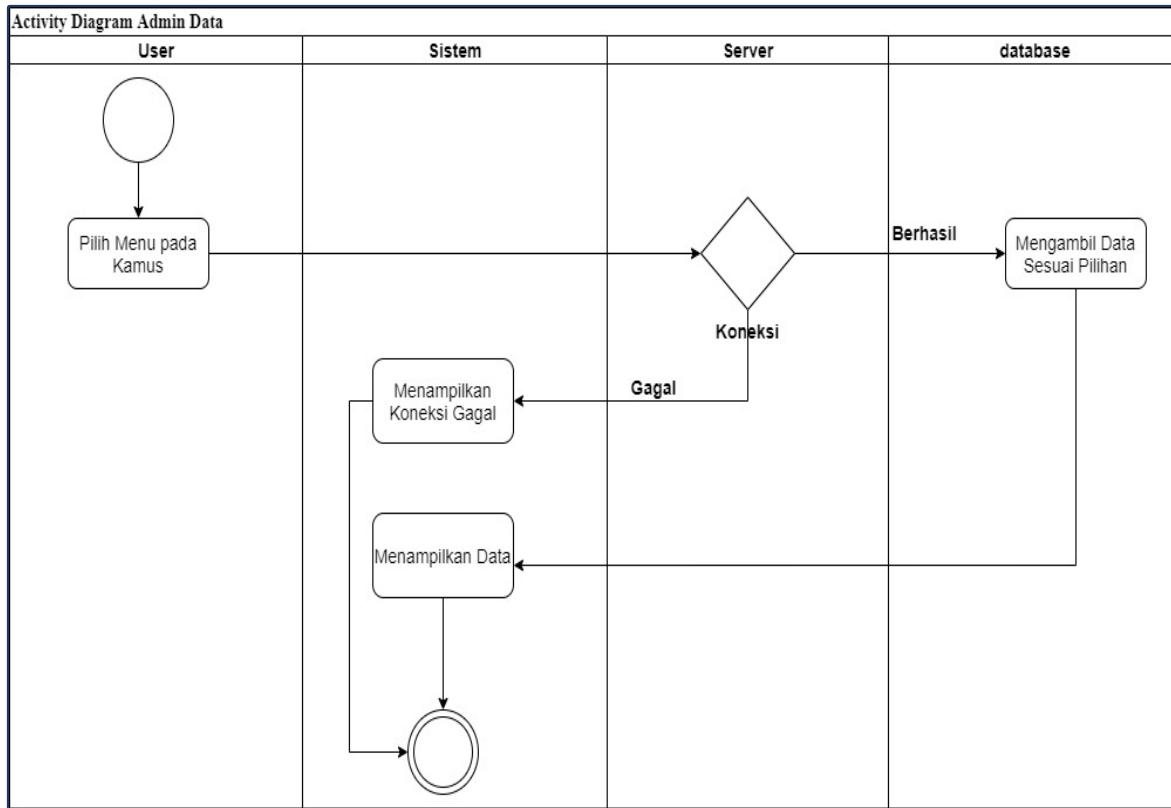
Diagram ini menjelaskan tentang alur *admin* dalam melakukan penambahan data pada *database* kamus. Admin melakukan *request* terhadap *server* dengan cara memilih salah satu sub menu yang ada kemudian admin melakukan pengisian data pada *form* yang telah disediakan. Jika *form* tersebut lengkap maka *form* akan disimpan ke *database* dan jika terdapat *form* yang belum lengkap maka sistem akan memberikan informasi bahwa terjadi kesalahan. *Activity diagram* alur ini dapat dilihat pada gambar 3.58.



Gambar 3.58 Activity Diagram Admin Kamus (Tambah Data)

i. *Activity Diagram Admin Data*

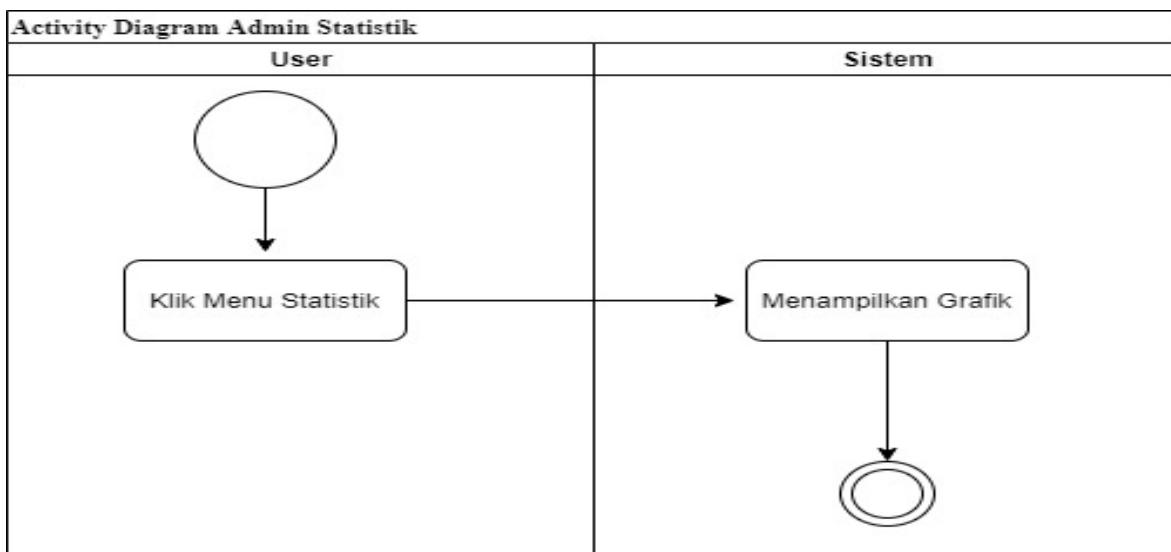
Diagram ini menggambarkan alur *admin* dalam menampilkan data berupa *data training*, data pengujian postingan dan data pengujian komentar. Diagram ini menggambarkan *admin* melakukan *request* kepada *server* sesuai dengan sub menu yang dipilih kemudian server akan melakukan koneksi terhadap *database* dan menampilkan data sesuai dengan sub menu yang dipilih oleh admin. *Activity diagram* alur ini dapat dilihat pada gambar 3.59.



**Gambar 3.59 Activity Diagram Admin Data**

j. *Activity Diagram Admin Statistik*

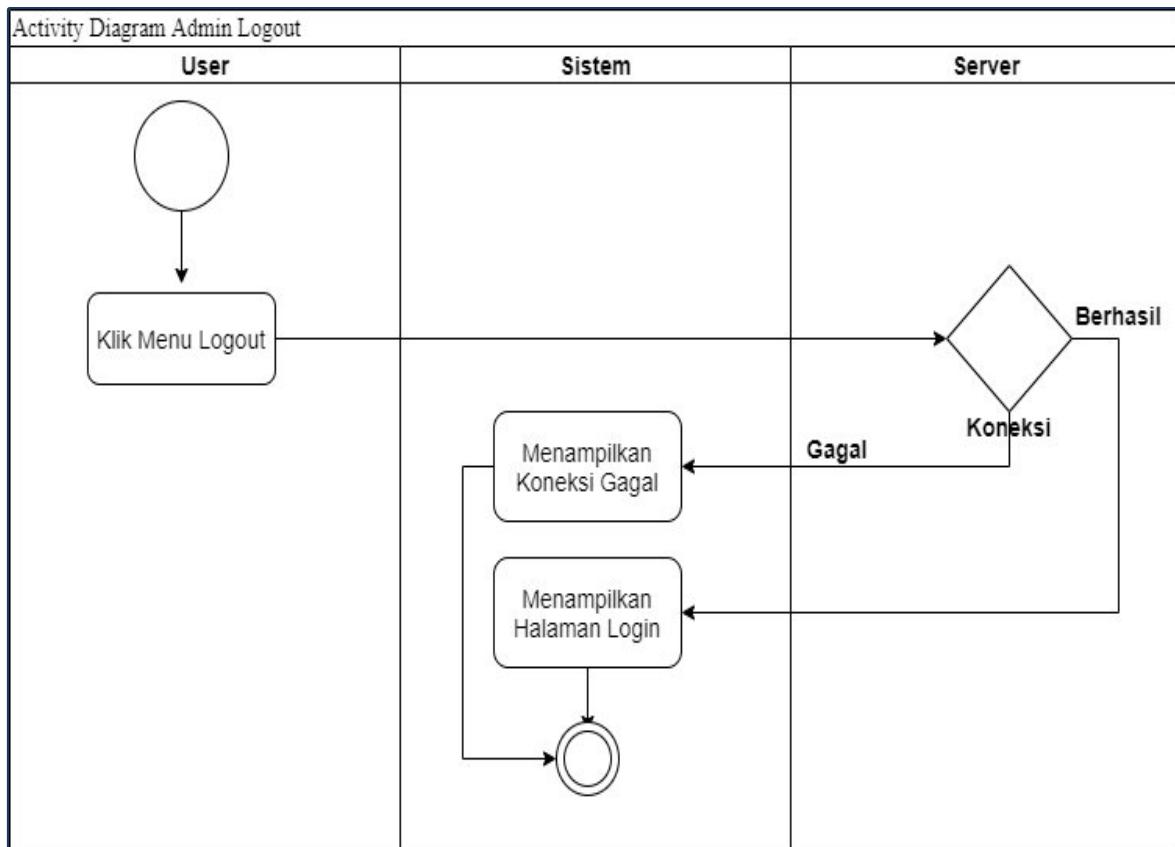
Diagram ini menggambarkan alur *admin* dalam menampilkan grafik. *Admin* memilih menu *statistic* dan melakukan pilihan pada sub menu yang tersedia. *Server* kemudian akan menampilkan grafik sesuai dengan sub menu yang dipilih oleh admin. *Activity diagram* pada alur tersebut dapat dilihat pada gambar 3.60.



**Gambar 3.60 Activity Diagram Admin Statistik**

k. *Activity Diagram Admin Logout*

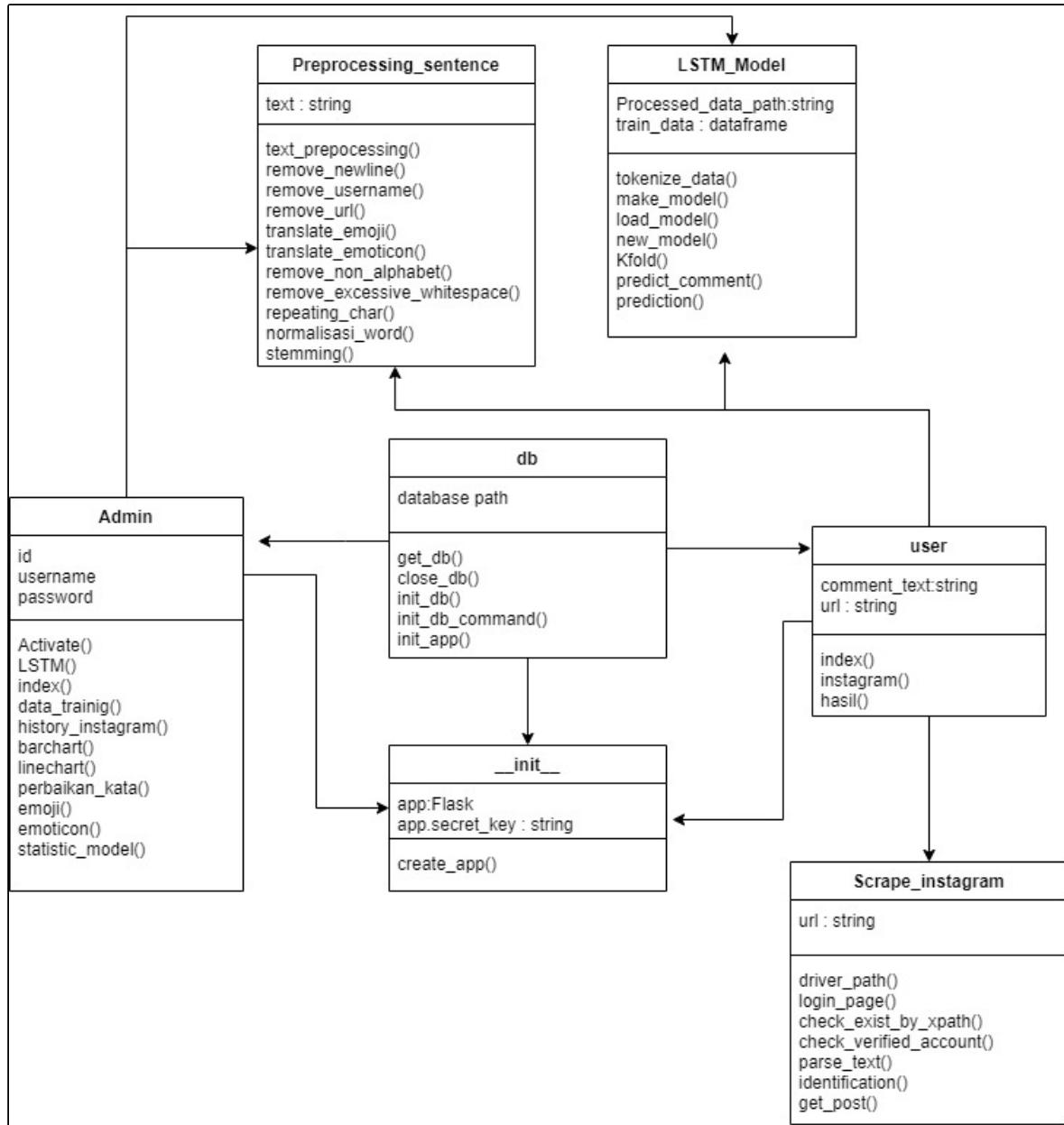
Diagram ini menggambarkan alur *admin* dalam melakukan *logout*. *Admin* meng-klik menu *logout* dan kemudian akan dilakukan *request* untuk proses *logout* pada *server*, kemudian setelah proses tersebut maka sistem akan menampilkan halaman *login* untuk *admin*. *Activity diagram* pada alur tersebut dapat dilihat pada gambar 3.61.



Gambar 3.61 *Activity Diagram Admin Logout*

l. *Class Diagram*

Perancangan *class diagram* merupakan penggambaran struktur meliputi atribut dan *method* yang ada pada setiap kelas. Kelas diagram pada sistem ini terdiri dari kelas *auth* untuk melakukan authentikasi, *db* untuk melakukan koneksi database, *admin* untuk menjalankan akses yang dapat dilakukan admin, kelas *user* untuk menjalankan akses yang dilakukan oleh *user*, kelas *preprocessing\_sentence* untuk melakukan *preprocessing*, kelas *lstm\_model* untuk menjalankan model dan kelas *ScrapePost* untuk melakukan *scraping* pada instagram. *Class diagram* pada sistem ini disajikan dalam bentuk gambar 3.62.

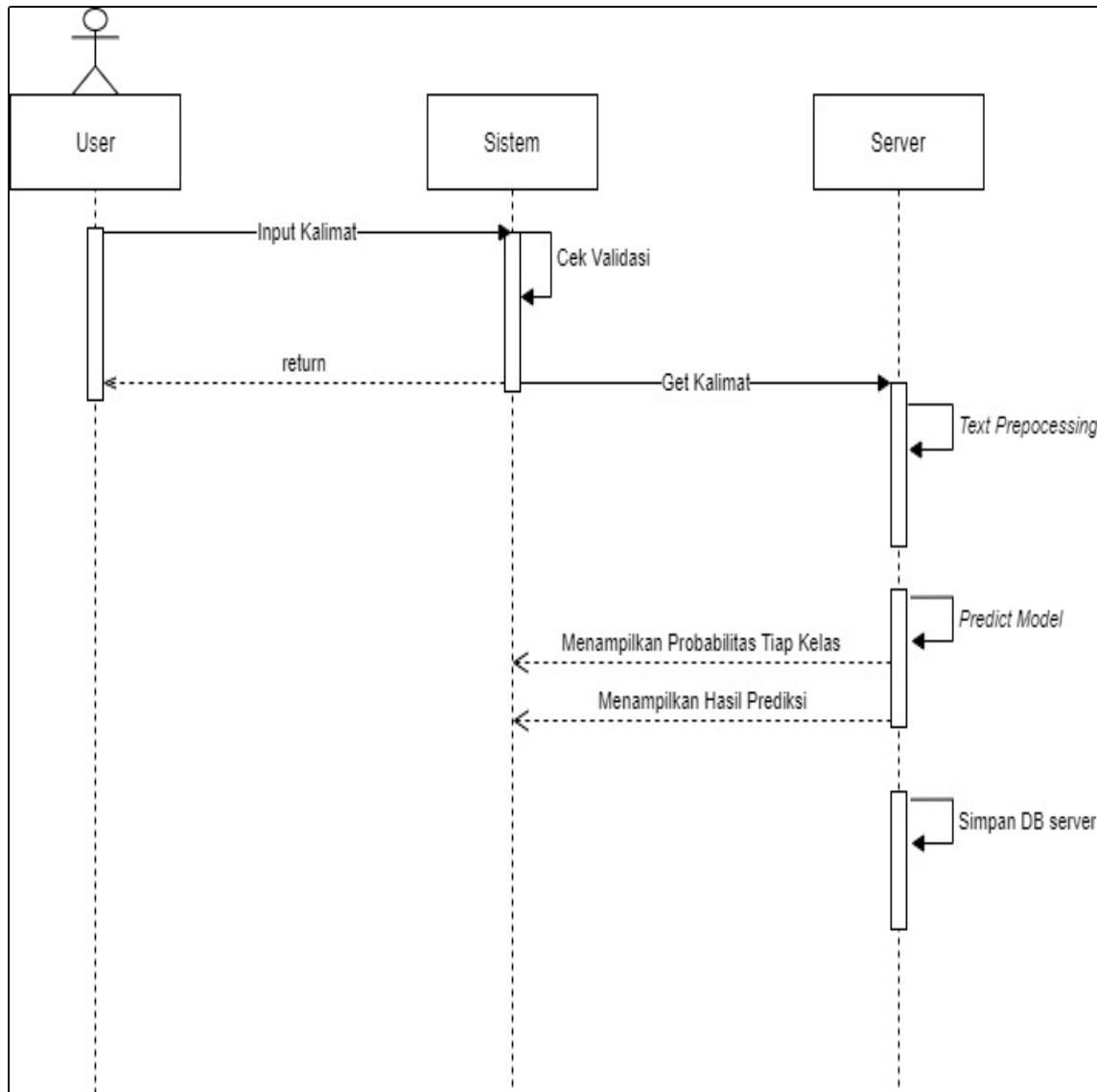


Gambar 3.62 Class Diagram

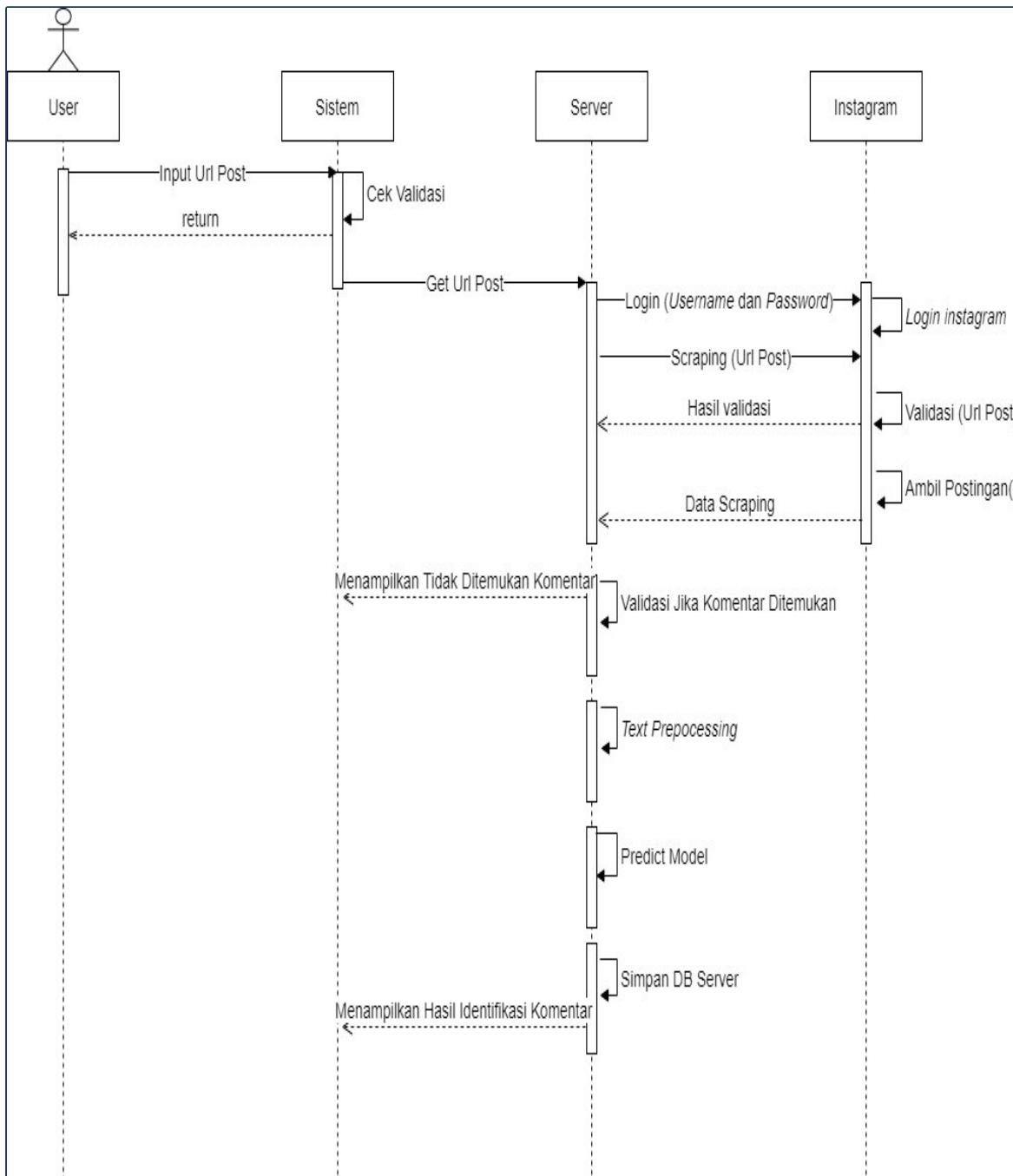
#### m. Sequence Diagram

*Sequence Diagram* menggambarkan kelakuan dari objek-objek dengan mendeskripsikan berdasarkan urutan waktu dan pesan yang diterima maupun dikirimkan antar objek. *Sequence Diagram* digunakan biasa digunakan untuk menggambarkan langkah-langkah yang dilakukan sebagai *response* dari sebuah aktivitas untuk menghasilkan *output* tertentu. Diawali dari proses paling mendasar dari aktivitas tersebut, proses perubahan yang

terjadi secara *internal* dan *output* yang dihasilkan. *Sequence diagram* user melakukan analisa komentar dan analisa postingan dapat dilihat pada gambar 3.63 dan 3.64.



Gambar 3.63 Sequence Diagram User Analisa Komentar



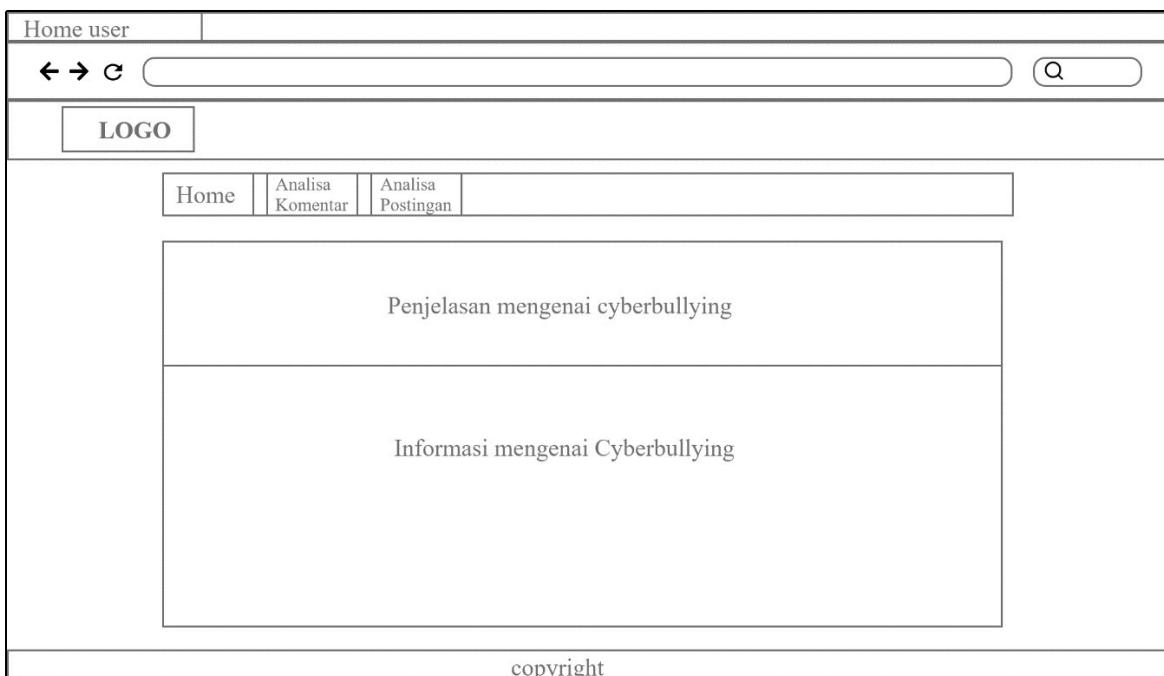
Gambar 3.64 Sequence Diagram User Analisa Postingan

#### 3.4.4. Perancangan User Interface

Perancangan *user interface* dimaksudkan untuk memberikan gambaran mengenai rancangan tampilan antar muka yang akan dibuat pada sistem. Penggunanya adalah *user* sebagai penerima informasi serta *admin* sebagai akses keluar masuk data dan informasi.

a. Perancangan Halaman Utama User

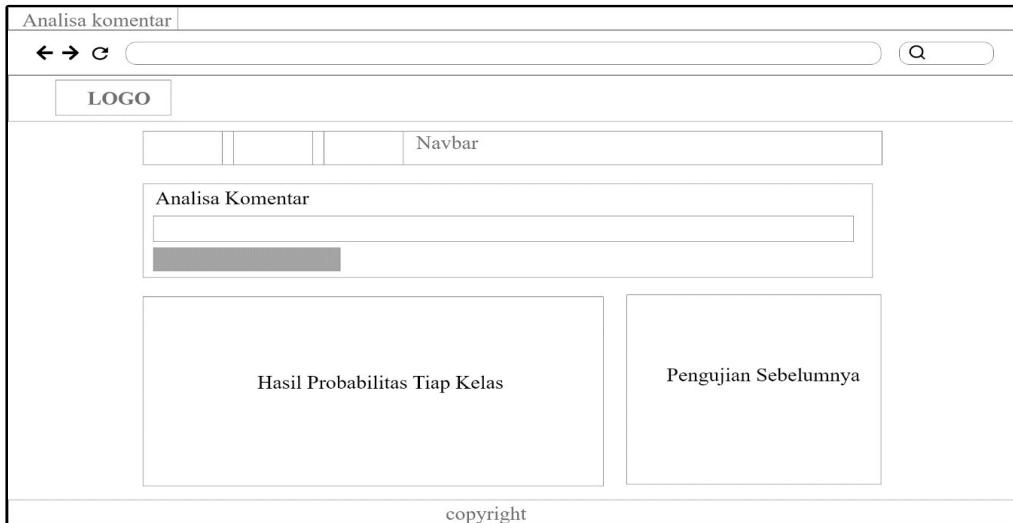
Perancangan halaman utama merupakan tampilan awal ketika *user* menjalankan sistem. Pada halaman utama, terdapat tiga menu pada *navigation bar* bagian atas di mana menu tersebut adalah *home*, analisa komentar dan analisa postingan. Pada menu utama *home*, terdapat beberapa informasi mengenai *cyberbullying* dan informasi lainnya. Rancangan *user interface* ini dapat dilihat pada gambar 3.65.



**Gambar 3.65** Perancangan Halaman Utama *User*

b. Perancangan Halaman *User* Analisa Komentar

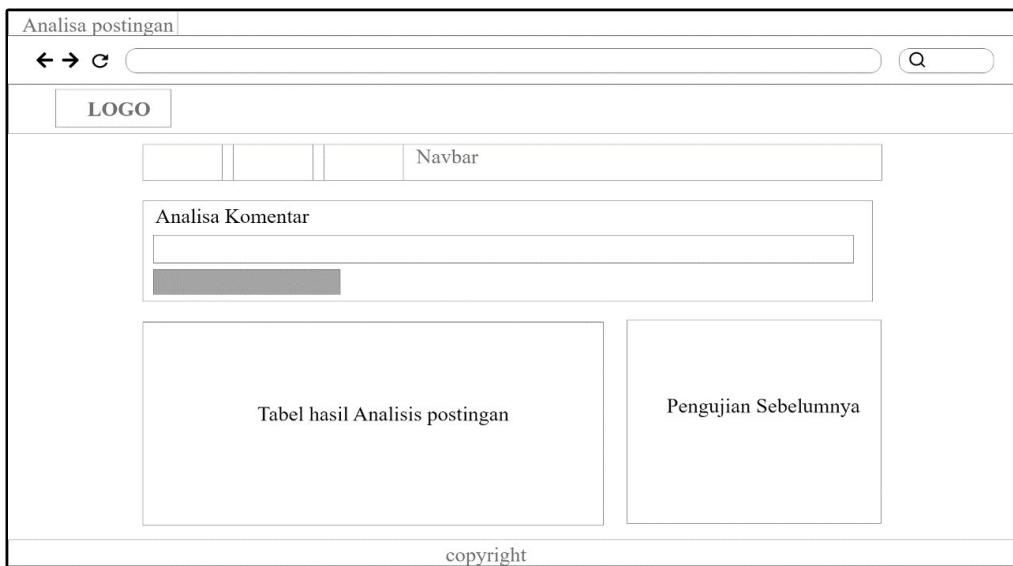
Perancangan halaman analisa komentar menampilkan *navigation bar* beserta sebuah *container* yang berisi *form* beserta parameter perhitungan probabilitas tiap kelas. *Form* tersebut memiliki sebuah *field* untuk memasukkan komentar dan sebuah tombol untuk melakukan pengujian. Selain itu di bagian bawah juga terdapat rekaman sebelumnya. Rancangan halaman tersebut dapat dilihat pada gambar 3.66.



**Gambar 3.66** Perancangan Halaman *User* Analisa Komentar

c. Perancangan Halaman *User* Analisa Postingan

Perancangan halaman *user* analisa postingan memiliki struktur yang sama dengan halaman analisa komentar, hanya saja *text form* yang ada hanya bisa digunakan untuk melakukan *input* berupa *url* instagram. Di bagian bawah terdapat dua buah *container* yang berisi daftar halaman yang telah dilakukan pengujian serta sebuah tabel untuk menampilkan hasil pengujian. Perancangan halaman tersebut dapat dilihat pada gambar 3.67.



**Gambar 3.67** Perancangan Halaman *User* Analisa Postingan

d. Perancangan Halaman *Admin Login*

Pada perancangan ini, ditampilkan *form* yang berisikan dua buah *text field* di mana *text field* tersebut berfungsi untuk memasukkan *username* dan *password*, selain itu terdapat

sebuah tombol untuk mengirimkan *form* tersebut kepada sistem. Perancangan halaman ini dapat dilihat pada gambar 3.68.

The diagram shows a wireframe of a web browser window. At the top, there's a header bar with the word 'Admin' on the left and standard browser controls (back, forward, search) on the right. Below the header is a large central area containing a 'LOGIN' form. The form has fields for 'Username' and 'Password', a 'Remember me' checkbox, and a 'Submit' button. At the bottom of the page, there's a footer bar with the text 'Copyright @2020 All Right Reserved'.

**Gambar 3.68** Perancangan Halaman *Admin Login*

e. Perancangan Halaman Utama *Admin*

Perancangan halaman utama *admin* berisi mengenai informasi tentang jumlah data tiap kelas pada data *training* serta grafik *K-fold* hasil pengujian model. Selain itu ditampilkan jumlah data tiap kamus yang digunakan. Selain itu terdapat menu bar yang berisi menu-menu yang dapat diakses oleh *admin*. Perancangan halaman ini dapat dilihat pada gambar 3.69.

The diagram shows a wireframe of a web browser window. At the top, there's a header bar with the word 'Admin' on the left and standard browser controls (back, forward, search) on the right. Below the header is a 'LOGO' placeholder. Underneath the logo is a horizontal menu bar with buttons for 'Home', 'Kamus', 'Data', 'Statistik', 'Model', and 'Logout'. To the right of the menu is a large rectangular area divided into four smaller boxes labeled 'Jumlah data cyberbullying', 'Jumlah data Irrelevant', 'Jumlah data Netral', and 'Jumlah data Non Cyberbullying'. Below these four boxes is a large empty area labeled 'GRAFIK K-FOLD'. To the right of this area is another large empty box labeled 'Statistik Data'.

**Gambar 3.69** Perancangan Halaman Utama *Admin*

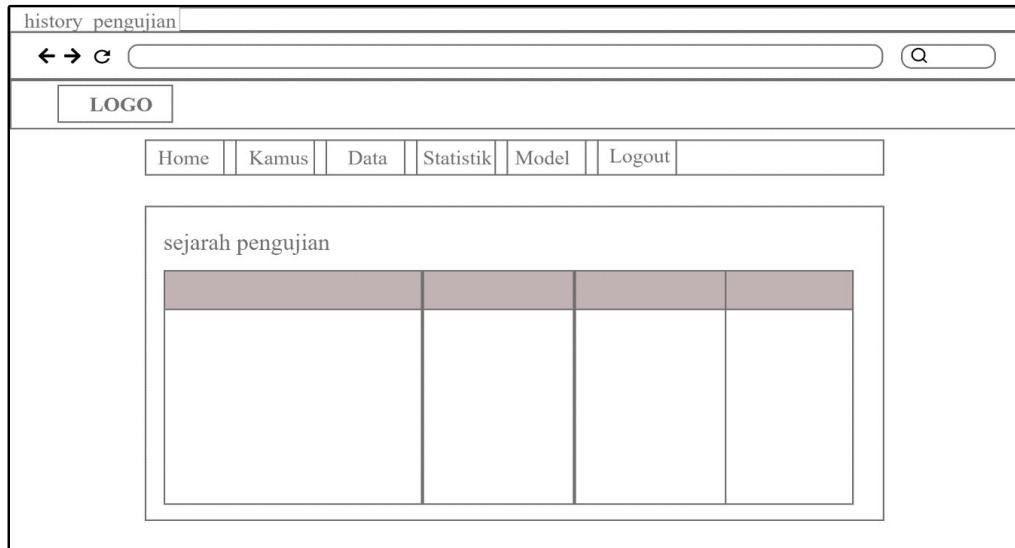
f. Perancangan Halaman *Admin Kamus*

Perancangan halaman ini berisi tabel data kamus sesuai dengan sub menu beserta sebuah *form* untuk melakukan penambahan data. Apabila *form* tersebut dijalankan terdapat sebuah *box* dialog yang akan muncul apabila *form* tersebut kosong saat terjadi pengiriman ke *server*. Perancangan halaman ini dapat dilihat pada gambar 3.70.

**Gambar 3.70** Perancangan Halaman *Admin Kamus*

g. Perancangan Halaman Admin Riwayat Uji Komentar

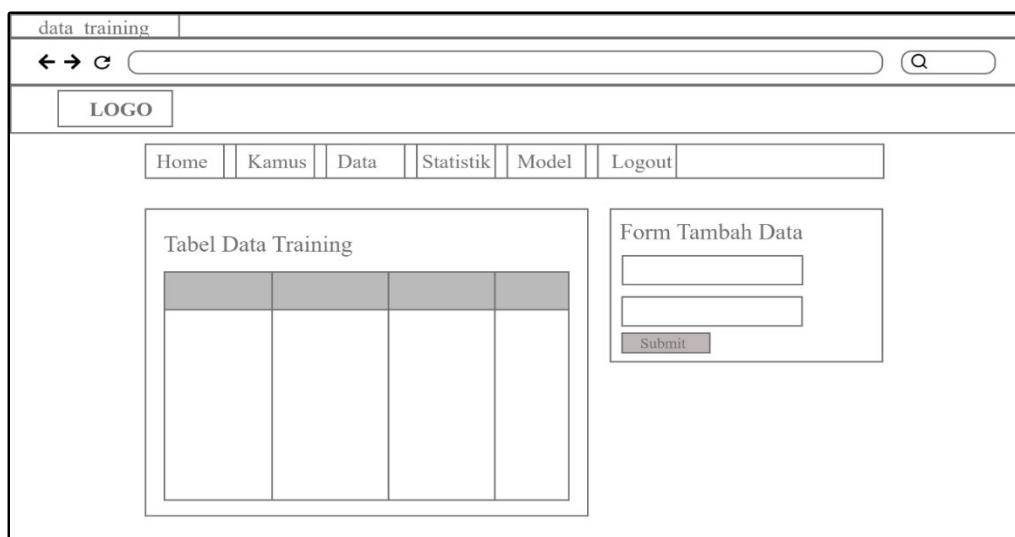
Pada perancangan ini akan ditampilkan sebuah tabel yang berisikan riwayat pengujian, tabel tersebut berisikan *username*, komentar, hasil pengujian dan validitas. Perancangan ini dapat dilihat pada gambar 3.71.



**Gambar 3.71** Perancangan Halaman Admin Riwayat Uji Komentar

h. Perancangan Halaman Admin *Data Training*

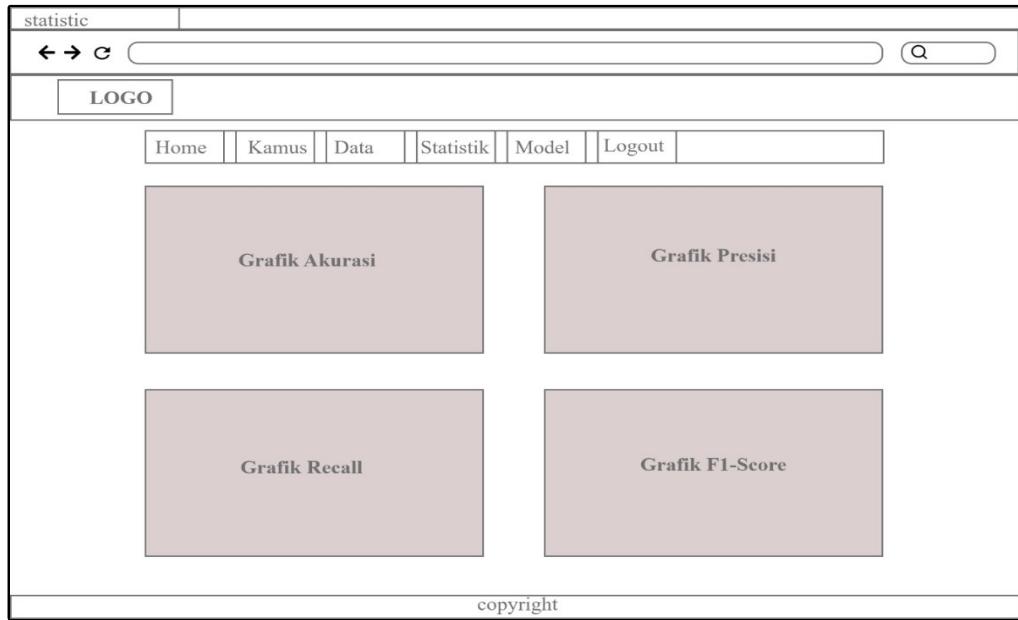
Perancangan halaman ini akan menampilkan tabel *data training* yang berisi data awal, data setelah *preprocessing* dan label data. Selain itu terdapat sebuah *form* yang berisi sebuah *text form* untuk memasukkan data awal serta label data. Perancangan ini dapat dilihat pada gambar 3.72.



**Gambar 3.72** Perancangan Halaman Admin Data Training

i. Perancangan Halaman Admin Statistik

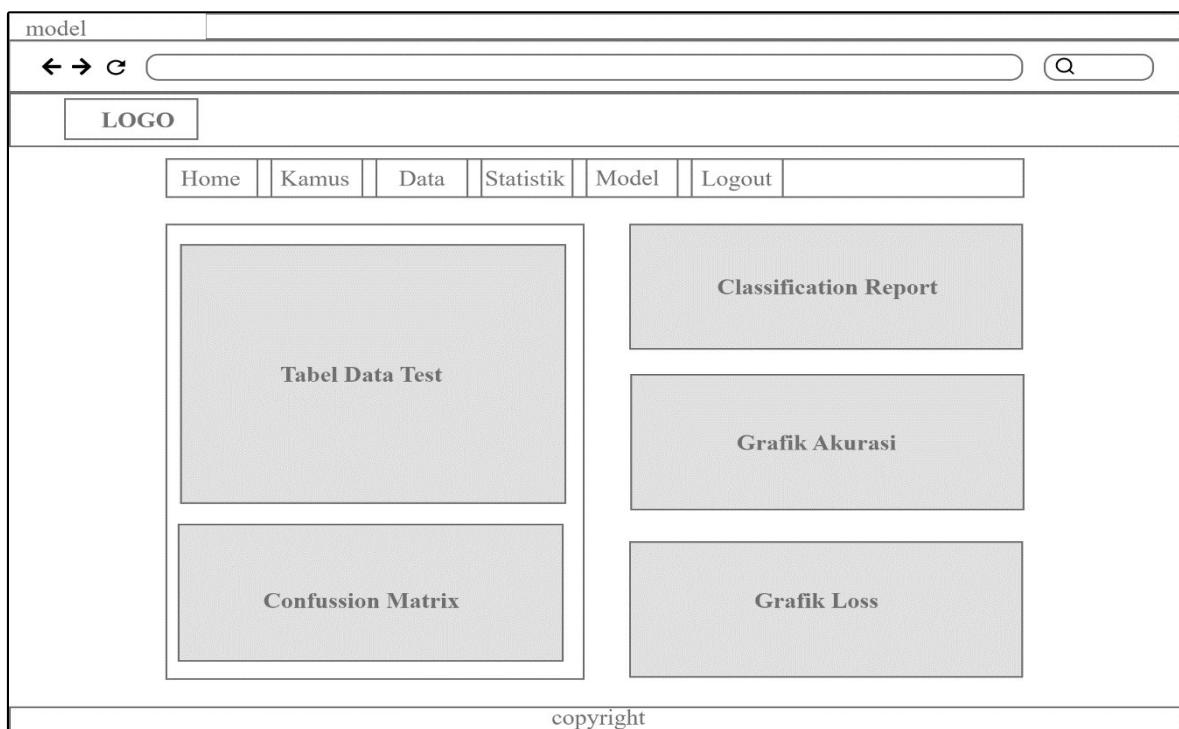
Perancangan halaman ini akan menampilkan empat grafik hasil pengujian *k-fold* secara terpisah, grafik yang akan di tampilkan berisi grafik akurasi, grafik presisi, grafik *recall* dan grafik *f1-score*. Perancangan ini dapat dilihat pada gambar 3.73.



**Gambar 3.73** Perancangan Halaman *Admin* Statistik

j. Perancangan Halaman Admin Model

Perancangan halaman ini akan menampilkan tabel data data uji yang berisi label asli dan hasil prediksi terhadap label tersebut, selain itu di tampilkan *confusion matrix*, *classification report* serta grafik akurasi dan *loss* pengujian. Perancangan ini dapat dilihat pada gambar 3.74.



**Gambar 3.74** Perancangan Halaman *Admin* Model

### 3.5. Rancangan Pengujian

Rancangan pengujian memiliki tujuan untuk mempersiapkan konsep pengujian yang akan digunakan untuk mengukur kinerja sistem yang dibangun. Pengujian perangkat lunak juga memberikan pandangan mengenai perangkat lunak secara lebih objektif dan independen. Pengujian yang akan digunakan dalam penelitian ini adalah dengan melakukan *k-fold cross validation*. Pengujian dilakukan dengan membuat *confusion matrix* yang merupakan tabel yang digunakan untuk mengetahui tingkat klasifikasi sistem. Melalui *confusion matrix* hasil pengujian berupa akurasi, presisi, *recall* dan *F-1 Score* dapat diketahui. Rancangan tabel yang akan digunakan dalam *k-fold cross validation* dan *confusion matrix* disajikan dalam tabel 3.19 dan 3.20.

**Tabel 3.19** Tabel *Confusion Matrix* Model Klasifikasi Komentar

Fold	<i>Confusion Matrix</i>															
	TP				FP				FN				TN			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
1																
2																
Ke-n																

**Tabel 3.20** Tabel Pengujian *K-fold Cross Validation* Model Klasifikasi Komentar

Fold	Akurasi	Presisi	Recall	F1-Score
1				
2				
Ke-n				

Selain pengujian tersebut, untuk mengukur tingkat performa dari algoritma LSTM dilakukan pengujian dengan membandingkan nilai akurasi, presisi, *recall* dan *F1-Score* antara algoritma LSTM dan *Random Forest Classifier*. Perbandingan dilakukan menggunakan data yang sama dengan pembagian antara data latih sebesar 80% dan data uji sebesar 20%. Tabel hasil perbandingan performa algoritma disajikan dalam tabel 3.21.

**Tabel 3.21** Tabel Perbandingan Performa Algoritma

Parameter	LSTM	Random Forest
Accuracy		
Presisi		
Recall		
F1-Score		

## **BAB IV**

### **HASIL, PENGUJIAN DAN PEMBAHASAN**

Hasil, pengujian dan pembahasan merupakan tahap sistem ini sudah siap untuk digunakan. Dari pembahasan ini, akan diketahui sistem ini sudah mencapai tujuan yang diinginkan berdasarkan analisis dan perancangan yang telah dilakukan. pembahasan akan dibagi menjadi pembahasan hasil kuesioner dan pembahasan hasil penelitian.

#### **4.1. Hasil Kuesioner**

Terdapat 132 tanggapan dari berbagai usia dan dengan berbagai macam latar belakang pekerjaan. Cara untuk menghitung presentase pada kuesioner ini adalah sebagai berikut:

$$\text{Presentase} = (f/n) \times 100\%$$

Keterangan:

f = frekuensi setiap kelas

n = nilai total (jumlah)

##### a. Hasil Pertanyaan Nomor Dua (2)

Dari 132 tanggapan, frekuensi yang menjawab laki-laki sebanyak 64 dan frekuensi yang menjawab perempuan sejumlah 68. Kalkulasi untuk mendapatkan hasil presentase responden yaitu:

$$\text{Presentase jawaban laki-laki} = (64/132)*100\% = 48.5\%$$

$$\text{Presentase jawaban perempuan} = (68/132)*100\% = 51.5\%$$

Berdasarkan hasil tersebut, maka kesimpulan yang di dapatkan adalah responden yang berpartisipasi di dominasi oleh perempuan dengan presentase sebesar 51.5%.

b. Hasil Pertanyaan Nomor Tiga (3)

Dari 132 tanggapan yang di dapatkan, didapatkan hasil berbagai macam kelompok umur yang berpartisipasi dalam kuesioner. Frekuensi jawaban tiap opsi disajikan pada tabel 4.1.

**Tabel 4.1** Frekuensi Jawaban Pertanyaan nomor Tiga (3)

Usia	Frekuensi	Presentase	Valid Presentase	Presentase Komulatif
15	2	1.5	1.5	1.5
17	2	1.5	1.5	3.0
18	5	3.8	3.8	6.8
19	13	9.8	9.8	16.7
20	27	20.5	20.5	37.1
21	36	27.3	27.3	64.4
22	21	15.9	15.9	80.3
23	4	3.0	3.0	83.3
24	2	1.5	1.5	84.8
25	6	4.5	4.5	89.4
26	2	1.5	1.5	90.9
27	5	3.8	3.8	94.7
28	4	3.0	3.0	97.7
32	1	.8	.8	98.5
38	1	.8	.8	99.2
39	1	.8	.8	100.0
Total	132	100	100	

Berdasarkan tabel 4.1, kesimpulan yang didapatkan adalah responden yang berpartisipasi didominasi oleh responden berusia 21 tahun yaitu sebesar 27.3%, dilanjutkan responden berusia 20 tahun sebesar 20,5 % , serta dilanjutkan responden berusia 22 tahun sebesar 15.9% dan kelompok umur lain yang mendapatkan hasil di bawah 10%.

c. Hasil Pertanyaan Nomor Empat (4)

Dari 132 tanggapan yang didapatkan, terdapat berbagai macam jenis pekerjaan yang di dapatkan dari responden. Frekuensi jawaban responden disajikan pada tabel 4.2.

**Tabel 4.2** Frekuensi Jawaban Pertanyaan Nomor Empat (4)

Pekerjaan	Frekuensi	Presentase	Valid Presentase	Presentase Komulatif
Bidan	1	.8	.8	.8
Freelance	1	.8	.8	1.5
Fresh graduate	1	.8	.8	2.3
Jobseeker	1	.8	.8	3.0
Karyawan	16	12.1	12.1	15.2
Kuli ketik	1	.8	.8	15.9
Mahasiswa	97	73.5	73.5	89.4

**Tabel 4.3 Lanjutan Frekuensi Jawaban Pertanyaan Nomor Empat (4)**

Pekerjaan	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Pelajar	4	3.0	3.0	92.4
Perawat	2	1.5	1.5	93.9
Serabutan	1	.8	.8	94.7
Wiraswasta	7	5.3	5.3	100.0
Total	100	100	100	

Berdasarkan tabel 4.2, kesimpulan yang didapatkan adalah responden yang berpartisipasi didominasi oleh responden yang berprofesi sebagai mahasiswa dengan presentase sebesar 73.5%, dilanjutkan dengan responden berprofesi sebagai karyawan dengan presentase sebesar 12,5 % dan profesi lainnya yang mendapatkan hasil di bawah 10%.

d. Hasil Pertanyaan Nomor Lima (5)

Dari 132 tanggapan, 132 responden memilih opsi ya sedangkan memilih tidak sebanyak 0 responden. Sehingga hasil presentasenya adalah:

$$\text{Jawaban ya} = (132/132)*100\% = 100\%$$

$$\text{Jawaban tidak} = (0/132)*100\% = 0\%$$

Dari presentase tersebut, dapat disimpulkan bahwa responden yang berpartisipasi adalah responden yang menggunakan internet selama satu tahun terakhir.

e. Hasil Pertanyaan Nomor Enam (6)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis komentar jahat, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.4.

**Tabel 4.4 Frekuensi Jawaban Pertanyaan Nomor Enam (6)**

Opsi	Frekuensi	Presentase	Presentase valid	Presentase Komulatif
Tidak Pernah	2	1.5	1.5	1.5
Sekali	3	2.3	2.3	3.8
Beberapa Kali	39	29.5	29.5	33.3
Sering	88	66.7	66.7	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.4, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 66.7%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 29.5%, opsi jawaban sekali sebesar 2.3% dan opsi jawaban tidak pernah sebesar 1.5%.

f. Hasil Pertanyaan Nomor Tujuh (7)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis foto, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.5.

**Tabel 4.5** Frekuensi Jawaban Pertanyaan Nomor Tujuh

Opsi	Frekuensi	Presentase	Valid Presentase	Presentase Komulatif
Tidak Pernah	4	3.0	3.0	3.0
Sekali	19	14.4	14.4	17.4
Beberapa Kali	77	58.3	58.3	75.8
Sering	32	24.2	24.2	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.5, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 58.3%, dilanjutkan dengan opsi jawaban sering sebesar 24.2%, opsi jawaban sekali sebesar 14.4% dan opsi jawaban tidak pernah sebesar 3.0%.

g. Hasil Pertanyaan Nomor Delapan (8)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis video, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.6..

**Tabel 4.6** Frekuensi Jawaban Pertanyaan Nomor Delapan

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	12	9.1	9.1	9.1
Sekali	16	12.1	12.1	21.2
Beberapa Kali	63	47.7	47.7	68.9
Sering	41	31.1	31.1	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.6, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 47.7%, dilanjutkan dengan opsi jawaban sering sebesar 31.1%, opsi jawaban sekali sebesar 12.1% dan opsi jawaban tidak pernah sebesar 9.1%.

#### h. Hasil Pertanyaan Nomor Sembilan (9)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis gosip, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.7.

**Tabel 4.7** Frekuensi Jawaban Pertanyaan Nomor Sembilan

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	4	3.0	3.0	3.0
Sekali	8	6.1	6.1	9.1
Beberapa Kali	34	25.8	25.8	34.8
Sering	86	65.2	65.2	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.7, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 65.2%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 25.8%, opsi jawaban sekali sebesar 6.1% dan opsi jawaban tidak pernah sebesar 3.0%.

#### i. Hasil Pertanyaan Nomor Sepuluh (10)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis pengancaman, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.8.

**Tabel 4.8** Frekuensi Jawaban Pertanyaan Nomor Sepuluh

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	20	15.2	15.2	15.2
Sekali	43	32.6	32.6	47.7
Beberapa Kali	51	38.6	38.6	86.4
Sering	18	13.6	13.6	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.8, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 38.6%, dilanjutkan dengan opsi jawaban sekali sebesar 32.6%, opsi jawaban sering sebesar 13.6% dan opsi jawaban tidak pernah sebesar 15.2%.

j. Hasil Pertanyaan Nomor Sebelas (11)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis berpura-pura menjadi orang lain, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.9.

**Tabel 4.9** Frekuensi Jawaban Pertanyaan Nomor Sebelas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	11	8.3	8.3	8.3
Sekali	19	14.4	14.4	22.7
Beberapa Kali	46	34.8	34.8	57.6
Sering	56	42.4	42.4	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.9, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 42.4%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 34.8%, opsi jawaban sekali sebesar 14.4% dan opsi jawaban tidak pernah sebesar 8.3%.

k. Hasil Pertanyaan Nomor Dua Belas (12)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis komentar tentang ras dan warna kulit, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil yang disajikan pada tabel 4.10.

**Tabel 4.10** Frekuensi Jawaban Pertanyaan Nomor Dua Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	6	4.5	4.5	4.5
Sekali	14	10.6	10.6	15.2
Beberapa Kali	56	42.4	42.4	57.6
Sering	56	42.4	42.4	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.10, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering atau beberapa kali dengan presentase sama besar yaitu 42.4%, dilanjutkan dengan opsi jawaban sekali sebesar 10.6% dan opsi jawaban tidak pernah sebesar 4.5%.

1. Hasil Pertanyaan Nomor Tiga Belas (13)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis komentar bermakna seksualitas, frekuensi jawaban setiap opsi disajikan pada tabel 4.11.

**Tabel 4.11** Frekuensi Jawaban Pertanyaan Nomor Tiga Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	11	8.3	8.3	8.3
Sekali	23	17.4	17.4	25.8
Beberapa Kali	51	38.6	38.6	64.4
Sering	47	35.6	35.6	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.11, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 38.5%, dilanjutkan dengan opsi jawaban sering sebesar 35.6%, opsi jawaban sekali sebesar 17.4% dan tidak pernah sebesar 8.3%.

m. Hasil Pertanyaan Nomor Empat Belas (14)

Dari 132 tanggapan pada pertanyaan *cyberbullying* dengan jenis komentar tentang agama, dari empat opsi jawaban dengan *skala linkert*, di dapatkan hasil dengan presentase yang berbeda. Frekuensi jawaban tiap opsi disajikan pada tabel 4.12.

**Tabel 4.12** Frekuensi Jawaban Pertanyaan Nomor Empat Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	11	8.3	8.3	8.3
Sekali	12	9.1	9.1	17.4
Beberapa Kali	44	33.3	33.3	50.8
Sering	65	49.2	49.2	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.12, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar

49.2%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 33.3%, opsi jawaban sekali sebesar 9.1% dan opsi jawaban tidak pernah sebesar 8.3%.

n. Hasil Pertanyaan Nomor Lima Belas (15)

Dari 132 tanggapan tentang pengalaman mengalami *cyberbullying* secara pribadi, dari empat opsi jawaban, frekuensi jawaban setiap opsi disajikan pada tabel 4.13.

**Tabel 4.13** Frekuensi Jawaban Pertanyaan Nomor Lima Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	77	58.3	58.3	58.3
Sekali	31	23.5	23.5	81.8
Beberapa Kali	16	12.1	12.1	93.9
Sering	8	6.1	6.1	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.13, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban tidak pernah dengan presentase sebesar 58.3%, dilanjutkan dengan opsi jawaban sekali sebesar 23.5%, opsi jawaban beberapa kali sebesar 12.1% dan opsi jawaban sering sebesar 6.1%.

o. Hasil Pertanyaan Nomor Enam Belas (16)

Dari 132 tanggapan tentang media yang sering dijumpai tindakan *cyberbullying*, frekuensi jawaban setiap opsi pada pilihan *chat room* disajikan pada tabel 4.14.

**Tabel 4.14** Frekuensi Jawaban Pertanyaan Nomor Enam Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	38	28.8	28.8	28.8
Sekali	34	25.8	25.8	54.5
Beberapa Kali	46	34.8	34.8	89.4
Sering	14	10.6	10.6	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.14, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 34.8%, dilanjutkan dengan opsi jawaban tidak pernah sebesar 28.8%, opsi jawaban tidak pernah sebesar 25.8% dan opsi jawaban sering sebesar 10.6%.

p. Hasil Pertanyaan Nomor Tujuh Belas (17)

Dari 132 tanggapan tentang media yang sering dijumpai tindakan *cyberbullying*, frekuensi jawaban setiap opsi pada pilihan *instagram* disajikan pada tabel 4.15.

**Tabel 4.15** Frekuensi Jawaban Pertanyaan Nomor Tujuh Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	11	8.3	8.3	8.3
Sekali	21	15.9	15.9	24.2
Beberapa Kali	44	33.3	33.3	57.6
Sering	56	42.4	42.4	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.15, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 42.4%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 33.3%, opsi jawaban sekali sebesar 15.9% dan opsi jawaban tidak pernah sebesar 8.3%.

q. Hasil Pertanyaan Nomor Delapan Belas (18)

Dari 132 tanggapan tentang media yang sering dijumpai tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada pilihan *twitter* disajikan pada tabel 4.16.

**Tabel 4.16** Frekuensi Jawaban Pertanyaan Nomor Delapan Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	47	35.6	35.6	35.6
Sekali	19	14.4	14.4	50.0
Beberapa Kali	39	29.5	29.5	79.5
Sering	27	20.5	20.5	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.16, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban tidak pernah dengan presentase sebesar 35.6%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 29.5%, opsi jawaban sering sebesar 20.5% dan opsi jawaban sekali sebesar 14.4%.

r. Hasil Pertanyaan Nomor Sembilan Belas (19)

Dari 132 tanggapan tentang media yang sering dijumpai tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada pilihan *facebook* disajikan pada tabel 4.17.

**Tabel 4.17** Frekuensi Jawaban Pertanyaan Nomor Sembilan Belas

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	36	27.3	27.3	27.3
Sekali	29	22.0	22.0	49.2
Beberapa Kali	43	32.6	32.6	81.8
Sering	24	18.2	18.2	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.17, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 32.6%, dilanjutkan dengan opsi jawaban tidak pernah sebesar 27.3%, opsi jawaban sekali sebesar 22% dan opsi jawaban sering sebesar 18.2%.

s. Hasil Pertanyaan Nomor Dua Puluh (20)

Dari 132 tanggapan tentang media yang sering dijumpai tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada pilihan media lainnya disajikan pada tabel 4.18.

**Tabel 4.18** Frekuensi Jawaban Pertanyaan Nomor Dua Puluh

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	18	13.6	13.6	13.6
Sekali	35	26.5	26.5	40.2
Beberapa Kali	58	43.9	43.9	84.1
Sering	21	15.9	15.9	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.18, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 43.9%, dilanjutkan dengan opsi jawaban sekali sebesar 26.5%, opsi jawaban sering sebesar 15.9% dan opsi jawaban tidak pernah sebesar 13.6%.

t. Hasil Pertanyaan Nomor Dua Puluh Satu (21)

Dari 132 tanggapan tentang *public figure* yang sering menerima tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada *figure* Deddy Corbuzier disajikan pada tabel 4.19.

**Tabel 4.19** Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Satu

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	46	34.8	34.8	34.8
Sekali	46	34.8	34.8	69.7
Beberapa Kali	35	26.5	26.5	96.2
Sering	5	3.8	3.8	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.19, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban tidak pernah dan sekali dengan presentase sebesar 34.8%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 26.5% dan opsi jawaban sering sebesar 3.8%.

u. Hasil Pertanyaan Nomor Dua Puluh Dua (22)

Dari 132 tanggapan tentang *public figure* sering menerima tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada *figure* Via vallen disajikan pada tabel 4.20.

**Tabel 4.20** Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Dua

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	37	28.0	28.0	28.0
Sekali	35	26.5	26.5	54.5
Beberapa Kali	46	34.8	34.8	89.4
Sering	14	10.6	10.6	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.20, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban beberapa kali dengan presentase sebesar 34.8%, dilanjutkan dengan opsi jawaban tidak pernah sebesar 28.0%, opsi jawaban sekali sebesar 26.5% dan opsi jawaban sering sebesar 10%.

v. Hasil Pertanyaan Nomor Dua Puluh Tiga (23)

Dari 132 tanggapan tentang *public figure* yang sering menerima tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada *figure* Atta halilintar disajikan pada tabel 4.21.

**Tabel 4.21** Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Tiga

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	25	18.9	18.9	18.9
Sekali	31	23.5	23.5	42.4
Beberapa Kali	28	21.2	21.2	63.6
Sering	48	36.4	36.4	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.21, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 36.4%, dilanjutkan dengan opsi jawaban sekali sebesar 23.5%, opsi jawaban beberapa kali sebesar 21.2% dan opsi jawaban tidak pernah sebesar 18.9%.

w. Hasil Pertanyaan Nomor Dua Puluh Empat (24)

Dari 132 tanggapan tentang *public figure* yang sering menerima tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada *figure* Samuel alexander pieterdisajikan pada tabel 4.22.

**Tabel 4.22** Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Empat

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	26	19.7	19.7	19.7
Sekali	15	11.4	11.4	31.1
Beberapa Kali	33	25.0	25.0	56.1
Sering	58	43.9	43.9	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.22, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 43.9%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 25%, opsi jawaban tidak pernah sebesar 19.7% dan opsi jawaban sekali sebesar 11.4%.

x. Hasil Pertanyaan Nomor Dua Puluh Lima (25)

Dari 132 tanggapan tentang *public figure* yang sering menerima tindakan *cyberbullying*, frekuensi jawaban tiap opsi pada Karin novilda disajikan pada tabel 4.23.

**Tabel 4.23** Frekuensi Jawaban Pertanyaan Nomor Dua Puluh Lima

Opsi	Frekuensi	Presentase	Valid presentase	Presentase komulatif
Tidak Pernah	20	15.2	15.2	15.2
Sekali	21	15.9	15.9	31.1
Beberapa Kali	43	32.6	32.6	63.6
Sering	48	36.4	36.4	100.0
Total	132	100.0	100.0	

Berdasarkan tabel 4.23, dapat disimpulkan bahwa responden yang berpartisipasi di dominasi dengan responden yang memilih opsi jawaban sering dengan presentase sebesar 36.4%, dilanjutkan dengan opsi jawaban beberapa kali sebesar 32.6%, opsi jawaban sekali sebesar 15.9% dan opsi jawaban tidak pernah sebesar 15.2%.

#### y. Hasil Rekapitulasi

Dari kuesioner tersebut, pertanyaan di kelompokan menjadi tiga yaitu pertanyaan mengenai pengalaman, media dan korban. Untuk kelompok jenis pengalaman, pertanyaan yang masuk dalam kategori ini adalah pertanyaan nomor enam sampai dengan lima belas. Hasil dari pertanyaan tersebut kemudian dianalisis secara *diskriptif* yang disajikan dalam tabel 4.24.

**Tabel 4.24** Hasil Rekap Kategori Jenis Komentar Jahat

Pertanyaan	N	Minimum	Maximum	Rata-rata	Standart deviasi
Komentar Jahat	132	0	3	2.61	.613
Foto	132	0	3	2.04	.714
Video	132	0	3	2.01	.895
Gosip	132	0	3	2.53	.746
Pengancaman	132	0	3	1.51	.912
Berpura menjadi orang lain	132	0	3	2.11	.946
Komentar tentang ras atau warna kulit	132	0	3	2.23	.816
Komentar dengan makna seksualitas	132	0	3	2.02	.933
Pernah Mengalami	132	0	3	.66	.915
Komentar tentang agama	132	0	3	2.23	.932

Berdasarkan hasil analisis deskriptif yang dilakukan, komentar jahat merupakan jenis *cyberbullying* yang mendapatkan rata-rata tertinggi dari responden dengan nilai sebesar 2.61 dan tindakan *cyberbullying* yang dialami secara personal mendapatkan rata-rata terendah sebesar 0.66.

Kemudian dari kelompok media sosial, dari pertanyaan nomor enam belas sampai nomor dua puluh dilakukan analisis deskriptif yang disajikan disajikan dalam tabel 4.25.

**Tabel 4.25** Hasil Rekap Kategori Media Sosial

Pertanyaan	N	Minimum	Maximum	Rata-rata	Standart deviasi
Chat room	132	0	3	1.27	.997
Instagram	132	0	3	2.10	.956
Twitter	132	0	3	1.35	1.166
Facebook	132	0	3	1.42	1.077
Media sosial lainnya	132	0	3	1.62	.912

Dari analisis deskriptif tersebut, didapatkan hasil bahwa instagram merupakan media sosial yang mendapatkan rata-rata tertinggi sebesar 2.10 sedangkan ruang obrolan merupakan media sosial yang mendapatkan rata-rata terendah yaitu sebesar 1.27.

Kemudian pada jenis kelompok korban, dari pertanyaan nomor dua puluh satu sampai dua puluh lima dilakukan analisis deskriptif yang disajikan pada tabel 4.26.

**Tabel 4.26** Hasil Rekap Kategori Korban

Pertanyaan	N	Minimum	Maximum	Rata-rata	Standart deviasi
Deddy Corbuzier	132	0	3	.99	.878
Via Vallen	132	0	3	1.28	.991
Atta Halilintar	132	0	3	1.75	1.142
Samuel Alexander Pieter	132	0	3	1.93	1.160
Karin Novilda	132	0	3	1.90	1.062

Dari hasil analisis tersebut, Samuel alexander pieter mendapatkan hasil rata-rata tertinggi sebesar 1.93 sedangkan deddy corbuzier mendapatkan rata-rata terendah sebesar 0.99.

Berdasarkan hasil analisis pada tabel 4.25, instagram akan digunakan sebagai sumber data pada penelitian ini, hal ini dikarenakan rata-rata yang didapatkan tertinggi dibandingkan opsi lainnya yang menunjukkan media ini cukup sering ditemukan tindakan *cyberbullying*. Sedangkan berdasarkan tabel 4.26, Samuel alexander pieter dan karin novilda akan dipilih sebagai studi kasus karena mendapatkan hasil rata-rata tertinggi sekaligus mewakili tindakan *cyberbullying* yang diterima oleh laki-laki serta perempuan.

## 4.2. Implementasi

Hasil penelitian merupakan penerapan metode *Long Short Term Memory* (LSTM) untuk melakukan identifikasi *cyberbullying* terhadap komentar pada halaman kirim Instagram. Tahap implementasi dibagi menjadi tiga bagian yaitu implementasi pembuatan model, implementasi pengujian model dan implementasi sistem. Setiap bagian disusun berdasarkan perancangan dengan cakupan meliputi antarmuka dan model program pendukung.

### 4.2.1. Implementasi Pembuatan Model

Pada tahap implementasi pembuatan model, terdapat beberapa tahapan yang dilakukan sebelum pembentukan model sistem seperti koneksi *database*, *text preprocessing*, *word embedding* dengan *word2vec* serta pembentukan model LSTM. Langkah-langkah tersebut dijelaskan sebagai berikut:

#### a. Koneksi *Database*

Koneksi *database* merupakan mekanisme untuk menghubungkan sistem dengan *database server* yang digunakan. Dalam sistem ini *database* yang digunakan adalah sqlite, database yang tersimpan merupakan database yang berisi hasil *scraping* dan pelabelan yang telah dilakukan. Koneksi *database* disajikan dalam modul program 4.1.

```
app = Flask(__name__, instance_relative_config=True)
app.config.from_mapping(
    SECRET_KEY='dev',
    DATABASE=os.path.join(app.instance_path, 'flaskr.sqlite'),
)
```

#### Modul Program 4.1 Source Code Koneksi Database

Modul program 4.1 merupakan *source code* untuk menghubungkan *database* dengan *framework* pada sistem. *App* merupakan inisiasi variabel dari *framework* yang digunakan, serta fungsi *from\_mapping* digunakan untuk melakukan *mapping* terhadap *database* yang digunakan.

b. *Text Preprocessing*

Proses *text preprocessing* merupakan sebuah proses yang dilakukan untuk mendapatkan data teks yang minim dari *noise* sehingga dapat memberikan hasil optimal dalam pembentukan model. Secara umum proses *text preprocessing* dapat dilihat pada modul program 4.2.

```
def preprocessing(self,data):
    self.data = str(data)
    self.data = self.data.lower()
    self.data = self.remove_newline()
    self.data = self.remove_username()
    self.data = self.remove_url()
    self.data = self.remove_non_alphabet()
    self.data = self.remove_excessive_whitespace()
    self.data = self.repeating_char()
    self.data = self.spelling_correction()
    self.data = self.stopword_removal()
    self.data = self.stemming()
    return self.data
```

**Modul Program 4.2 Source Code Text Preprocessing**

Pada modul 4.2, tahapan *text preprocessing* yang dijalankan antara lain adalah *case folding*, *cleansing*, *spelling correction*, *stopword removal* dan *stemming*. Tahapan pertama dalam *case folding* adalah perubahan huruf kapital menjadi huruf kecil yang kemudian dilanjutkan dengan tahapan *cleansing*, proses ini dimulai dengan penghapusan baris baru (*newline*) dalam data. Proses penghapusan baris baru disajikan dalam modul 4.3.

```
def remove_newline(text):
    return re.sub('\n', ' ', text)
```

**Modul Program 4.3 Source Code Remove Newline**

Proses kemudian dilanjutkan dengan penghapusan *username* dalam data. Proses ini disajikan dalam modul program 4.4.

```
def remove_username(text):
    text = re.sub(r'@[A-Za-z0-9]+', ' ', text)
    return text
```

**Modul Program 4.4 Source Code Remove Username**

Proses kemudian dilanjutkan dengan penghapusan *url* atau alamat sebuah website. Proses ini ditunjukkan oleh modul program 4.5.

```
def remove_url(text):
    return re.sub('((www\.[^\s]+)|([https?://[^\s]+])|([http?://[^\s]+))',
                 ' ', text)
```

#### **Modul Program 4.5 Source Code Remove Url**

Selanjutnya dilanjutkan dengan proses penghapusan seluruh karakter selain huruf alphabet.

Fungsi ini ditunjukkan dalam modul program 4.6.

```
def remove_non_alphabet(text):
    output = re.sub('[^a-zA-Z ]+', ' ', text)
    return output
```

#### **Modul Program 4.6 Source Code Remove Non Alphabet**

Selanjutnya adalah proses penghapusan *whitespace*, proses ini berfungsi untuk menghapus karakter spasi berlebih dalam data. Fungsi ini ditunjukkan dalam modul program 4.7.

```
def remove_whitespace(text):
    return re.sub(' +', ' ', text)
```

#### **Modul Program 4.7 Source Code Remove Whitespace**

Selanjutnya adalah proses penghapusan karakter huruf dengan pengulangan huruf lebih dari dua kali dalam sebuah kata. Proses ditunjukkan dalam modul program 4.8.

```
def repeating_char(text):
    return ''.join([''.join(s)[:-2] for _, s in itertools.groupby(text)])
```

#### **Modul Program 4.8 Source Code Remove Repeating Char**

Tahapan selanjutnya dalam *text preprocessing* adalah *spelling correction*, pada tahapan ini kata yang memiliki kesalahan penulisan, ambiguitas, kata bermakna serupa akan disesuaikan dengan kamus yang telah disusun. Proses *spelling correction* ini dapat dilihat pada modul program 4.9.

```
def spelling_correction(text):
    word_list = text.split()
    word_list_len = len(word_list)
    transformed_word_list = []
    i = 0
    while i < word_list_len:
        if (i + 1) < word_list_len:
            two_words = ''.join(word_list[i:i+2])
            if two_words in normal_dict:
                transformed_word_list.append(normal_dict[two_words])
                i += 2
            continue
        transformed_word_list.append(normal_dict.get(word_list[i],
word_list[i]))
        i += 1
    return ' '.join(transformed_word_list)
```

#### **Modul Program 4.9 Source Code Normalisasi Kata**

Proses tersebut kemudian di lanjutkan dengan *stopword removal*. Proses ini bertujuan untuk menghapus kata-kata dengan tingkat kemunculan yang berlebihan serta kata-kata yang kurang berpengaruh dalam kasus sentimen. *Stopword* yang digunakan mengacu pada *stopword* bahasa indonesia (Tala, 2003). Proses *stopword removal* disajikan dalam modul program 4.10.

```
def stopword_removal(self):
    konjungsi = ["ya", "tidak"]
    stopword = [line.strip('\n') for line in open('stopword.txt')]
    word_list = self.data.split()
    text = []
    for i in word_list:
        if i not in stopword:
            if i in konjungsi:
                text.append(i)
            else:
                if len(i) > 2:
                    text.append(i)
    return ' '.join(text)
```

#### **Modul Program 4.10 Source Code Stopword Removal**

Proses selanjutnya adalah *stemming*, data yang telah mengalami proses pada tahapan sebelumnya akan diubah ke dalam bentuk kata dasar. Pada proses ini terjadi penghapusan kata awalan, imbuhan dan sejenisnya. Proses *stemming* yang digunakan menggunakan library sastrawi yang mengimplementasian algoritma nazief dan adriani. Proses ini dapat dilihat dalam modul 4.11.

```
def stemming(text):
    word_list = text.split()
    text = ' '.join([i for i in word_list if i not in stopword])
    print(text)
    factory=StemmerFactory()
    stemmer=factory.create_stemmer()
    output=stemmer.stem(text)
    return output
```

#### **Modul Program 4.11 Source Code Stemming**

##### c. *Tokenisasi*

Tahapan *tokenisasi* dilakukan untuk membentuk kamus data. Data berupa kalimat yang telah mengalami proses *text preprocessing* akan dilakukan pemecahan tiap kata serta dilanjutkan penggabungan sesuai dengan kata yang sama, kemunculan kata akan

berpengaruh terhadap *index* kata dalam kamus. Mekanisme *tokenisasi* di tunjukkan dalam modul program 4.12.

```
tokenizer = Tokenizer(num_words=self.max_features, split=' ')
tokenizer.fit_on_texts(self.train_data['after_prepro'].values)
```

#### **Modul Program 4.12 Source Code Tokenisasi**

Berdasarkan modul program 4.12, dengan implementasi fungsi *Tokenizer* akan dilakukan proses *spliting* kalimat dengan *separator* yang telah ditentukan sekaligus pendefinisian ukuran kamus data (*vocabulary*). Proses tersebut kemudian dilanjutkan dengan proses pengelompokan kata serupa yang akan membentuk kamus kata yang berisi kata serta *index* urutan kata.

d. *convert* data menjadi *sequence*

Kamus data yang telah terbentuk pada tahapan sebelumnya kemudian akan menjadi dasar dalam perubahan komentar menjadi angka sesuai dengan indeks kata dalam *vocabulary*, proses tersebut kemudian dilanjutkan dengan proses *padding* yang bertujuan untuk menyamakan panjang *sequence* dan kemudian dilanjutkan dengan penyimpanan hasil *tokenizer* ke dalam file berekstensi *pickle*. Mekanisme ini disajikan dalam modul program 4.13.

```
X = tokenizer.texts_to_sequences(self.train_data['after_prep'].values)
X = pad_sequences(X, maxlen=50)
Y = pd.get_dummies(self.train_data['score']).values
with open(self.basepath_token, 'wb') as handle:
    pickle.dump(self.tokenizer, handle,
                protocol=pickle.HIGHEST_PROTOCOL)
self.vocab_size = len(self.tokenizer.word_index)+1
```

#### **Modul Program 4.13 Source Code Convert data ke Sequence**

e. Pembagian data latih dan data uji

Proses selanjutnya adalah pembagian antara data latih dan data uji. Proses ini dilakukan untuk membagi data yang akan digunakan sebagai pembentukan model LSTM dan data yang akan digunakan untuk menguji model LSTM. Pembagian data yang digunakan

yaitu sebesar 90% sebagai data latih dan 10% data uji. Proses pembagian data latih dan data uji disajikan dalam modul program 4.14.

```
X_train, X_test, Y_train, Y_test = train_test_split(X, Y,
test_size=0.10, random_state=32, shuffle=True)
```

**Modul Program 4.14 Source Code Pembagian Data Latih dan Data Uji**

f. *Word Embedding*

Data yang telah mengalami tahapan *preprocessing* kemudian akan ditambahkan ke dalam korpus *word2vec*, dalam penelitian ini korpus yang digunakan berasal dari Wikipedia (407.034 artikel sampai dengan 17 september 2020), penggunaan korpus ini agar mendapatkan representasi kata yang lebih baik. Dalam membangun *pretrained word2vec*, program dijalankan secara eksternal melalui *google colaboratory* untuk mempercepat waktu pemrosesan. *Library gensim* juga digunakan dalam proses pembentukan *pretrained word2vec*. Proses ekstraksi korpus ke dalam *file txt* dijelaskan dalam modul program 4.15.

```
if __name__ == '__main__':
    start_time = time.time()
    print('Streaming wiki...')
    id_wiki = gensim.corpora.WikiCorpus(
        '/root/input/idwiki-latest-pages-articles.xml.bz2',
        lemmatize=False, dictionary={}, lower=True
    )
    article_count = 0
    with io.open('idwiki_new_lower.txt', 'w', encoding='utf-
8') as wiki_txt:
        for text in id_wiki.get_texts():
            wiki_txt.write(" ".join(text) + '\n')
            article_count += 1
            if article_count % 10000 == 0:
                print('{} articles processed'.format(article_count))
                print('total: {} articles'.format(article_count))
    finish_time = time.time()
    print('Elapsed time: {}'.format(timedelta(seconds=finish_time-
start_time)))
```

**Modul Program 4.15 Source Code Ekstraksi Corpus Word2Vec**

Dalam modul program 4.15 tersebut, dengan menggunakan fungsi *WikiCorpus* program akan mengekstrak dan memproses *corpus* dengan *preprocessing* sederhana.

Lematisasi yang dijalankan bernilai *false* dengan tujuan agar proses tidak di perlambat dengan adanya proses lematisasi dimana proses ini hanya berjalan dalam bahasa Inggris. Proses ekstraksi yang berlangsung akan mengekstrak *corpus* menjadi sebuah *item* kata yang disimpan dalam sebuah *dictionary*. Setelah proses tersebut dilakukan, kemudian akan dilanjutkan dengan proses *training* corpus yang ditunjukkan pada modul program 4.16.

```
start_time = time.time()
print('Training Word2Vec Model...')
sentences = word2vec.LineSentence('idwiki_new_lower.txt')
id_w2v = word2vec.Word2Vec(sentences, size=128, workers=multiprocessing
    .cpu_count()-1)
id_w2v.save('/root/input/idwiki_word2vec_128_new_lower.model')
finish_time = time.time()
```

#### **Modul Program 4.6 Source Code Training Word2Vec**

Dalam modul 4.16, variabel *sentences* akan menjadi variabel penampung untuk menyimpan kalimat. Fungsi *LineSentence* menunjukkan bahwa proses *word2vec* akan dilakukan dengan membaca setiap baris pada *file* berekstensi *txt* secara lebih efisien. Proses *training* yang berlangsung akan menghasilkan representasi vector pada setiap *token* kata. Jumlah dimensi atau *hidden unit* yang akan digunakan pada mekanisme *word2vec* sejumlah 128. *Pretrained* yang telah terbentuk kemudian akan disimpan menggunakan fungsi *save*. Proses selanjutnya adalah menambahkan data latih yang telah melalui tahapan *text preprocessing* ke dalam *pretrained* untuk dilakukan *training*, proses ini diawali dengan *tokenisasi* data yang ditunjukkan pada modul program 4.17.

```
import codecs
import nltk
nltk.download('punkt')
from nltk.tokenize import word_tokenize
with codecs.open('dataset.csv', 'r')as f:
    for line in f:
        tweet = f.readlines()
        tokenized_sent = [word_tokenize(i) for i in tweet]
        print("Success")
id_w2v.build_vocab(tokenized_sent, update=True)
id_w2v.train(tokenized_sent, epochs=10, total_examples=len(tokenized_sent
))
```

#### **Modul Program 4.17 Source Code Update Pretrained Word2Vec**

Dalam modul 4.17, proses pembaharuan *pretrained* dimulai dengan ekstraksi dataset yang telah mengalami *text preprocessing* ke dalam sebuah *list*. Proses tersebut kemudian dilanjutkan dengan proses *training* hasil ekstraksi terhadap *pretrained* yang telah terbentuk sebelumnya. Selanjutnya adalah proses penyimpanan bobot *fitur* kata tiap *token* pada *pretrained* yang terbentuk ke dalam *file* berekstensi *txt*. Proses penyimpanan bobot ditunjukkan pada modul 4.18.

```
from numpy import asarray
embeddings_index = dict()
f = open('/root/input/word2vec.txt')
for line in f:
    values = line.split()
    word = values[0]
```

#### **Modul Program 4.18 Source Code Penyimpanan Pretrained Word2Vec**

```
coefs = asarray(values[1:], dtype='float32')
embeddings_index[word] = coefs
f.close()
print('Loaded %s word vectors.' % len(embeddings_index))
```

#### **Modul Program 4.19 Lanjutan Source Code Penyimpanan Pretrained Word2Vec**

Selanjutnya adalah proses pengambilan data sesuai dengan *dataset* yang akan digunakan dalam pembuatan model LSTM. Proses ini berlangsung dengan menyamakan kata tiap *token* pada *dataset* dengan *pretrained* yang telah tersimpan, proses ini ditunjukkan dalam modul program 4.20.

```
from numpy import zeros
embedding_matrix = zeros((vocab_size, 128))
for word, i in tokenizer.word_index.items():
    embedding_vector = embeddings_index.get(word)
    if embedding_vector is not None:
        embedding_matrix[i] = embedding_vector
```

#### **Modul Program 4.20 Source Code Penyimpanan**

Tahapan terakhir dalam mekanisme *word2vec* adalah menyimpan hasil representasi *token* tiap kata untuk digunakan dalam proses *training* model LSTM. Pada proses penyimpanan ini, data disimpan menggunakan *library numpy* ke dalam file berekstensi *npy*. Mekanisme penyimpanan ini ditunjukkan dalam modul program 4.21.

```
import numpy as np
np.save('/root/input/embedding_matrix.npy', embedding_matrix)
```

**Modul Program 4.21 Source Code Penyimpanan Embedding**

g. Pembentukan Model

Dalam pembentukan model, terdapat beberapa *hyperparameter* yang digunakan.

*Hyperparameter* yang digunakan dalam pembentukan model disajikan dalam tabel 4.27:

**Tabel 4.27 Hyperparameter Model**

No	Hyperaprameter	Nama	Nilai
1	Jaringan	<i>epoch</i>	100
		<i>activation</i>	softmax
		<i>hidden unit</i>	128
		optimasi	adam, L1L2 regularizer
		<i>dropout</i>	0.3
		n-layer	4
2	Arsitektur	loss	categorical cross-entropy

Proses pembentukan model algoritma menggunakan *library tensorflow*, proses pembentukan model dapat dilihat pada modul program 4.22.

```
def make_model(self):
    embedding_matrix=np.load(self.basepath_matrix)
    embed_dim = 128
    lstm_out = 64
    model = tensorflow.keras.Sequential()
    model.add(Embedding(vocab_size,embed_dim,weights=
                        [embedding_matrix], input_length=50, trainable=True))
    model.add(SpatialDropout1D(0.25))
    model.add(LSTM(lstm_out, dropout=0.3))
    model.add(Dropout(0.2))
    kernel_regularizer=tf.keras.regularizers.L1L2()
    model.add(Dense(4, activation='softmax',kernel_regularizer=
                    kernel_regularizer))
    optimizer = tf.keras.optimizers.Adam()
    model.compile(loss='categorical_crossentropy',
                  optimizer=optimizer, metrics=['accuracy'])
```

**Modul Program 4.22 Source Code Pembentukan Model**

Berdasarkan modul program 4.22, Proses pembentukan jaringan di mulai dengan inisiasi *model* yang kemudian dilanjutkan dengan penambahan layer ke dalam model. *Layer* pertama dalam jaringan yang akan dibangun adalah *layer embedding*, pada *layer* ini akan di definisikan karakteristik *input* ke dalam jaringan yang dimulai dengan inisiasi jumlah kamus data, jumlah dimensi *input*, jumlah panjang maksimal tiap *input* serta pemuatan bobot tiap

fitur sesuai dengan hasil *word2vec*. Pada layer ini juga akan terjadi mekanisme *embedding lookup* yang merupakan perubahan *sequence* angka menjadi *sequence vector* sesuai dengan gambar 3.45. Setelah layer tersebut, dilakukan penambahan *layer SpatialDropout1D*, *layer* ini memiliki fungsi untuk memetakan elemen fitur ke dalam bentuk satu dimensi sehingga bukan merupakan sebuah elemen individual. Proses dilanjutkan dengan penambahan *layer LSTM*, pada *layer* ini akan terjadi mekanisme perkalian matriks antara *input* dan bobot yang berbeda pada tiap *gate*, proses pada *layer* ini dimulai dengan inisiasi *output* jaringan serta nilai *dropout* yang berfungsi menonaktifkan *neuron* yang berupa *hidden* atau *layer* guna mencegah *overfitting*, penonaktifan *neuron* yang terjadi pada jaringan bersifat sementara. Pada *layer* ini juga terdapat mekanisme *backpropagation* yang berfungsi untuk melakukan perubahan bobot pada tiap *gate* sehingga mendapatkan bobot yang efisien. Proses terakhir dalam pembuatan model LSTM adalah penambahan *dense layer*, *layer* ini memiliki fungsi untuk memadatkan hasil *output* *lstm* menjadi hasil klasifikasi sesuai kelas yang ditentukan, proses pemanjangan terjadi dengan proses perkalian antara *output* dengan bobot pada *layer* serta bias dan klasifikasi berdasarkan fungsi aktivasi *softmax* untuk mendapatkan hasil probabilitas pada kelas yang bervariatif. Model yang telah terbentuk kemudian dilakukan proses *compiling* guna mendapatkan nilai *loss* serta *metrics* yang akan diukur, dalam penelitian ini *metrics* yang akan digunakan berupa *accuracy*. Proses selanjutnya adalah proses *training* terhadap model yang disajikan dalam modul program 4.23.

```
csv_loger=CSVLogger(os.path.join(self.path, "data", "model",
"logger2.csv"), append=True)
history = model.fit(X_train, Y_train, epochs=100,
batch_size=self.batch_size, validation_data=(X_test, Y_test), verbose=1,
callbacks=[csv_loger])
model.save(os.path.join(self.path, "data", "model", "model.h5"))
np.save(self.basepath history, history.history)
```

#### **Modul Program 4.23 Source Code Training Model**

Berdasarkan modul program 4.23, proses *training* yang terjadi akan disimpan dalam bentuk *logger* ke dalam *file* berekstensi *csv*, *logger* yang tersimpan berisi tentang *loss* dan

*accuracy* tiap iterasi pada data yang dilakukan *training*. Model yang terbentuk kemudian disimpan ke dalam *file* berekstensi *h5* untuk digunakan pada proses identifikasi komentar.

Ilustrasi *training* model ditunjukkan dalam gambar 4.1..

```
WARNING:tensorflow:Model was constructed with shape (None, 50) for input Tensor("embedding_22_input:0", shape
WARNING:tensorflow:Model was constructed with shape (None, 50) for input Tensor("embedding_22_input:0", shape
19/20 [=====>..] - ETA: 0s - loss: 1.3469 - accuracy: 0.3549WARNING:tensorflow:Model wa
20/20 [=====] - 1s 49ms/step - loss: 1.3476 - accuracy: 0.3531 - val_loss: 1.2435 -
Epoch 2/100
20/20 [=====] - 1s 25ms/step - loss: 1.2433 - accuracy: 0.4494 - val_loss: 1.1386 -
Epoch 3/100
20/20 [=====] - 1s 25ms/step - loss: 1.1685 - accuracy: 0.5258 - val_loss: 1.0504 -
Epoch 4/100
20/20 [=====] - 1s 25ms/step - loss: 1.1093 - accuracy: 0.5364 - val_loss: 0.9789 -
Epoch 5/100
20/20 [=====] - 0s 25ms/step - loss: 1.0543 - accuracy: 0.5603 - val_loss: 0.9314 -
Epoch 6/100
20/20 [=====] - 0s 25ms/step - loss: 1.0196 - accuracy: 0.5868 - val_loss: 0.9069 -
Epoch 7/100
20/20 [=====] - 1s 25ms/step - loss: 0.9731 - accuracy: 0.6067 - val_loss: 0.8986 -
Epoch 8/100
20/20 [=====] - 0s 25ms/step - loss: 0.9302 - accuracy: 0.6132 - val_loss: 0.8742 -
Epoch 9/100
20/20 [=====] - 0s 25ms/step - loss: 0.9010 - accuracy: 0.6440 - val_loss: 0.8645 -
Epoch 10/100
20/20 [=====] - 1s 26ms/step - loss: 0.8934 - accuracy: 0.6489 - val_loss: 0.8547 -
Epoch 11/100
20/20 [=====] - 0s 24ms/step - loss: 0.8695 - accuracy: 0.6534 - val_loss: 0.8551 -
Epoch 12/100
20/20 [=====] - 1s 25ms/step - loss: 0.8455 - accuracy: 0.6587 - val_loss: 0.8501 -
Epoch 13/100
20/20 [=====] - 1s 25ms/step - loss: 0.8415 - accuracy: 0.6640 - val_loss: 0.8424 -
Epoch 14/100
20/20 [=====] - 1s 25ms/step - loss: 0.7970 - accuracy: 0.6810 - val_loss: 0.8308 -
```

Gambar 4.1 Proses *Training*

*Logger training* model disajikan dalam gambar 4.2.

epoch	accuracy	loss	val_accuracy	val_loss
0	0.485955059528351	1.19858777523041	0.618589758872986	1.06232857704163
1	0.591492772102356	1.03217446804047	0.652243614196777	0.964902520179749
2	0.632022500038147	0.957008838653564	0.665064096450806	0.906644761562347
3	0.654494404792786	0.888416886329651	0.682692289352417	0.880729973316193
4	0.662921369075775	0.870046377182007	0.689102590084076	0.863787412643433
5	0.684590697288513	0.818211853504181	0.690705120563507	0.84581983089447
6	0.689406096935272	0.807720601558685	0.70352566242218	0.826155424118042
7	0.697030484676361	0.786164820194244	0.716346144676209	0.828325212001801
8	0.703049778938293	0.765149056911469	0.709935903549194	0.813686370849609
9	0.723916530609131	0.749628961086273	0.711538434028626	0.816708862781525
10	0.723916530609131	0.740113258361816	0.705128192901611	0.840327739715576
11	0.73515248298645	0.711940169334412	0.711538434028626	0.807364881038666
12	0.730738341808319	0.708395838737488	0.721153855323792	0.808178842067719
13	0.738764047622681	0.697017431259155	0.717948734760284	0.818983256816864
14	0.75	0.667492568492889	0.717948734760284	0.814683854579926
15	0.74638843536377	0.660384237766266	0.725961565971375	0.805818676948547
16	0.74638843536377	0.652844488620758	0.725961565971375	0.814527630805969
17	0.755216717720032	0.640276372432709	0.735576927661896	0.809736311435699
18	0.764446198940277	0.623450458049774	0.732371807098389	0.824348151683807
19	0.774478316307068	0.608948290348053	0.722756385803223	0.824446082115173
20	0.77046549320221	0.602587938308716	0.735576927661896	0.825397372245789
21	0.781701445579529	0.593962848186493	0.727564096450806	0.822291910648346
22	0.790128409862518	0.559894859790802	0.730769217014313	0.851098299026489
23	0.800160527229309	0.555220305919647	0.716346144676209	0.847113788127899
24	0.798956632614136	0.545890867710114	0.713141024112701	0.843978881835938
25	0.798555374145508	0.526551485061646	0.719551265239716	0.866867542266846

Gambar 4.2 *Logger training*

#### 4.2.2. Implementasi Pengujian Model

Pada bagian ini tahap pengujian menggunakan dua mekanisme yaitu pengujian menggunakan *k-fold cross validation* dan perbandingan algortima LSTM dengan *random forest classifier*. Proses pengujian model dijelaskan sebagai berikut:

a. Pengujian *K-Fold Cross Validation*

Pengujian *k-fold cross validation* digunakan untuk mendapatkan hasil terbaik dari pengujian antara data latih dan data uji sehingga seluruh data dalam *dataset* mendapatkan kesempatan untuk menjadi data uji. Untuk penggunaan jumlah *fold* terbaik untuk uji validitas, dianjurkan menggunakan *10-fold cross validation* dalam model (Kohavi, 1995).

Proses pengujian *k-fold cross validation* yang disajikan dalam modul program 4.24.

```
def Kfold_cross(self):
    embedding_matrix = np.load(self.basepath_matrix)
    n_split = 10
    i = 0
    batch_size = 128
    accuracy_model = []
    f1_model = []
    precision_model = []
    recall_model = []
    for train_index, test_index in KFold(n_split,
                                          shuffle=True).split(X):
        i = i + 1
        x_train, x_test = X[train_index], X[test_index]
        y_train, y_test = Y[train_index], Y[test_index]
        model = tf.keras.Sequential()
        model.add(Embedding(vocab_size, 128,
                            weights=[embedding_matrix], input_length=X.shape[1],
                            trainable=True))
        model.add(SpatialDropout1D(0.25))
        model.add(LSTM(64, dropout=0.3))
        model.add(Dropout(0.2))
        model.add(Dense(4, activation='softmax'))
        optimizer = tf.keras.optimizers.Adam()
        model.compile(loss='categorical_crossentropy',
                      optimizer=optimizer, metrics=['accuracy'])
        print(model.summary())
        model.fit(x_train, y_train, epochs=100)
        accuracy_model.append(accuracy_score(y_test, np.round(
            model.predict(x_test)), normalize=True) * 100)
        f1_model.append(f1_score(y_test, np.round(
            model.predict(x_test)), average='weighted') * 100)
        recall_model.append(recall_score(y_test, np.round(
            model.predict(x_test)), average='weighted') * 100)
        precision_model.append(precision_score(y_test, np.round(
            model.predict(x_test)), average='weighted') * 100)
```

**Modul Program 4.24 Source Code K-fold Cross validation**

```

        model.predict(x_test)), average='weighted') * 100)
model.save(os.path.join(self.path, "data", "fold",
    "model{}.h5").format(i))
y_pred = model.predict_classes(x_test,
    batch_size=batch_size)
df_test = pd.DataFrame({'true': y_test.tolist(), 'pred':
    y_pred})
df_test['true'] = df_test['true'].apply(lambda x:
    np.argmax(x))
print("confusion matrix", confusion_matrix(
    df_test.true, df_test.pred))
df_test.to_csv(os.path.join(self.path, "data", "confusion",
    "confusion_matrix{}.csv").format(i))

```

#### **Modul Program 4.25 Lanjutan Source Code K-Fold Cross Validation**

Berdasarkan modul program 4.24, Pengujian *k-fold* akan dilakukan sesuai dengan nilai *fold* yaitu sebesar 10. Beberapa variabel di definisikan sebagai parameter pengujian seperti *accuracy\_model* untuk menampung data hasil akurasi, *f1\_model* untuk menampung data *F1-Score*, *recall\_model* untuk menampung nilai *recall* dan *precision\_model* untuk menampung nilai presisi yang dilakukan setiap *fold* sejumlah 10. Proses pengujian *k-fold* dimulai dengan melakukan pembagian antara data latih dan data uji. Pengujian dilakukan dengan mendefinisikan arsitektur *layer* yaitu *embedding layer*, *SpatialDropout1D*, *lstm layer*, *Dropout layer* dan *dense layer*. Pengujian dilakukan dengan menggunakan *pretrained word2vec* yang telah dilakukan, jumlah dimensi yang digunakan sejumlah 128 dengan jumlah *hidden unit* sebanyak 64. Proses pelatihan menggunakan 100 *epoch*. Setiap *fold* pengujian yang dilakukan akan diverifikasi menggunakan *confusion matrix* dan disimpan ke dalam *file csv*.

#### b. Perbandingan LSTM dan *Random forest classifier*

Pengujian dengan membandingkan dua algoritma bertujuan untuk mendapatkan algoritma terbaik dalam menangani sebuah kasus. Pada perbandingan ini dilakukan perbandingan algoritma antara LSTM dan *random forest classifier*, *random forest classifier* dipilih berdasarkan hasil penelitian sebelumnya yang menghasilkan nilai *F1-Score* sebesar 0.90 (Novalaita, 2018). Proses perbandingan ini menggunakan empat matriks pengujian yaitu akurasi, presisi, *recall* dan *F1-Score*. Data yang digunakan dibagi menjadi 80% sebagai

data latih dan 20% sebagai data uji. Pengujian dilakukan dengan melakukan pembentukan model *random forest classifier* yang ditunjukan pada modul program 4.26.

```
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
accuracy_score,f1_score, precision_score, recall_score

class random_forest:
    def __init__(self,X,Y):
        self._X=X
        self._Y=Y

    def __repr__(self):
        return f"Random forest Algoritm"

    def random_main(self):
        X=self._X
        Y=self._Y
        vectorizer = TfidfVectorizer (max_features=5000, min_df=7,
                                      max_df=0.8)
        processed_features = vectorizer.fit_transform(X).toarray()
        X_train, X_test, y_train, y_test = train_test_split(
            processed_features, Y, test_size=0.2,
            random_state=32,shuffle=True)
        print(y_test)
        text_classifier = RandomForestClassifier(n_estimators=200,
                                                random_state=0)
        text_classifier.fit(X_train, y_train)
        predictions = text_classifier.predict(X_test)
        print(confusion_matrix(y_test,predictions))
        print(classification_report(y_test,predictions))
        return {
            'accuracy_score':accuracy_score(y_test, predictions),
            'precision_score':precision_score(y_test,
                                              predictions,average='weighted'),
            'recall_score':recall_score(y_test,
                                         predictions,average='weighted'),
            'f1_score':f1_score(y_test, predictions,average='weighted')
        }
```

#### **Modul Program 4.26 Source Code Pengujian Random Forest Classifier**

Proses selanjutnya adalah pembuatan model LSTM, model LSTM yang digunakan memiliki arsitektur *layer* berupa *embedding layer*, *SpatialDropout1D*, *lstm layer*, *Dropout layer* dan *dense layer*. *Source code* pembentukan model LSTM ditunjukan dalam modul program 4.27.

```
import numpy as np
import pandas as pd
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from tensorflow.keras.layers import Embedding, Dense, LSTM,
```

#### **Modul Program 4.27 Source Code Pengujian LSTM**



### Modul Program 4.28 Lanjutan Source Code Pengujian LSTM

Hasil matriks *accuracy*, *precisi*, *recall* dan *F1-Score* kedua algoritma kemudian kemudian disajikan dalam sebuah grafik menggunakan *library matplotlib*. *Source code* grafik perbandingan algoritma disajikan dalam modul program 4.29.

```

labels=['accuracy','precision','recall','f1-score']
lstm_value=[int(i*100) for i in lstm_data.values()]
random_value=[int(i*100) for i in random_data.values()]
x=np.arange(len(labels))
width=0.35
fig,ax=plt.subplots()
rects1=ax.bar(x-width/2,lstm_value,width,label='LSTM')
rects2=ax.bar(x+width/2,random_value,width,label='Random Forest Classifier')
ax.set_ylabel('Score')
ax.set_title("Perbandingan LSTM VS Random forest Classifier")
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend(loc='lower left')
def auto_label(rects):
    for rect in rects:
        height=rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 3), textcoords="offset points",
                    ha='center', va='bottom')
auto_label(rects1)
auto_label(rects2)
fig.tight_layout()
plt.show()

```

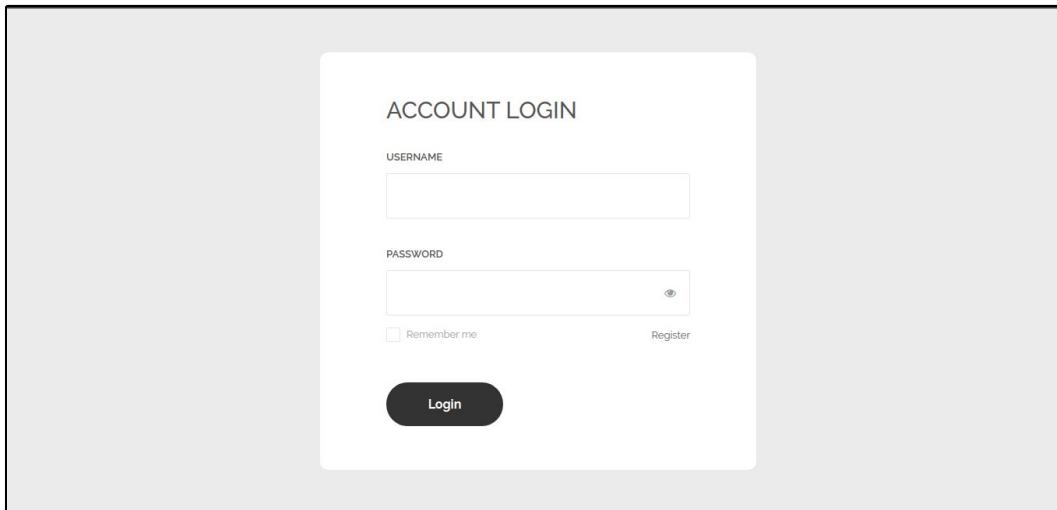
### Modul Program 4.29 Source Code Grafik Perbandingan Algoritma

#### 4.2.3. Implementasi Sistem

Tahap implementasi sistem merupakan tahapan pengimplementasian model yang telah dibentuk ke dalam sistem yang dirancang. Tahapan implementasi model ke dalam sistem dijelaskan sebagai berikut.

##### a. Halaman *Login*

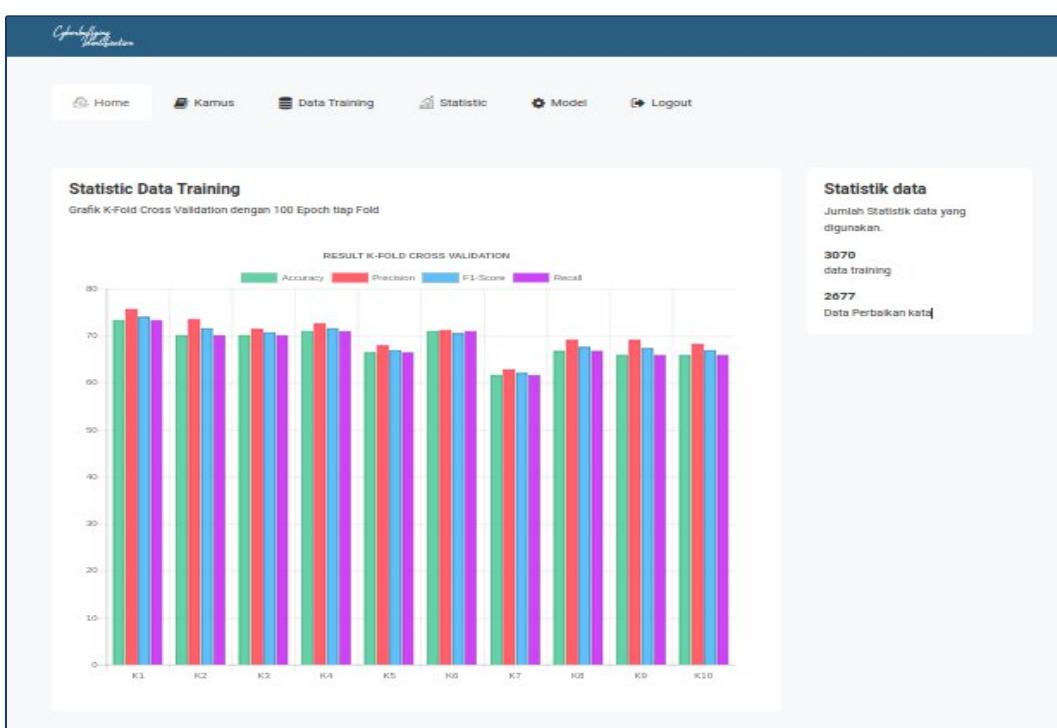
Halaman *login* merupakan halaman awal yang berisi sebuah *application form* dengan *textfield username*, *textfield password* dan sebuah *button*. Halaman *login* perlu dilakukan untuk proses autentikasi bagi *admin*. Tampilan halaman *login* disajikan dalam gambar 4.3.



**Gambar 4.3** Tampilan Halaman *Login*

b. Halaman *Home Admin*

Halaman *home admin* merupakan halaman yang tampil setelah admin sukses melakukan proses *login*. Halaman ini berisi garafik hasil pengujian *k-fold cross validation* yang dilakukan terhadap model. Selain itu, halaman ini juga menampilkan jumlah *dataset* yang digunakan dalam pembentukan model serta jumlah kamus perbaikan kata yang digunakan pada tahap *spelling correction* pada mekanisme *text preprocessing*. Tampilan halaman *home admin* disajikan dalam gambar 4.4.



**Gambar 4.4** Tampilan Halaman *Home Admin*

c. Halaman Perbaikan Kata *Admin*

Halaman perbaikan kata merupakan halaman yang tampil jika admin menekan menu kamus pada *navigation bar*. Halaman ini berisi kamus perbaikan kata yang digunakan pada tahapan *spelling correction*. Pada halaman ini kata perbaikan ditampilkan dalam bentuk tabel, setiap kata memiliki tombol *edit* dan *delete*. Jika admin menekan tombol *delete* maka data akan terhapus dari tabel, sedangkan jika admin menekan *edit* maka data akan ditampilkan pada *application form* di sebelah tabel. Selain itu *application form* di sebelah tabel juga berfungsi untuk menambahkan data pada kamus perbaikan kata. Halaman Proses perbaikan kata disajikan dalam modul program 4.30.

```
@bp.route('/perbaikan_kata', methods=['POST', 'GET'])
@login_required
def perbaikan_kata():
    db = get_db()
    if request.form.get('update') is not None:
        message=None
        data_update=db.execute("SELECT      *      from      typo_2      WHERE
                               id=?",(request.form.get('update'),)).fetchone()
    elif request.form.get('delete') is not None:
        message = "Data Sucessfull Deleted"
        data_update=None
        db.execute("DELETE      FROM      typo_2      WHERE      id=?",
                  (request.form.get('delete'),))
        db.commit()
    elif request.method=='POST':
        data=request.form
        typo = data["typo"]
        correct = data["correct"]
        try:
            id = data["id"]
            db.execute('UPDATE typo_2 SET false=?,true=? WHERE id=?', (typo,
                                                                      correct, id,))
            message = "Successfull Update Data"
        except:
            db.execute('INSERT      INTO      typo_2      (false,true)      VALUES
                       (?,?)',(typo,correct,))
            message="Successfull Adding Data"
        finally:
            data_update=None
            db.commit()
    else:
        message = None
        data_update=None
    data = db.execute("SELECT * from typo_2").fetchall()
    return render_template('admin/kamus.html',data=data,message=message,
                          data_update=data_update)
```

Berdasarkan modul program 4.30, sistem akan menampilkan data perbaikan kata dari *database*, data tersebut di simpan dalam variabel *data* yang kemudian di *render* dengan halaman *html* menggunakan *jinja template engine*. Halaman perbaikan kata admin ditampilkan dalam gambar 4.5.

Kata Awal	kata Perbaikan	Action
aadaa	ada	<button>Edit</button> <button>Delete</button>
ashh	ah	<button>Edit</button> <button>Delete</button>
sa o	saj a	<button>Edit</button> <button>Delete</button>
sa jo	saj o	<button>Edit</button> <button>Delete</button>
as khirmya	akhirnya	<button>Edit</button>

**Form Perbaikan Data**

ambahkan/edit kata kedalam data, data dalam kamus perbaikan data akan digunakan dalam tahapan text preprocessing dalam pembangunan model.

Gambar 4.5 Tampilan Halaman Perbaikan Kata

d. Halaman *Data Training Admin*

Halaman *data training* merupakan halaman yang tampil jika admin menekan menu *data training* pada *navigation bar*. Halaman ini berisi *dataset* yang digunakan pada proses pembentukan model LSTM. Pada halaman ini *data training* ditampilkan dalam bentuk tabel, setiap data ditampilkan berupa data sebelum *preprocessing*, data setelah *preprocessing* dan label pada data tersebut. Setiap komentar pada tabel memiliki tombol *edit* dan *delete*. Jika admin menekan tombol *delete* maka data akan terhapus dari tabel, sedangkan jika admin menekan *edit* maka data akan ditampilkan pada *application form* di sebelah tabel. Selain itu *application form* di sebelah tabel juga berfungsi untuk menambahkan data pada *dataset* yang

kemudian akan dilakukan *training*. Halaman *data training* disajikan dalam modul program

#### 4.31.

```
@bp.route('/data_training', methods=['POST', 'GET'])
@login_required
def data_training():
    db = get_db()
    if request.form.get('update') is not None:
        message = None
        data_update = db.execute("SELECT * from dataset_baru WHERE id=?",
                               (request.form.get('update'),)).fetchone()
    elif request.form.get('delete') is not None:
        message = "Data Sucessfull Deleted"
        data_update = None
        db.execute("DELETE FROM dataset_baru WHERE id=?",
                   (request.form.get('delete'),))
        db.commit()
    elif request.method == 'POST':
        data = request.form
        text1 = data["message"]
        normal = Activate()
        text2=normal.preprocessing(text1)
        score=data["score"]
        try:
            id = data["id"]
            db.execute('UPDATE dataset_baru SET real_comment=?,after_repro=?,score=? WHERE id=?', (text1, text2,score, id,))
            message = "Successfull Update Data"
        except:
            db.execute('INSERT INTO dataset_baru (real_comment,after_repro,score) VALUES (?,?,?)', (text1, text2,score))
            message = "Successfull Adding Data"
        finally:
            data_update = None
            db.commit()
    else:
        message = None
        data_update = None
        data = db.execute("SELECT * from dataset_baru").fetchall()
    return render_template('admin/data_training.html', data_update=
                           data_update,message=message,data=data)
```

#### Modul Program 4.31 Source Code Halaman Data Training Admin

Tampilan dalam halaman data *training* disajikan dalam gambar 4.6.

The screenshot shows a web application interface for data training. At the top, there are navigation links: Home, Kamus, Data Training (which is the active link), Statistic, Model, and Logout. Below this is a section titled "Tabel Data Training" with the sub-section "Real Comment". The table has columns: Real Comment, After Preprocessing, Score, and Action (with buttons for Edit and Delete). There are four rows of data:

Real Comment	After Preprocessing	Score	Action
Bang.. Apa betul sukses kya lu hrs idiot diu..?? Klu gtu ajarin jd org idiot dong bang kya lu.. Biar gue suskses	abang apa betul sukses seperti kamu harus idiot dahulu klu begitu ajarin que jadi orang idiot abang seperti kamu bliar gue suskses	0	<button>Edit</button> <button>Delete</button>
Abangg giginya kuning kaya kerak wc 😂	abang gigi kuning kaya kerak tangis	0	<button>Edit</button> <button>Delete</button>
Bang ikut gue yok, gue periksain itu pikiran lu sehat apa kakak	abang ikut gue ayo gue periksain di pikir kamu sehat apa tidak	0	<button>Edit</button> <button>Delete</button>
Bang @young_1ex18 itu cara fotonya pake green screen atau bugil tetulan . Serius nanya masa mau bugil sange donik cameramenya?	abang itu cara foto pakai green screen atau bugil tetul serius nanya masa mau bugil	0	<button>Edit</button> <button>Delete</button>

To the right of the table is a "Form Tambah Data Training" section with a text input field labeled "Kalimat" and a "Score 0-3" input field. A "Submit" button is located below these fields.

Gambar 4.6 Tampilan Halaman Data Training

#### e. Halaman *History Analysis Instagram* Admin

Halaman *history analysis instagram* merupakan halaman yang muncul saat admin menekan submenu *history analysis instagram* pada menu *data training* pada *navigation bar*. Halaman ini menampilkan data pengujian instagram yang telah dilakukan oleh user. Halaman ini akan menampilkan tabel berisi url instagram yang telah dilakukan beserta jumlah data yang diuji. *Source code* halaman *history analysis instagram* disajikan dalam modul program 4.32.

```
@bp.route('/history_instagram', methods=['POST', 'GET'])
@login_required
def history_instagram():
    db=get_db()
    if request.method=='POST':
        if request.form.get('show') is not None:
            url=request.form.get('show')
            print(url)
            data = db.execute("SELECT * FROM history_instagram WHERE
                url='{}'".format(url)).fetchall()
            result=True
    else:
        result=None
        url = request.form.get('delete')
        print(url)
        db.execute("DELETE FROM history_instagram WHERE
            url='{}'".format(url))
```

Modul Program 4.32 Halaman *History Analysis Instagram*

```

        db.commit()
        data = db.execute("SELECT COUNT(url) as jumlah,url FROM
        history_instagram GROUP BY url").fetchall()
    else:
        db = get_db()
        data = db.execute("SELECT COUNT(url) as jumlah,url FROM
        history_instagram GROUP BY url").fetchall()
        result=None
    return
render_template('admin/history_instagram.html',data=data,result=result)

```

### Modul Program Lanjutan 4.34 Halaman History Analysis Instagram

Tampilan halaman history *analysis instagam* disajikan dalam gambar 4.7 dan 4.8.

The screenshot shows a table with columns 'url', 'Total Komentar', and 'Action'. The 'Action' column contains two buttons: 'Lihat' (View) and 'Delete'. The table lists six URLs with their respective comment counts:

url	Total Komentar	Action
<a href="https://www.instagram.com/p/B_2khM4mOy/">https://www.instagram.com/p/B_2khM4mOy/</a>	50	Lihat Delete
<a href="https://www.instagram.com/p/CBPrP15HbjY/">https://www.instagram.com/p/CBPrP15HbjY/</a>	25	Lihat Delete
<a href="https://www.instagram.com/p/CCQZqH8BAQc/">https://www.instagram.com/p/CCQZqH8BAQc/</a>	23	Lihat Delete
<a href="https://www.instagram.com/p/CCV8K5_mp12/">https://www.instagram.com/p/CCV8K5_mp12/</a>	24	Lihat Delete
<a href="https://www.instagram.com/p/CDiG77nJcD/">https://www.instagram.com/p/CDiG77nJcD/</a>	80	Lihat Delete
<a href="https://www.instagram.com/p/CGuF1ezpwuA/">https://www.instagram.com/p/CGuF1ezpwuA/</a>	5	Lihat Delete

Gambar 4.7 Tampilan History Analysis Instagram Admin

The screenshot shows a table with columns 'Username', 'Comments', 'Label', and 'validitas'. The 'Label' column shows 'Non Cyberbullying' and the 'validitas' column shows 'valid'. The table lists nine comments from various users:

Username	Comments	Label	validitas
_noway	cantikkkkk bg!!! ❤	Non Cyberbullying	valid
antisewak_	mood banget ni orang 🌟	Non Cyberbullying	valid
awayuu_	CANTIK BANGET KEK AKU 😊	Non Cyberbullying	valid
bitch3ssx	aaa sayang kanni ❤	Non Cyberbullying	valid
bomobim	bahagia selalu kaa ❤	Non Cyberbullying	valid
bomobim	bahagia selalu kaa ❤	Non Cyberbullying	valid
bul_din	❤️❤️❤️ be strong, gun!!!	Non Cyberbullying	valid
citra.asri.754	Liat semuanya aja udah seneng 😊	Non Cyberbullying	valid
ellarianaaa	Cangtip selalu sihhhhh ❤	Non Cyberbullying	valid

Gambar 4.8 Tampilan History Analysis Instagram Detail Admin

f. Halaman *History Analysis Text Admin*

Halaman *history analysis text* merupakan halaman yang muncul saat admin menekan submenu *history analysis text* pada menu *data training* pada *navigation bar*. Halaman ini menampilkan data pengujian komentar yang telah dilakukan oleh *user*. Halaman ini akan menampilkan tabel berisi komentar yang di inputkan, hasil *text preprocessing* dan hasil identifikasinya. Selain itu terdapat sebuah tombol *delete* yang digunakan untuk menghapus hasil pengujian. *Source code* halaman *history analysis text* disajikan dalam modul program 4.35.

```
@bp.route('/history_analysis', methods=['POST', 'GET'])
@login_required
def history_analysis():
    db=get_db()
    if request.method=='POST':
        url = request.form.get('delete')
        print(url)
        db.execute("DELETE FROM text_analysis WHERE id='{}'".format(url))
        db.commit()
        db = get_db()
        data = db.execute("SELECT * from text_analysis").fetchall()
    return render_template('admin/history analisis.html', data=data)
```

**Modul Program 4.35 Source Code Halaman History Analysis Text Admin**

Halaman *history analysis text* disajikan dalam gambar 4.9.

The screenshot shows a web application interface titled 'Cyberbullying Identification'. The top navigation bar includes links for Home, Kamus, Data Training, Statistic, Model, and Logout. The main content area is titled 'Data Pengujian Instagram' and contains a message: 'Data yang berisi ujicoba yang telah dilakukan oleh pengguna.' Below this is a section titled 'Keterangan' with the note: 'Data tidak valid mendanakn komentar tidak dapat dianalisa oleh sistem.' A table displays the following data:

Komentar	hasil Preprocessing	Label	Action
Ada yang baru	ada yang baru	Irrelevant	<button>delete</button>
badan kamu dekil	badan kamu dekil	Cyberbullying	<button>delete</button>
badan kamu gemuk seperti badut	badan kamu gemuk seperti badut	Cyberbullying	<button>delete</button>
badan kamu gendut	badan kamu gendut	Cyberbullying	<button>delete</button>
badan kamu kurus kering seperti orang tidak pernah makan	badan kamu kurus kering seperti orang tidak pernah makan	Cyberbullying	<button>delete</button>
badan kurus seperti triplex	badan kurus seperti triplex	Cyberbullying	<button>delete</button>
dasar belagu	dasar belagu	Cyberbullying	<button>delete</button>
dasar bodoh	dasar bodoh	Cyberbullying	<button>delete</button>
dasar gendut	dasar gendut	Cyberbullying	<button>delete</button>

**Gambar 4.9 Tampilan History Analisis Text Admin**

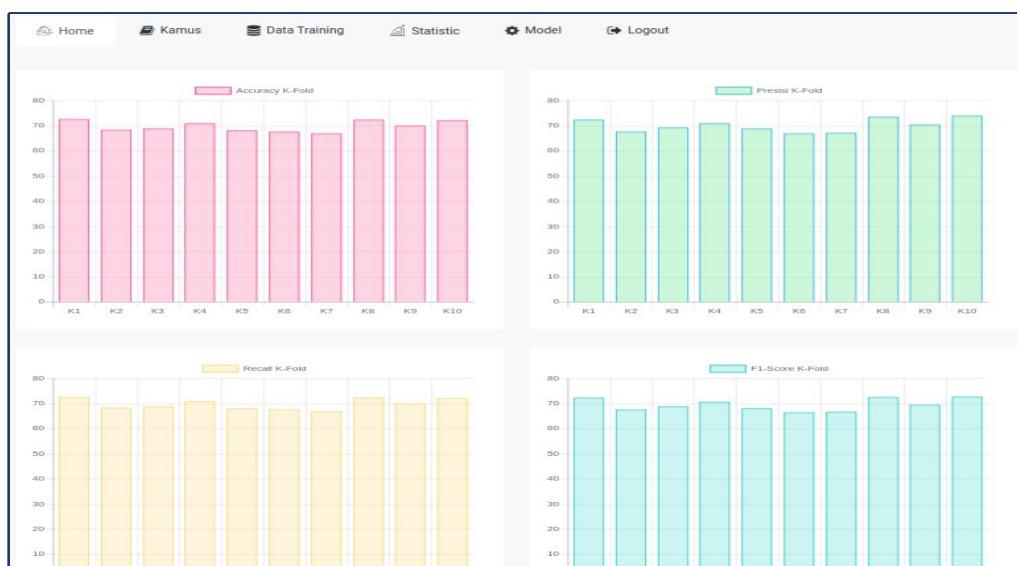
g. Halaman *Statistic Admin*

Halaman *Statistic* merupakan halaman yang akan tampil ketika admin memilih menu *statistic* pada *navigation bar*. Halaman ini menampilkan grafik hasil pengujian *k-fold cross validation* pada tiap matriks. Grafik ditampilkan dengan menggunakan *chart.js*. Grafik yang ditampilkan berupa akurasi, presisi, *recall* dan *F1-Score*. *Source code* halaman ini disajikan dalam modul program 4.36.

```
@bp.route('/barchart')
@login_required
def barchart():
    path = 'flaskr'
    basepath = os.path.join(path, "data", "fold", "kfold.csv")
    data = pd.read_csv(basepath)
    values = {
        'type': "bar",
        'accuracy': data['Accuracy'],
        'f1': data['F1-Score'],
        'presisi': data['Presisi'],
        'recall': data['Recall']
    }
    return render_template('admin/chart.html', values=values)
```

**Modul Program 4.36 Source Code Halaman Statistic**

Berdasarkan modul program 4.36, sistem akan membuka hasil pengujian yang telah disimpan berdasarkan modul program 4.24. Data pengujian disimpan dalam variabel *values* dan akan dikirimkan sebagai parameter ke halaman *html*. Halaman grafik pengujian ditampilkan dalam gambar 4.10.



**Gambar 4.10** Tampilan Halaman *Statistic Admin*

#### h. Halaman Model Admin

Halaman model admin merupakan halaman yang menampilkan pengujian yang dilakukan dalam membangun model. Pada halaman ini ditampilkan hasil pengujian data dalam bentuk hasil prediksi, *confusion matrix* model, *classification report* dan grafik pembentukan sistem. *Source code* halaman ini disajikan dalam modul program 4.37.

```
@bp.route('/statistic_model')
@login_required
def statistic_model():
    path='flaskr'
    basepath=os.path.join(path,"data","model","logger.csv")
    file=pd.read_csv(basepath)
    data={
        'accuracy':file['accuracy'],
        'loss':file['loss'],
        'epoch':file['epoch'],
        'val_accuracy':file['val_accuracy'],
        'val_loss':file['val_loss'],
    }
    report_csv = pd.read_csv(os.path.join('flaskr', "data",
                                         "confusion", "report.csv"))
    conf_mat = pd.read_csv(os.path.join('flaskr', "data", "confusion",
                                         "cf.csv"))
    classification = pd.read_csv(os.path.join('flaskr', "data",
                                                "confusion", "confusion_matrix.csv"))
    print(report_csv.iloc[:, 0])
    print(conf_mat.iloc[:, 0])
    data_y = {
        'row': classification.iloc[:, 0],
        'true': classification['true'],
        'pred': classification['pred']
    }
    report = {
        'row': report_csv.iloc[:, 0],
        'precision': report_csv['precision'],
        'recall': report_csv['recall'],
        'f1': report_csv['f1-score'],
        'support': report_csv['support'],
    }
    cf = {
        'row': conf_mat.iloc[:, 0],
        'Cyberbullying': conf_mat.iloc[:, 1],
        'Irrelevant': conf_mat.iloc[:, 2],
        'Netral': conf_mat.iloc[:, 3],
        'NonCyberbullying': conf_mat.iloc[:, 4],
    }
    return render_template('admin/model.html',data=data,report=report,
                           cf=cf,data_y=data_y)
```

**Modul Program 4.37** Source Code Halaman Model Admin

Berdasarkan modul program 4.37, ada beberapa variabel yang digunakan untuk menampilkan data seperti variabel *data* yang digunakan untuk menyimpan data *logger training* model, *data\_y* untuk menyimpan hasil pengujian data dan label asli, *cf* digunakan untuk menampilkan *confusion matrix* dan variabel *report* yang digunakan untuk menampilkan data pengujian model berupa akurasi, *f1-score*, presisi dan *recall*. Halaman model disajikan dalam gambar 4.11 dan gambar 4.12.

The screenshot shows the 'Cyberbullying Identification' application's admin panel. At the top, there are navigation links: Home, Kamus, Data Training, Statistic, Model, and Logout. Below these, the main content area has two sections:

- Pengujian Data Test:** A table showing test data with columns: Nomor, Score Data, and Prediksi. The data consists of 10 entries from index 0 to 9. The 'Score Data' column contains values like 0, 2, 0, 0, 1, 3, 0, 3, 0, 3. The 'Prediksi' column contains values like 0, 0, 0, 0, 1, 3, 0, 3, 3, 3.
- Classification Report:** A table showing classification metrics for different parameters (Param) and overall averages. The table includes columns: Param, Precision, Recall, F1-Score, and Support.

Param	Precision	Recall	F1-Score	Support
0	0.87	0.69	0.77	197.00
1	0.77	0.81	0.79	84.00
2	0.40	0.76	0.53	58.00
3	0.90	0.85	0.87	285.00
accuracy	0.79	0.79	0.79	0.79
macro avg	0.73	0.78	0.74	624.00
weighted avg	0.82	0.79	0.80	624.00

Below the report is a section titled 'Statistik Training dan Validasi' with a subtitle 'Grafik tersebut menggambarkan perbandingan accuracy dan val accuracy.' It features a line graph titled 'Grafik Accuracy dan Val Accuracy' comparing 'Accuracy' (green line) and 'Val Accuracy' (red line) over time.

Gambar 4.11 Tampilan Halaman Model Admin

This screenshot shows the same admin panel as above, but with different sections displayed:

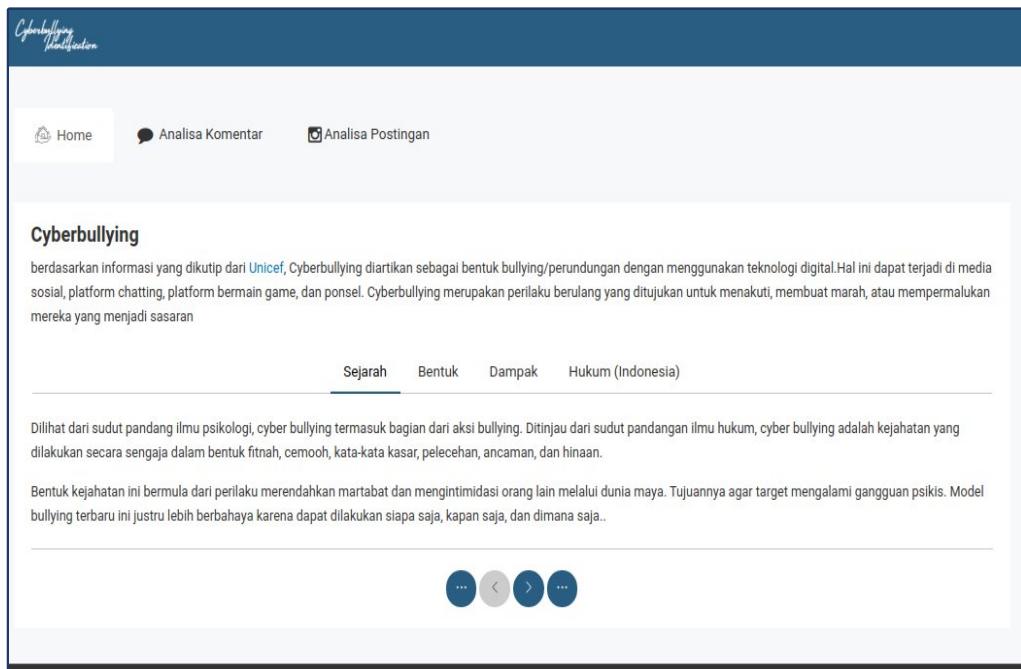
- Confusion Matrix:** A table showing the distribution of predicted vs actual classes. The table includes columns: Param, Cyberbullying, Irrelevant, Neutral, and Non Cyberbullying.
- Statistik Training dan Validasi:** A section with a subtitle 'Grafik tersebut menggambarkan perbandingan accuracy dan val accuracy.' It features a line graph titled 'Grafik Accuracy dan Val Accuracy' comparing 'Accuracy' (green line) and 'Val Accuracy' (red line) over time.
- Statistik Training dan Validasi:** A section with a subtitle 'Grafik dibawah menggambarkan perbandingan loss dan val loss.' It features a line graph titled 'Grafik Loss dan Val Loss' comparing 'loss' (green line) and 'val loss' (red line) over time.

Param	Cyberbullying	Irrelevant	Neutral	Non Cyberbullying
0	136	6	39	16
1	4	68	5	7
2	5	4	44	5
3	12	10	21	242

Gambar 4.12 Tampilan Halaman Model Admin

i. Halaman *Home User*

Halaman *home user* merupakan tampilan awal ketika user menjalankan program. Halaman ini menampilkan beberapa informasi terkait *cyberbullying*. Tampilan halaman *home user* disajikan dalam gambar 4.13.



**Gambar 4.13** Tampilan Halaman *Home User*

j. Halaman Analisa Komentar *User*

Halaman analisa komentar merupakan halaman yang muncul ketika user memilih menu analisis komentar. Halaman ini berisi sebuah *application form* berupa sebuah *textfield* dan *button*. User terlebih dahulu menginputkan komentar ke dalam *textfield* dan kemudian menekan tombol analisa. *Source code* halaman ini dapat dilihat pada modul program 4.38.

```
@bp.route('/hasil', methods=['GET', 'POST'])
def hasil():
    if request.method=='POST':
        data = request.form
        text_real = data["kalimat"]
        text=Activate().preprocessing(text_real)
        predict=list(LSTM().predict_comment(text)*100)
        valid=LSTM().cek_valid2(text)
        label = ["Cyberbullying", "Irrelevant", "Netral", "Non
        Cyberbullying"]
        print(valid)
```

**Modul Program 4.38** Source Code Halaman Analisa Komentar User

```

high_prob=[]
high_label=[]
if(valid=="valid"):
    for x in range(len(predict)):
        if max(predict) == predict[x]:
            high_prob.append(predict[x])
            high_label.append(label[x])
            predict.pop(x)
            label.pop(x)
            break
    db=get_db()
    db.execute('INSERT INTO text_analysis(teks,result,
                                                real_text) VALUES (?,?,?,?)', (text,
                                                high_label[0],text_real,))
    db.commit()
    result={
        'maximum_prob':high_prob,
        'maximum_label':high_label,
        'prob':predict,
        'label':label
    }
else:
    label=["Tidak teridentifikasi","Tidak teridentifikasi",
           "Tidak teridentifikasi"]
    high_label=["Tidak teridentifikasi"]
    high_prob=[0]
    predict=[0,0,0]
    db=get_db()
    db.execute('INSERT INTO text_analysis (teks,result,
                                                real_text) VALUES (?,?,?,?)', (text,
                                                high_label[0],text_real,))
    db.commit()
    result={
        'maximum_prob':high_prob,
        'maximum_label':high_label,
        'prob':predict,
        'label':label
    }
else:
    text=None
    result=None
    valid=None
db=get_db()
data=db.execute("SELECT real_text,result from text_analysis ORDER
                BY id DESC LIMIT 10").fetchall()
return render_template('user/identifikasi_komentar.html',
                      text=text,result=result,data=data,valid=valid)

```

#### **Modul Program 4.39 Lanjutan Source Code Halaman Analisa Komentar User**

Berdasarkan modul 4.38, proses analisa dimulai dengan melakukan proses *text preprocessing*, proses *text preprocessing* yang terjadi meliputi *case folding*, *cleansing*, *spelling correction*, *stopword removal* dan *stemming*. Komentar yang telah mengalami *text preprocessing* kemudian diuji dengan model yang telah dibentuk, proses pengujian dimulai dengan menginisiasikan model yang telah dibentuk dan kemudian dilanjutkan dengan

mengubah komentar ke dalam *sequence* dilanjutkan proses *padding*. *Sequence* yang telah terbentuk kemudian akan di prediksi menggunakan fungsi *predict*. *Source code* prediksi komentar disajikan dalam modul program 4.40.

```
def predict_comment(self, text):
    model = load_model(self.basepath_model)
    with open(self.basepath_token, 'rb') as handle:
        tokenizer2 = pickle.load(handle)
    komentar = [text]
    komentar = tokenizer2.texts_to_sequences(komentar)
    twt = pad_sequences(komentar, maxlen=50, dtype='int32',
                         value=0)
    sentiment = model.predict(komentar, batch_size=1, verbose=2)[0]
    return sentiment
```

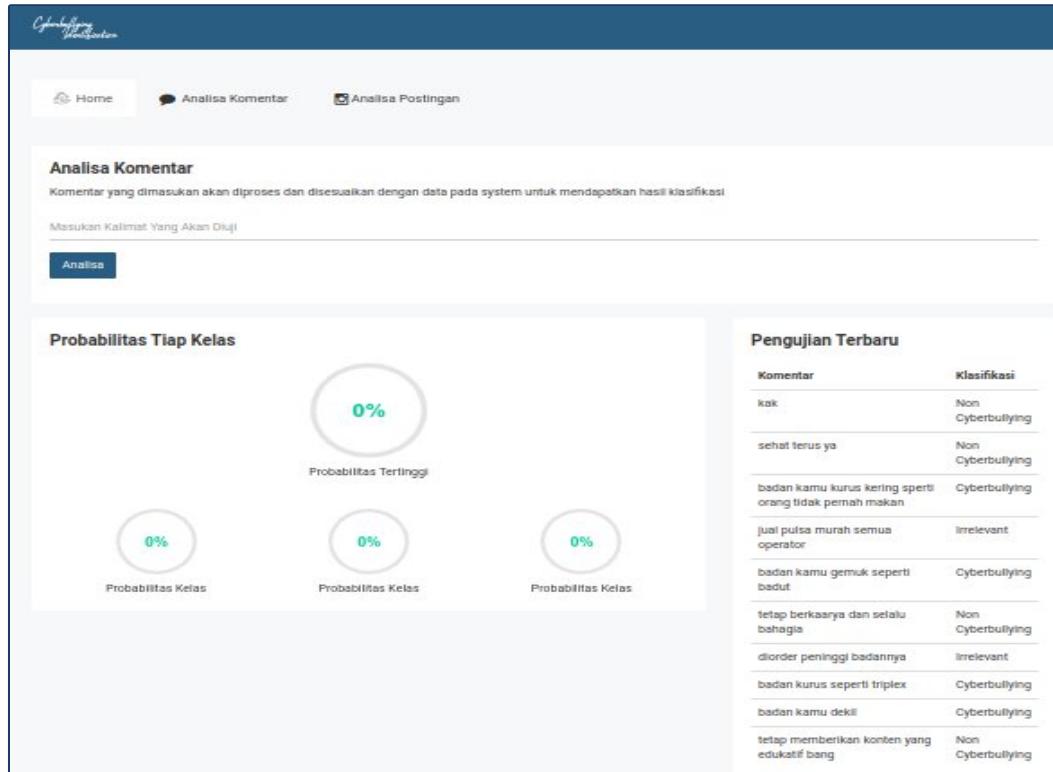
#### **Modul Program 4.40** Source Code Fungsi Prediksi Komentar

Sentimen yang dihasilkan kemudian akan dikembalikan ke fungsi utama. Proses tersebut kemudian dilanjutkan dengan proses *validasi* hasil yang ditunjukkan dengan modul program 4.41.

```
def cek_validasi(self, array):
    with open(self.basepath_token, 'rb') as handle:
        tokenizer2 = pickle.load(handle)
    komentar = [array]
    komentar = tokenizer2.texts_to_sequences(komentar)
    comment2 = pad_sequences(komentar, maxlen=50, dtype='int32',
                             value=0)
    x = comment2[0]
    print(x)
    if x[-1] == 0:
        return "invalid"
    else:
        return "valid"
```

#### **Modul Program 4.41** Source Code Fungsi Cek Validitas

Proses validasi dilakukan dengan melakukan pengecekan indeks terakhir dari *sequence*, jika indeks terakhir bernilai 0 maka hasil pengujian tidak valid namun jika indeks terakhir selain 0 maka pengujian tersebut valid. Tampilan dari halaman pengujian disajikan dalam gambar 4.14.



**Gambar 4.14** Tampilan Halaman Analisa Komentar

#### k. Halaman Analisis Postingan User

Halaman analisis postingan merupakan halaman dengan fungsi melakukan *scraping* data sekaligus pengujian. Halaman ini berisi *application form* berupa sebuah *textfield* untuk *input url* dan sebuah *button* untuk melakukan *request*. User memasukan alamat postingan ke dalam *textfield* dan kemudian menekan tombol analisa postingan. *Source code* halaman ini disajikan dalam modul program 4.42.

```
@bp.route('/instagram', methods=['GET', 'POST'])
def instagram():
    if request.method=='POST':
        text="correct"
        data = request.form
        url = data["url"]
        name=os.environ.get('USERNAME_INSTAGRAM')
        password=os.environ.get('PASSWORD_INSTAGRAM')
        tuple=ScrapePost().login_page(url, name, password)
        db=get_db()
        try:
            db.executemany('INSERT INTO history_instagram
                           (url,score,comments,valid,username,image) VALUES
                           (?,?,?,?,?,?)',tuple)
        except:
            db.execute('INSERT INTO
```

**Modul Program 4.42** *Source Code* Halaman Analisis Postingan User

```

        history_instagram(url,score,comments,valid,username,image) VALUES (?,?,?,?,?,?)', tuple)
        db.commit()
        result=db.execute("SELECT * from history_instagram WHERE url='{}'".format(url)).fetchall()
    else:
        text = ["sucess"]
        db = get_db()
        url = request.args.get('url2', '')
        result = db.execute("SELECT * from history_instagram WHERE url='{}'".format(url)).fetchall()
        db=get_db()
        data=db.execute("SELECT * from text_analysis ORDER BY id DESC LIMIT 10").fetchall()
        pengujian_terdahulu=db.execute("SELECT COUNT(url) as jumlah,url FROM history_instagram GROUP BY url LIMIT 15").fetchall()
    return render_template('user/identifikasi_instagram.html', text=text, result=result,data=data,data_other=pengujian_terdahulu)

```

#### **Modul Program 4.43 Lanjutan Source Code Halaman Analisa Postingan User**

Berdasarkan modul program 4.42, Proses pertama pada tahapan *scraping* postingan instagram adalah dengan melakukan proses *login*. Proses *login* yang terjadi dilakukan secara otomatis menggunakan *library selenium*. *Source code* proses otomatisasi *login* ditunjukkan dengan modul program 4.44.

```

def login_page(self,url,username,password):
    try:
        self.driver.get(self.login)
        WebDriverWait(self.driver, 20).until(
            EC.element_to_be_clickable((By.XPATH,
                "//input[@name='username']"))).send_keys(username)
        WebDriverWait(self.driver, 20).until(
            EC.element_to_be_clickable((By.XPATH,
                "//input[@name='password']"))).send_keys(password)
        WebDriverWait(self.driver,
            20).until(EC.element_to_be_clickable((By.XPATH,
                "//input[@name='password']"))).send_keys(
                    Keys.ENTER)
        time.sleep(5)
        return self.get_post(url)
    except TimeoutException:
        return False

```

#### **Modul Program 4.44 Source Code Fungsi Otomatisasi Login**

Jika proses *login* berjalan sukses, kemudian akan dilanjutkan dengan proses *scraping* halaman postingan. *Source code* *scraping* halaman postingan disajikan dalam modul program 4.45.

```

def get_post(self,url):
    self.driver.get(url)
    while self.check_exists_by_xpath("//div/ul/li/div/button"):

```

#### **Modul Program 4.45 Source Code Scraping Postingan Instagram**

```

load_more_comments_element = self.driver.find_element
    _by_xpath("//div/ul/li/div/button")
load_more_comments_element.click()
time.sleep(0.2)
time.sleep(2)
soup = BeautifulSoup(self.driver.page_source, 'lxml')
comms = soup.find_all('div', attrs={'class': 'C4VMK'})
soup_2 = BeautifulSoup(str(comms), 'lxml')
spans = soup_2.find_all('span')
image=soup.find_all('div',attrs={'class':'Jv7Aj'})
soup_3=BeautifulSoup(str(image), 'lxml')
img=soup_3.find_all('img')
for x in img:
    self.image.append(x["src"])
if len(image)!=len(comms):
    self.image.pop(0)
comments = [i.text.strip() for i in spans if i != '']
self.check_verified_account(comments)
self.parse_text()
self.identification()
hasil_label=[]
for x in self.result:
    if x==0:
        hasil_label.append("Cyberbullying")
    elif x==1:
        hasil_label.append("Irrelevant")
    elif x==2:
        hasil_label.append("Netral")
    else:
        hasil_label.append("Non Cyberbullying")
source=[]
for x in range(len(hasil_label)):
    source.append(url)
tuple = list(zip(source, hasil_label, self.comments,
                 self.valid, self.username, self.image))
self.driver.close()
return tuple

```

#### **Modul Program 4.46 Lanjutan Source Code Scraping Postingan Instagram**

Berdasarkan modul program 4.45, *browser* akan secara otomatis melakukan pencarian terhadap *xpath* yang telah ditentukan. Bila *xpath* ditemukan kemudian akan dilakukan proses pengambilan elemen dengan menggunakan fungsi *beautifulSoup*. Komentar yang ditemukan kemudian akan di identifikasi dengan menggunakan fungsi *identification*. *Source code* pada fungsi *identification* disajikan dalam modul program 4.47.

```

def identification(self):
    result = LSTM()
    valid=result.cek_valid(self.comments)
    for x in valid:
        self.valid.append(x)
    prediction=result.prediction()
    for x in prediction:
        self.result.append(x)

```

#### **Modul Program 4.47 Source Code Fungsi Identification**

Proses identifikasi didahului dengan pengecekan validitas komentar yang disesuaikan dengan kamus fitur yang telah terbentuk. Jika komentar terdapat dalam kamus maka dianggap valid dan jika tidak maka tidak dianggap valid. *Source code* pada program ini disajikan dalam modul program 4.48.

```
def cek_valid(self, array):
    global comment
    for i in range(len(array)):
        clear = self.normal.preprocessing(array[i])
        self.data_scrape.append(clear)
    with open(self.basepath_token, 'rb') as handle:
        tokenizer2 = pickle.load(handle)
    comment = tokenizer2.texts_to_sequences(self.data_scrape)
    comment = pad_sequences(comment, maxlen=50, dtype='int32',
                           value=0)
    validitas = []
    for x in comment:
        if x[-1] == 0:
            validitas.append("invalid")
        else:
            validitas.append("valid")
    return validitas
```

#### **Modul Program 4.48** Source Code Cek Validitas Data

Berdasarkan modul program 4.48, pengecekan validitas ditunjukkan dengan mengecek elemen terakhir dari *sequence*. Jika indeks terakhir dari *sequence* bernilai 0 (*padding*) menunjukkan bahwa tidak ada fitur yang merepresentasikan kalimat tersebut. Selanjutnya merupakan proses identifikasi yang ditunjukkan dengan fungsi *prediction*. *Source code* pada fungsi *prediction* disajikan dengan modul pogram 4.49.

```
def prediction(self):
    model = load_model(self.basepath_model)
    return model.predict_classes(comment, batch_size=1)
```

#### **Modul Program 4.49** Source Code Fungsi Prediction

Tampilan pada halaman analisis komentar disajikan dalam gambar 4.15.

The screenshot shows a web-based application for analyzing Instagram comments. At the top, there are three navigation buttons: 'Home', 'Analisa Komentar' (Analysis Comment), and 'Analisa Postingan' (Analysis Post). Below this is a section titled 'Analisa Komentar' with a sub-instruction: 'Komentar yang dimasukan akan diproses dan disesuaikan dengan data pada system untuk mendapatkan hasil klasifikasi'. A text input field labeled 'Masukan URL post Instagram' is present, along with a blue button labeled 'Analisa Postingan'. To the right, a sidebar titled 'Recent Posts' lists several Instagram URLs and their corresponding comment counts.

Data Pengujian		
hasil pengujian dari postingan akan muncul pada tabel dibawah ini.		
Show	entries	
username	komentar	klasifikasi
	mantap	Non Cyberbullying
	Seperi yang selalu que bilang dari dulu.Yang paling penting baik aja dulu sama orang.Urusan dibalas dengan kebalasan atau sebaliknya itu belakangan, gausah dipikirin malah.Susah hari mereka juga akan sadar, kalian itu sehama ini tulus dan berusaha yang terbaik.	Non Cyberbullying
	Cangtip bgt karin 😍😍	Non Cyberbullying
	panutan😊	Non Cyberbullying
	idola❤	Non Cyberbullying
	sayang bgt❤️❤️❤️	Non Cyberbullying

Gambar 4.15 Tampilan Halaman Analisa Instagram

### 4.3. Pengujian

Tahap pengujian dilakukan dengan tujuan untuk mengetahui tingkat keberhasilan proses identifikasi yang telah dibangun. Implementasi pengujian dengan menggunakan *confusion matrix*. Melalui *confusion matrix* dapat diketahui nilai akurasi, presisi, *recall* dan *F-1 Score*. Pengujian metode LSTM pada penelitian ini dilakukan dengan cara membangun model berdasarkan data komentar pada *dataset* yang terbagi menjadi empat kelas. Pengujian yang dilakukan dengan metode evaluasi *k-fold cross validation*. Proses evaluasi dengan *k-fold cross validation* dilakukan sejumlah  $k=10$  (*10 fold*). Setiap *fold* akan menggunakan pembagian data dengan proporsi yang sama namun dengan data uji yang berbeda. Setiap iterasi membagi data ke dalam 10% data uji dan sisanya sebagai data latih. Setiap *subset* data memiliki data latih dan uji yang berbeda, jika *subset* pertama digunakan sebagai data uji maka *subset* kedua hingga ke sepuluh akan digunakan sebagai data latih. Hal tersebut juga akan terjadi pada *subset* data selanjutnya.

#### 4.3.1. Pengujian *Confusion Matrix*

Tabel *confusion matrix* digunakan sebagai acuan dalam melakukan pengujian akurasi, presisi, *recall* dan *F1-Score*. Melalui *confusion matrix* dapat diketahui nilai *True Positive* (TP), *False Positive* (FP), *False Negative* (FN) dan *True Positive* (TP) yang dihasilkan oleh klasifikasi model. Berikut ini merupakan hasil TP, FP, FN dan TN tabel *confusion matrix* model klasifikasi sentimen menggunakan *10-fold cross validation* dapat dilihat pada tabel 4.28.

**Tabel 4.28** *Confusion Matrix K-fold Cross Validation*

Fold	Confusion Matrix															
	TP				FP				FN				TN			
	0	1	2	3	0	1	2	3	0	1	2	3	0	1	2	3
1	76	35	22	96	11	12	33	22	29	4	23	22	153	194	207	133
2	73	34	26	85	18	15	25	31	12	9	39	29	145	184	192	133
3	84	34	23	74	24	13	33	22	25	6	33	28	131	181	192	141
4	78	47	22	74	14	12	35	24	25	8	22	31	143	174	199	147
5	75	36	14	83	26	15	42	16	24	10	34	31	133	172	208	125
6	74	28	22	97	11	14	39	22	26	7	21	32	147	193	199	124
7	63	37	21	75	21	15	45	30	30	11	37	33	133	159	175	121
8	76	37	22	71	18	14	44	25	33	10	40	28	130	169	184	135
9	64	37	27	80	24	14	34	27	20	10	33	36	144	171	181	128
10	76	30	31	70	21	9	36	34	27	14	30	29	131	177	176	137

Selanjutnya akan dilakukan perhitungan nilai akurasi, presisi, *recall* dan *F1-Score*. Berikut ini merupakan hasil pengujian model klasifikasi sentimen menggunakan *10-fold cross validation* yang disajikan dalam tabel 4.29.

**Tabel 4.29** Hasil Pengujian *K-fold Cross Validation*

Fold	Accuracy	F1-Score	Presisi	Recall
1	73.2 %	74.0 %	75.5 %	73.2 %
2	70.0 %	71.5 %	73.4 %	70.0 %
3	70.0 %	70.5 %	71.3 %	70.0 %
4	71.0 %	71.4 %	72.6 %	71.0 %
5	66.4 %	66.8 %	67.8 %	66.4 %
6	71.0 %	70.4 %	71.0 %	71.0 %
7	61.5 %	62.0 %	62.7 %	61.5 %
8	66.7 %	67.5 %	69.0 %	66.7 %
9	65.7 %	67.3 %	69.1 %	65.7 %
10	65.7 %	66.8 %	68.1 %	65.7 %
rataan	68.1 %	68.8 %	70.08 %	68.1 %

### 4.3.2. Perbandingan Algoritma

Perbandingan algoritma dilakukan untuk mengukur tingkat performa algoritma LSTM dalam mengidentifikasi komentar. Perbandingan dilakukan dengan membandingkan algoritma LSTM dengan *random forest classifier* (Novalaita et al., 2018). Prediksi menggunakan algoritma LSTM memperoleh hasil *confusion matrix* yang disajikan dalam tabel 4.30.

**Tabel 4.30** *Confusion Matrix LSTM*

Prediksi	Cyberbullying	Irrelevant	Neutral	NonCyberbullying
<b>Cyberbullying</b>	151	4	15	8
<b>Irrelevant</b>	4	74	9	9
<b>Neutral</b>	26	9	37	34
<b>NonCyberbullying</b>	26	8	29	171

Prediksi menggunakan algoritma *random forest classifier* memperoleh hasil *confusion matrix* yang disajikan dalam tabel 4.31.

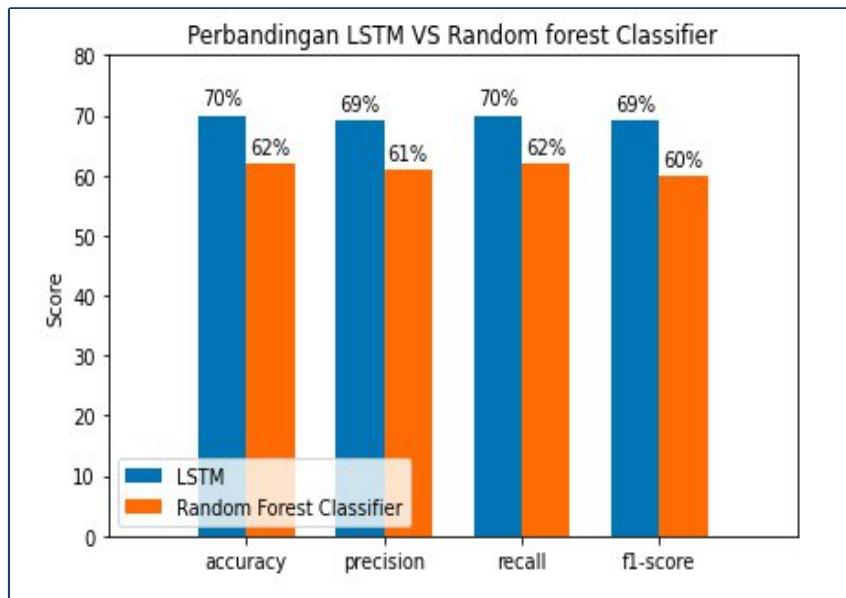
**Tabel 4.31** *Confusion Matrix Random Forest Classifier* (Novalaita et al., 2018)

Prediksi	Cyberbullying	Irrelevant	Neutral	NonCyberbullying
<b>Cyberbullying</b>	134	5	9	30
<b>Irrelevant</b>	13	64	8	11
<b>Neutral</b>	46	3	21	36
<b>NonCyberbullying</b>	40	6	25	163

Hasil perbandingan performa algoritma disajikan dalam tabel 4.32 dan gambar 4.16.

**Tabel 4.32** Hasil Perbandingan Algoritma

Parameter	LSTM	<i>Random Forest Classifier</i> (Novalaita et al., 2018)
Accuracy	70 %	62 %
Presisi	69 %	61 %
Recall	70 %	62 %
F1-Score	69 %	60 %



Gambar 4.16 Perbandingan Performa LSTM dan *Random Forest Classifier*

#### 4.3.3. Pembahasan

Dari implementasi dan pengujian yang telah dilaksanakan, diperoleh hasil bahwa sistem dapat melakukan klasifikasi sentimen terhadap komentar dengan baik. Nilai rata-rata akurasi model yang diperoleh sebesar 68%, presisi sebesar 70.08%, *recall* sebesar 68.1% dan *F1-Score* sebesar 68.8%.

Kondisi ideal dari suatu sistem adalah apabila suatu *recall* dan presisi memiliki nilai sebesar 1:1 (Pao, 2008). Berdasarkan hasil tersebut, presisi dan *recall* yang didapatkan dari pengujian tersebut memiliki nilai yang hampir sama walaupun berbeda 2% sehingga sistem ini dapat dikatakan mendekati ideal. kemudian berdasarkan teori Lancaster dalam (Pendit, 2008) menyatakan bahwa efektifitas suatu sistem di kategorikan menjadi dua yaitu efektif jika nilai presisi dan *recall* diatas 50% dan tidak efektif jika nilainya dibawah 50% dalam skala 1-100%. Pada penelitian ini presisi dan *recall* yang didapat telah berada diatas 50%, sehingga dapat dikatakan bahwa sistem yang dibangun sudah efektif.

Dari segi performa, performa algoritma *long short term memory* dibandingkan dengan algoritma *random forest classifier* (Novalaita et al., 2018) dalam melakukan klasifikasi komentar terhadap data yang *imbalance* memperoleh hasil yang lebih baik.

Dalam pengujian akurasi lstm mencapai 70% sedangkan *random forest classifier* hanya mencapai 62%, nilai presisi 69% dibandingkan 61 % untuk lstm, *recall* 70% dibanding 62% untuk lstm dan nilai *F1-Score* lstm mencapai 79% dibandingkan *random forest classifier* yang hanya mencapai 60%. Hasil ini menunjukkan bahwa algoritma *long short term memory* memiliki performa yang lebih baik dibandingkan *random forest classifier* dalam menangani klasifikasi komentar instagram.

## BAB V

### PENUTUP

#### 5.1. Kesimpulan

Berdasarkan penelitian yang telah dilakukan, maka diperoleh kesimpulan sebagai berikut :

1. Telah dibangun sistem identifikasi komentar dari media sosial instagram menggunakan metode *long short term memory*. Komentar di identifikasi menjadi empat kelas yaitu *cyberbullying*, *irrelevant*, *neutral* dan *non cyberbullying*.
2. Hasil pengujian analisis komentar instagram dengan menggunakan *confusion matrix* dengan pendekatan *10-fold cross validation* menghasilkan rata-rata nilai akurasi untuk model sebesar 68%, presisi sebesar 70.08%, *recall* sebesar 68.1% dan *F1-score* sebesar 68.8%.
3. Hasil perbandingan algortima *long short term memory* dibandingkan dengan algortitma *random forest classifier* memiliki performa yang lebih baik yang ditunjukan keunggulan hasil akurasi, presisi, *recall* dan *F1-Score*.

#### 5.2. Saran

Penelitian ini masih terdapat beberapa keterbatasan dan kekurangan. Keterbatasan dan kekurangan ini bisa dijadikan acuan atau pertimbangan untuk penelitian selanjutnya. Pada penelitian ini didapatkan hasil pengujian dengan nilai akurasi yang belum tinggi karena jumlah data yang belum seimbang antar kelas atau *imbalance data*. Saran untuk penelitian selanjutnya adalah dengan menambahkan data latih dengan menyeimbangkan data.

## DAFTAR PUSTAKA

- Andriansyah, M. (2017). *Cyberbullying Comment Classification on Indonesian Selebgram Using Support Vector Machine Method*. November.
- <https://doi.org/10.1109/IAC.2017.8280617>
- Arifuddin, M. R., Magister, M., Komunikasi, I., & Harapan, U. P. (n.d.). *Dari foto dan video ke toko: perkembangan instagram dalam perspektif konstruksi sosial*.
- Hermiyanty, Wandira Ayu Bertin, D. S. (2017). 濟無No Title No Title. *Journal of Chemical Information and Modeling*, 8(9), 1–58.
- <https://doi.org/10.1017/CBO9781107415324.004>
- Le, X. H., Ho, H. V., Lee, G., & Jung, S. (2019). Application of Long Short-Term Memory (LSTM) neural network for flood forecasting. *Water (Switzerland)*, 11(7).
- <https://doi.org/10.3390/w11071387>
- Luqyana, W. A., Cholissodin, I., & Perdana, R. S. (2018). *Analisis Sentimen Cyberbullying pada Komentar Instagram dengan Metode Klasifikasi Support Vector Machine*. 2(11), 4704–4713.
- Nurrohmat, M. A., & SN, A. (2019). Sentiment Analysis of Novel Review Using Long Short-Term Memory Method. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 13(3), 209. <https://doi.org/10.22146/ijccs.41236>
- Priyanto, A., & Ma’arif, M. R. (2018). Implementasi Web Scrapping dan Text Mining untuk Akuisisi dan Kategorisasi Informasi dari Internet (Studi Kasus: Tutorial Hidroponik). *Indonesian Journal of Information Systems*, 1(1), 25–33.
- <https://doi.org/10.24002/ijis.v1i1.1664>
- Rifauddin, M. (2016). *FENOMENA CYBERBULLYING PADA REMAJA ( Studi Analisis Media Sosial Facebook )*.
- Sakti, B. C., Yulianto, M., Komunikasi, J. I., & Index, G. (2013). Enggunaan Media Sosial

- Instagram Dalam Pembentukan Identitas Diri Remaja. *Interaksi-Online*, 1–12.  
<https://ejournal3.undip.ac.id/index.php/interaksi-online/article/download/21950/20197>
- Utami, A. S. F., & Baiti, N. (2018). Pengaruh Media Sosial Terhadap Perilaku Cyberbullying Pada Kalangan Remaja. *Cakrawala - Jurnal Humaniora*, 18(2), 257–262. <http://ejournal.bsi.ac.id/ejurnal/index.php/cakrawala/article/view/3680/2624>
- Wangsanegara, N. K., & Subaeki, B. (2015). Implementasi Natural Language Processing Dalam Pengukuran Ketepatan Ejaan Yang Disempurnakan (Eyd) Pada Abstrak Skripsi Menggunakan Algoritma Fuzzy Logic. *Jurnal Teknik Informatika*, 8(2).  
<https://doi.org/10.15408/jti.v8i2.3185>
- Yana Choria Utami. (2014). Cyberbullying di Kalangan Remaja (Studi tentang Korban Cyberbullying di Kalangan Remaja di Surabaya). *Universitas Airlangga*, 3(3), 1–10.
- Zulfa, I., & Winarko, E. (2017). Sentimen Analisis Tweet Berbahasa Indonesia Dengan Deep Belief Network. *IJCCS (Indonesian Journal of Computing and Cybernetics Systems)*, 11(2), 187. <https://doi.org/10.22146/ijccs.24716>