IMPLEMENTASI SENTIMENTAL ANALYSIS DAN NETWORK ANALYSIS OPINI PENGGUNA TWITTER PADA ISU POLITIK BERBASIS WEB



SKRIPSI

SYAIR MUHARRAM

42619022

PROGRAM STUDI D-4 TEKNIK MULTIMEDIA DAN JARINGAN

JURUSAN TEKNIK ELEKTRO

POLITEKNIK NEGERI UJUNG PANDANG

2023

# HALAMAN PENGESAHAN

Skripsi ini dengan judul **Implementasi Sentimental Analysis Dan Network Analysis Opini Pengguna Twitter Pada Isu Politik Berbasis Web** oleh SYAIR MUHARRAM NIM 426 190 22 dinyatakan layak untuk diujikan.

Makassar, September 2023

Pembimbing I, Pembimbing II,

Asriyadi, S.ST.,M.Eng Muh. Ahyar, SST.,MT

NIP. 197708142003121002 NIP. 198410272008121003

Mengetahui

Ketua Program Studi,

Asriyadi, S.ST.,M.Eng

NIP. 197708142003121002

# HALAMAN PENERIMAAN

Pada hari ini, hari Jumat tanggal 29 September 2023, Tim Penguji Sidang Skripsi telah menerima dengan baik hasil skripsi oleh mahasiswa: Syair Muharram NIM 42619022 dengan judul Implementasi Sentimental Analysis Dan Network Analysis Opini Pengguna Twitter Pada Isu Politik Berbasis Web.

Makassar, September 2023

Tim Penguji Ujian Sidang Skripsi:

1. Mardawia M. Parenreng, S.ST., M.T. Ketua ( )

2. Ainun Jariyah, S.T., M.T. Sekretaris ( )

3. Alvian Bastian Anggota ( )

4. Dr. Ir. Hafsah Nirwana, M.T. Anggota ( )

5. Asriyadi, S.ST., M.Eng. Pembimbing I ( )

6. Muh. Ahyar, S.ST., M.T. Pembimbing II ( )

**BERITA ACARA PELAKSANAAN UJIANG SIDANG SKRIPSI**

# KATA PENGANTAR

Puji syukur kami panjatkan ke hadirat Allah swt. Yang telah melimpahkan rahmat dan karunia-Nya sehingga penulisan skripsi dengan judul **Implementasi Sentimental Analysis Dan Network Analysis Opini Pengguna Twitter Pada Isu Politik Berbasis Web** program diploma empat (D-4) bidang Teknik Multimedia dan Jaringan dapat kami selesaikan.

Penyusunan skripsi dimulai dengan perbaikan, pemutakhiran hingga proses konsultasi langsung antara penulis dengan dosen pengarah program studi berdasarkan kaidah kepenulisan yang berlaku di Politeknik Negeri Ujung Pandang.

Dengan kerendahan hati, tidak ada kata yang dapat menggantikan ucapan terima kasih kepada kedua orangtua, yang turut mendukung dan mendoakan. Ucapan terima kasih juga penulis sampaikan untuk kedua pembimbing skripsi yang turut membantu dalam penyusunan skripsi ini.

Penulis juga turut mengucapkan terima kasih dan apresasi setinggi-tingginya kepada:

1. Bapak Ir. Ilyas Mansyur, M.T. selaku Direktur Politeknik Negeri Ujung Pandang.

2. Bapak Ahmad Rizal Sultan, S.T., M.T., Ph.D. selaku Ketua Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang.

3. Bapak Asriyadi, S.ST.,M.Eng. selaku Ketua Program Studi D-4 Teknik Multimedia dan Jaringan Politeknik Negeri Ujung Pandang.

4. Bapak Asriyadi, S.ST.,M.Eng. Sebagai Pembimbing I dan Bapak Muh. Ahyar, S.ST., MT. selaku pembimbing II yang telah meluangkan waktunya dalam membimbing penulis sehingga penulisan skripsi ini dapat terselesaikan.

5. Bapak/Ibu dosen Jurusan Teknik Elektro Politeknik Negeri Ujung Pandang yang memberikan ilmu yang berharga selama perkuliahan sampai dengan penyusunan skripsi ini.

6. Bapak/Ibu Staff Program Studi D4 Teknik Multimedia dan Jaringan yang dimana dedikasi dan pelayanan yang diberikan patut diapresiasi.

6. Teman-teman di Program Studi D4 Teknik Multimedia dan Jaringan yang menemani selama empat tahun di Politeknik Negeri Ujung Pandang.

Skripsi ini masih jauh dari kata sempurna, maka dari itu penulis sangat menerima kritik dan saran yang dapat menjadikan skripsi ini lebih baik dan bermanfaat kedepannya.

Makassar, September 2023

Penulis

**DAFTAR ISI**

[HALAMAN PENGESAHAN ii](#_Toc162930348)

[HALAMAN PENERIMAAN iii](#_Toc162930349)

[BERITA ACARA PELAKSANAAN UJIAN SIDANG SKRIPSI iii](#_Toc162930349)

[KATA PENGANTAR v](#_Toc162930350)

[DAFTAR ISI v](#_Toc162930350)ii

[DAFTAR TABEL ix](#_Toc162930351)

[DAFTAR GAMBAR x](#_Toc162930352)

[DAFTAR RUMUS xii](#_Toc162930353)

[DAFTAR LAMPIRAN xiii](#_Toc162930354)

[SURAT PERNYATAAN xiii](#_Toc162930354)

[RINGKASAN xiii](#_Toc162930354)

[BAB I PENDAHULUAN 1](#_Toc162930355)

[1.1 Latar Belakang 1](#_Toc162930356)

[1.2 Rumusan Masalah 3](#_Toc162930357)

[1.3 Ruang Lingkup Penelitian 3](#_Toc162930358)

[1.4 Tujuan Penelitian 4](#_Toc162930359)

[1.5 Manfaat Penelitian 4](#_Toc162930360)

[BAB II TINJAUAN PUSTAKA 5](#_Toc162930361)

[2.1 Penelitian Terdahulu 5](#_Toc162930362)

[*2.2* *Natural Language Processing* 6](#_Toc162930363)

[2.3 LSTM 6](#_Toc162930364)

[*2.4* *Sentimental Analysis* 7](#_Toc162930365)

[*2.5* *Social Network Analysis* 8](#_Toc162930366)

[2.6 Twitter 11](#_Toc162930367)1

[*2.7* *Text Mining* 11](#_Toc162930368)

[2.8 Python 14](#_Toc162930369)

[2.9 NLTK 15](#_Toc162930370)

[2.10 TwitterWebScraper 16](#_Toc162930371)

[2.11 NetworkX 17](#_Toc162930372)

[2.12 Visual Studio Code 17](#_Toc162930373)

[2.13 Flask 17](#_Toc162930374)

[2.14 NodeJs 18](#_Toc162930375)

[2.15 MySQL 18](#_Toc162930376)

[2.16 Metode Pengembangan *Waterfall* 19](#_Toc162930377)

[2.17 Pengujian *Black Box* 19](#_Toc162930378)

[BAB III METODE PENELITIAN 21](#_Toc162930379)

[3.1 Tempat dan Waktu Penelitian 21](#_Toc162930380)

[3.2 Alat dan Bahan 21](#_Toc162930381)

[3.3 Prosedur Penelitian 21](#_Toc162930382)

[3.3.1 Metode Penelitian dan Tahapan Pengembangan Aplikasi 22](#_Toc162930383)

[3.3.2 Analisis Kebutuhan 22](#_Toc162930384)

[3.3.3 Desain Sistem 22](#_Toc162930385)

[3.3.4 Implementasi dan Pengkodean 28](#_Toc162930386)

[3.3.5 Testing 28](#_Toc162930387)

[3.3.6 Pemeliharaan 28](#_Toc162930388)

[3.4 Pengujian Sistem 28](#_Toc162930389)

[BAB IV HASIL DAN PEMBAHASAN 29](#_Toc162930390)

[4.1 Hasil](#_Toc162930380) 29

[4.1.1 *Sentiment Analysis*](#_Toc162930383) 29

[4.1.2 *Network Analysis*](#_Toc162930384) 35

[4.1.3 Tampilan *Website*](#_Toc162930384) 36

[4.2 Metode Validasi](#_Toc162930380) 42

[4.2.1 Metode Validasi *Sentimental* *Analysis*](#_Toc162930383) 42

[4.2.2 MetodeValidasi *Network Analysis*](#_Toc162930383) 44

[4.3 Metode Pengujian *Website*](#_Toc162930380) 45

[4.3.1 Pengujian pada akun @KawalPemilu\_org](#_Toc162930383) 46

[4.3.2 Pengujian *website*](#_Toc162930383) 50

[BAB V KESIMPULAN DAN SARAN 53](#_Toc162930391)3

[5.1 Kesimpulan 53](#_Toc162930392)3

[5.2 Saran 54](#_Toc162930393)4

DAFTAR PUSTAKA 55

LAMPIRAN [60](#_Toc162930394)

# DAFTAR TABEL

Tabel 4.1 Tabel contoh hasil *preprocessing* 30

Tabel 4.2 Sampel hasil sentimen 43

Tabel 4.3 Pengujian *Blackbox* 51

# DAFTAR GAMBAR

Gambar 2.1 Arsitektur LSTM 7

Gambar 2.2 Contoh diagram pada *sentimental analysis* 8

Gambar 2.3 Contoh implementasi *social network* *analysis* pada penyebaran informasi covid-19 10

Gambar 3.1 Metode *waterfall* 22

Gambar 3.2 Tampilan *flowchart sentiment analysis* 23

Gambar 3.3 Tampilan *flowchart network analysis* 24

Gambar 3.4 *Use case* diagram 25

Gambar 3.5 Tampilan halaman *home* 26

Gambar 3.6 Tampilan halaman LSTM *predictor* 26

Gambar 3.7 Tampilan halaman hasil 27

Gambar 3.8 Tampilan halaman admin 27

Gambar 4.1 *Lexicon* positif dan negatif dari InSet 31

Gambar 4.2 Akurasi *training* model LSTM 32

Gambar 4.3 Akurasi *testing* model LSTM 32

Gambar 4.4 Akurasi *training* setelah penerapan *K-Fold cross validation* 33

Gambar 4.5 *Confusion matrix* model LSTM 33

Gambar 4.6 Hasil perhitungan *confusion matrix* 34

Gambar 4.7 Prediksi sentimen dari model LSTM 35

Gambar 4.8 Hasil *cleaning* pada tahap n*etwork analysis* 36

Gambar 4.9 Tampilan halaman home 36

Gambar 4.10 Tampilan Persentase Polaritas Sentimen 37

Gambar 4.11 Tampilan hasil *wordcloud* *keyword* teratas 38

Gambar 4.12 Tampilan *wordcloud* positif dan negatif 38

Gambar 4.13 Tampilan hasil *network analysis* 39

Gambar 4.14 Tampilan halaman *login* 39

Gambar 4.15 Tampilan halaman admin 40

Gambar 4.16 Tampilan tambah sentimen 40

Gambar 4.17 Tampilan *database* 41

Gambar 4.18 *database* *sentimenresult* 41

Gambar 4.19 *database* *networkresuilt* 42

Gambar 4.20 Visualisasi dengan NetworkX 44

Gambar 4.21 Visualisasi dengan Gephi 45

Gambar 4.22 Pengambilan data dengan *tool* TwitterScraper 46

Gambar 4.23 Tampilan *dataframe* 47

Gambar 4.24 Hasil setelah *upload* data *file* *csv* 48

Gambar 4.25 Hasil *network analysis* setelah *upload* data *file* csv 49

Gambar 4.26 *Tweet* dari akun @KawalPemilu\_org 50

# DAFTAR RUMUS

Rumus 2.1 Rumus *degree centrality* 9

Rumus 4.1 Rumus *accuracy confusion matrix* 34

# DAFTAR LAMPIRAN

Lampiran 1 *Script* Program Flask 60

Lampiran 2 *Script* program *sentimental analysis* 63

Lampiran 3 *p*rogram *network analysis* 66

**SURAT PERNYATAAN**

Saya yang bertanda tangan di bawah ini:

Nama : Syair Muharram

NIM : 426 19 022

menyatakan dengan sebenar-benarnya bahwa segala pernyataan dalam skripsi ini yang berjudul **Implementasi Sentimental Analysis dan Network Analysis Opini Pengguna Twitter Pada Isu Politik Berbasis Web** merupakan gagasan dan hasil karya saya sendiri dengan arahan komisi pembimbing, dan belum pernah diajukan dalam bentuk apapun pada perguruan tinggi dan instansi manapun. Semua data dan informasi yang digunakan telah dinyatakan secara jelas dan dapat diperiksa kebenarannya. Sumber informasi yang berasal atau dikutip dari karya yang diterbitkan dari penulis lain telah disebutkan dalam naskah dan dicantumkan dalam skripsi ini.

Jika pernyataan saya tersebut diatas tidak benar, saya siap menanggung resiko yang ditetapkan oleh Politeknik Negeri Ujung Pandang.

Makassar, September 2023

Syair Muharram

426 19 022

**IMPLEMENTASI *SENTIMENTAL ANALYSIS* DAN *NETWORK ANALYSIS* OPINI PENGGUNA TWITTER PADA ISU POLITIK BERBASIS WEB**

**RINGKASAN**

Skripsi ini disusun dengan tujuan untuk membantu menganalisis opini publik dengan menggunakan teknik *sentimental analysis* dan *network analysis* yang bertujuan menganalisis kecenderungan polaritas opini publik di media sosial Twitter atau X dan untuk memetakan keterhubungan akun media sosial dengan akun media sosial lainnya.

Arsitektur *machine learning* yang digunakan dalam penentuan *sentimental analysis* yaitu metode LSTM, yang memiliki keunggulan dalam penanganan kata atau frasa teks yang panjang.

Alur kerja dalam penyusunan skripsi ini dibagi dalam dua alur yaitu alur pengerjaan *sentimental analysis*, dan alur kerja *network analysis*. Untuk alur kerja *sentimental analysis* terdiri atas tahap pengambilan data *tweet*, *preprocessing*, *labeling*, dan penyusuan model LSTM. Untuk alur kerja *social network* yaitu, pengambilan data *tweet*, *preprocessing*, dan pemetaan *network analysis*.

Adapun untuk implementasi ke dalam bentuk *website* menggunakan *library* dari bahasa pemrograman Python, yaitu Flask untuk menangani *function* yang melibatkan pengolahan data *tweet* menjadi hasil *sentimental analysis* dan *network analysis*, dan NodeJs untuk menangani *backend* dari *website*.

# PENDAHULUAN

## Latar Belakang

Majunya perkembangan teknologi informasi saat ini membawa banyak keuntungan dan manfaat bagi masyarakat. Keuntungan dan manfaat yang didapatkan antara lain memudahkan masyarakat untuk berinteraksi dengan banyak orang, memperluas relasi, jarak dan waktu bukan lagi masalah, lebih mudah dalam mengekspresikan diri, dan penyebaran informasi dapat berlangsung secara cepat.

Berdasarkan hasil dari penelitian Wearesocial Hootsuite, Pada bulan Januari 2019 mencapai 150 juta dari total populasi untuk pengguna media sosial di Indonesia, sedangkan untuk pengguna media sosial gadget mencapai 130 juta atau sekitar 48% dari total populasi yang ada (Izzulsyah dkk,. 2022:21). Pada Januari 2022, jumlah pengguna media sosial di Indonesia mencapai 191,4 juta. Jumlah pengguna jejaring sosial di Indonesia pada awal tahun 2022 tersebut setara dengan 68,9% dari total penduduk di Indonesia, jumlah pengguna jejaring sosial di Indonesia meningkat sebanyak 21 juta atau 12,5% dari tahun 2021 hingga tahun 2022 (Fauziah, 2023:1).

Pengguna sosial media dapat mengekspresikan opininya pada berbagai bidang isu, misalnya pada isu politik atau isu–isu lainnya pada *platform* media sosial. Adapun media sosial yang dipakai untuk dijadikan wadah beropini sesorang yaitu X atau yang sebelumnya bernama Twitter. Twitter atau X memiliki struktur postingan yang cukup memudahkan seseorang dapat memberikan opininya. Selain itu pada sisi pihak pengembang aplikasi Twitter menawarkan layanan API (*Application Programming Interface*) yang memungkinkan para pengembang aplikasi untuk mendapatkan data Twitter secara langsung untuk pengolahan data lebih lanjut. Hal tersebut membuat Twitter menjadi salah satu *platform* yang kerap digunakan untuk proses *data* *mining*, salah satu penerapan *data mining* yaitu *sentimental analysis* dan *network analysis*. Dikarenakan kebijakan yang dikeluarkan dari Twitter atau X per tanggal 9 Februari 2023 terkait penggunaan API menjadi terbatas dan berbayar maka dari itu langkah pada penelitian ini menggunakan *tools* *scraping*.

*Sentimental analysis* merupakan proses untuk mengetahui suatu kalimat mengandung tendensi positif, negatif atau netral (Handayanto dkk, 2021:154). *Sentimental analysis* merupakan salah satu cabang dari NLP (*Natural Language Processing*) yang berfungsi untuk mengklasifikan emosi atau sentimen orang – orang berdasarkan data teks yang didapat dari *tweet*, ulasan produk, ataupun komentar. *Sentimental analysis* digunakan untuk mengetahui seberapa cenderung opini seseorang terhadap suatu isu atau permasalahan, dengan titik kecenderungan berupa, positif, negatif atau netral.

*Social network analysis* adalah analisis yang digunakan untuk merepresentasikan hubungan antar beberapa orang, komunitas, atau perusahaan dengan menggunakan teknik analisis grafik (Selisker, 2017:212). Penerapan *social network analysis* dalam suatu aplikasi yang mampu menggambarkan relasi atau hubungan antar individu dengan melakukan visualisasi dalam bentuk graf kiranya dapat membantu proses pemecahan masalah yang ada. Selain itu, akan dilakukan proses kalkulasi terhadap setiap relasi antar individu untuk menemukan *centrality* dari sebuah jejaring sosial yang didasarkan pada posisi masing-masing individu yang terkait dalam struktur jaringan. Kedua analisis ini dapat digabungkan untuk mengevaluasi opini publik dari topik tertentu dan bagaimana itu terhubung dengan interaksi sosial. Namun sayangnya, untuk saat ini masyarakat kekurangan aplikasi untuk membantu dalam hal yang telah disebutkan diatas.

Oleh karena itu peneliti memilih judul implementasi *sentimental analysis* dan *social network analysis* opini pengguna Twitter pada isu politik berbasis *web*

## Rumusan Masalah

Berdasarkan latar belakang di atas, dapat dirumuskan permasalahan dalam penelitian ini sebagai berikut:

1. Bagaimana implementasi *sentimental analysis* dan *social network analysis pada data tweet* terhadap isu politik
2. Bagaimana visualisasi data *sentimental analysis* dan *social network analysis*

## Ruang Lingkup Penelitian

Untuk pembahasan topik yang lebih terarah pada tujuan yang diinginkan pada penelitian ini, maka penelitian ini diberikan batasan sebagai berikut:

1. Data *tweet* yang digunakan yakni *tweet* yang berbahasa Indonesia.
2. Opini publik pada data *tweet* yakni berkaitan dengan isu politik.
3. Metode yang digunakan dalam *sentimental analysis* yaitu metode LSTM.
4. Jumlah data *tweet* yang dipakai sebanyak 2600 data *tweet.*
5. Menampilkan hasil *sentiment analysis* dan *network analysis* dalam bentuk gambar.

## Tujuan Penelitian

Adapun tujuan penelitian yang dicapai penulis antara lain:

1. Untuk mengimplementasikan *sentimental analysis* dan *network analysis* pada data *tweet.*

2. Untuk memvisualisasikan data *sentimental analysis* dan *network analysis.*

## Manfaat Penelitian

Penelitian ini diharapkan dapat bermanfaat bagi penulis maupun pembaca tentang gambaran *sentimental analysis* dan *network analysis* pada data Twitter. Selain itu penelitian ini bermanfaat untuk masyarakat yang ingin mengetahui tentang *sentimental analysis* dan *network analysis* terkait isu-isu politik.

# **TINJAUAN PUSTAKA**

## Penelitian Terdahulu

Primandani Arsi dan Retno Waluyo dalam jurnalnya yang berjudul *Analisis Sentimen Wacana Pemindahan Ibu Kota Indonesia Menggunakan Algoritma* *Support Vector Machine (SVM).* Dalam penelitian tersebut diusulkan metode Sup*port Vector Machine (*SVM) untuk diterapkan pada *tweets* topik pemindahan ibu kota Indonesia untuk tujuan klasifikasi kelas sentimen pada media sosial Twitter. Berdasarkan hasil dari proses pengujian yang telah dilakukan terhadap *tweets* sentimen pemindahan ibu kota dari media sosial Twitter sebanyak 1.236 *tweets* (404 positif dan 832 negatif) menggunakan SVM maka hasil hipotesa awal sesuai hasil akhir yakni SVM lebih baik dari metode sebelumnya (BM25 + KNN dan *Naive Bayes*).

Jurnal dengan judul *Implementasi* *Social Network Analysis* *Dalam Penyebaran Informasi Virus Corona (Covid-19) Di Twitter* yang ditulis oleh Dwi Inayah dan Fredy Law Purba. Metode yang digunakan adalah *wordcloud analysis*, *sentiment analysis*, dan *social network analysis*. Hasil *wordcloud analysis* menunjukkan bahwa kata yang sering muncul terkait Covid-19 adalah “positif”, “pandemi” dan “Indonesia”. Hasil *sentiment analysis* menunjukkan bahwa cuitan bersentimen netral merupakan yang terbanyak, disusul cuitan bersentimen negatif, kemudian bersentimen positif.

Nico Munasatya dan Sendi Novianto dalam jurnal dengan judul *Natural Language Processing Untuk Analisis Sentimen Presiden Jokowi* *Menggunakan* *Multi-Layer Perceptron*. Hasil Pengujian pada jurnal tersebut menunjukan bahwa akurasi algoritma *multi-layer perceptro*n dengan jumlah *hidden layer*, *learning rate, shuffling* dan *drop o*ut berpengaruh dari nilai akurasi yang dihasilkan.

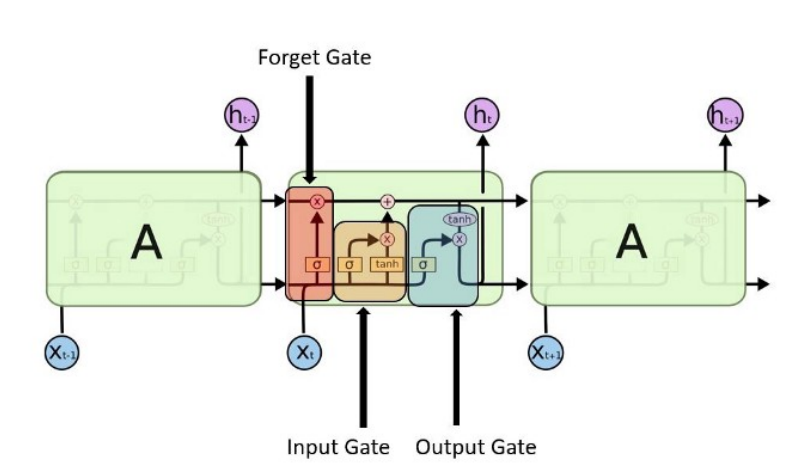
## *Natural Language Processing*

NLP atau *Natural Language Processing* adalah disiplin ilmu komputer yang bertujuan untuk memahami konsep dan maksud dari bahasa manusia. Sementara manusia cukup mahir memahami sintaks linguistik dan tata bahasa serta hubungan spasial tersirat, komputer memiliki kesulitan besar pengolahan *query* bahasa alami (Allen, 1995:10). Contoh *tools* yang disediakan pada Python adalah *Natural Language Toolkit (*NLTK) dan CoreNLP.

Tujuan NLP adalah merancang dan membuat aplikasi yang dapat memfasilitasi manusia untuk berinteraksi dengan mesin dan perangkat lain melalui penggunaan bahasa alami (Putra dan Akhmad Budi. 2020:2).

## LSTM

LSTM atau *Long Short Term Memory* adalah metode yang dapat digunakan untuk mempelajari suatu pola pada data deret waktu (Ningrum dkk,. 2021:539). LSTM merupakan salah satu bentuk dari *Recurrent Neural Network* atau RNN yang dapat melakukan pembelajaran pada dependensi jangka panjang (*long-term dependencies*) dikarenakan kemampuan dari LSTM yang memiliki pengatur memori dari tiap neuron (Ningrum dkk,. 2021:540). Model ini diperkenalkan oleh Hochreiter dan Schmidhuber pada tahun 1997

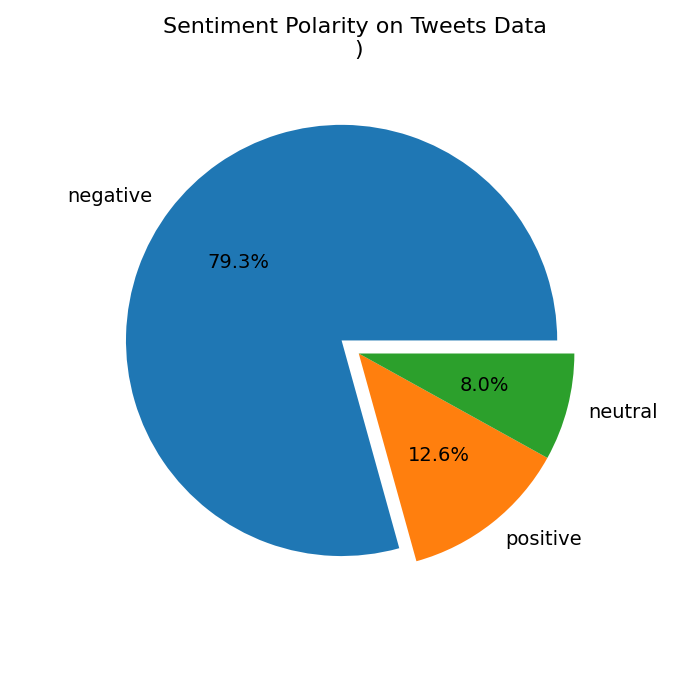


Gambar 2.1 Arsitektur LSTM

Pada *Long Short Term Memory* atau LSTM terdapat modifikasi pada RNN dengan menambahkan *memory cell* atau *memory unit* yang dapat menyimpan informasi yang dipelajari LSTM dalam jangka waktu yang panjang. LSTM memberikan solusi untuk mengatasi terjadinya *vanishing gradient* pada RNN saat memproses *data sequential* yang panjang.

## *Sentimental Analysis*

*Sentimental analysis* adalah proses komputasi untuk mendeteksi sentimen, opini, dan emosi terhadap suatu objek dalam kontekstual teks (Soemedhy dkk,. 2022:80). Hal tersebut berfungsi untuk menentukan apakah suatu teks tertentu positif, negatif, atau netral. S*entimental analysis* juga dapat dikatakan sebagai *opinion mining*, karena hasil yang diperoleh berasal dari opini atau kelakuan pembicara. Untuk analisis ini, opini yang dikumpulkan dari *user*, yang dapat digunakan untuk penelitian lebih lanjut. Media sosial bertindak sebagai sebuah media dimana *user-*nya dapat mengunggah banyak pendapat atau opini dalam sehari dan *microblog.* Berbagai aspek tersebut dapat digunakan untuk klasifikasi semtimen. Banyak penelitian yang dilakukan di bidang *sentimental analysis* karena signifikansi dalam persaingan tingkat pemasaran dan perubahan kebutuhan masyarakat. Analisis sentimen memerlukan pemakaian seperangkat pelatihan untuk kinerjanya, dan kualitasnya memainkan peran penting dalam evaluasi teks yang akurat.



Gambar 2.2 Contoh diagram pada *sentimental analysis*

## *Social Network Analysis*

*Social network analysis* merupakan suatu jejaring yang menggambarkan struktur dalam situasi sosial jejaring sosial menggunakan *nodes* sebagai representasi dari manusia atau sekelompok manusia dan *edges* sebagai representasi dari interaksi sosial diantara dua manusia. Interaksi sosial ini dapat berupa hubungan pertemanan, hubungan kekeluargaan, hubungan professional, pertukaran barang, pertukaran uang, pola komunikasi, hubungan romantika, dan berbagai jenis hubungan lainnya.

*Social Network Analysis* atau SNA dapat dibentuk untuk merepresentasikan berbagai kondisi dan konteks dalam dunia nyata. *Social network* direpresentasikan dengan menggunakan pendekatan teori graf, yang merupakan cabang dari ilmu matematika. *Social network* *analysis* dapat dibentuk ketika terdapat entitas yang berinteraksi dengan entitas - entitas lainnya dalam suatu lingkup sosial. Entitas ini dapat mewakili berbagai hal seperti manusia, perusahaan, ataupun organisasi.

SNA ini sering digunakan untuk menentukan aktor sentral di dalam sebuah jaringan dengan menghitung nilai *centrality* (Tabassum et al., 2018:2). *Centrality* yang digunakan pada penelitian ini yaitu *degree centrality. Degree centrality* merupakan jumlah interaksi atau *edge* yang terjadi pada sebuah *node* tertentu. Rumus dari *degree centrality* adalah :

............................................................................... (2.1)

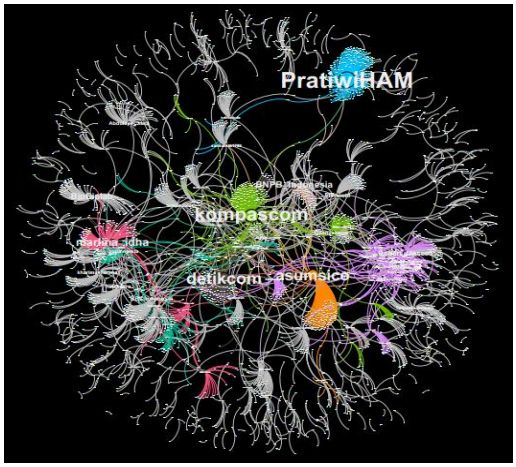
dimana:

CD = *Degree centrality* *node* i

aij = nilai baris ke-i dan kolom ke-j matriks *adjacency*

Sebagai contoh seperti pada gambar 2.3, label “*PratiwiHam*” termasuk dalam klaster yang cukup padat terlihat banyak *node* yang berkumpul di satu area membentuk kelompok. Setelah ditelusuri pada akun Twitter yang bersangkutan ternyata merupakan akun dari pasien yang terkonfirmasi positif yang berhasil sembuh dari virus Covid-19 dan akun tersebut sering membuat *thread* berisi kisah hidupnya selama menderita Covid-19 yang mana *threads* yang dibuat oleh @PratiwiHAM mendapat banyak *retweet* dan *likes* dari pengguna Twitter (Inayah dkk, 2020:297).

Selain akun pengguna biasa, akun berita resmi seperti @detikcom, @kompascom dan @asumsico serta akun resmi pemerintah @BNPB\_Indonesia juga memiliki peran penting dalam penyebaran informasi terkait Covid-19 di Twitter. Hal ini tidak terlepas dari seringnya akun-akun tersebut membuat *tweet* berkaitan dengan perkembangan dan informasi-informasi resmi dan terpercaya mengenai Covid-19 (Inayah dkk, 2020:297).



Gambar 2.3 Contoh implementasi *Social Network Analysis* pada penyebaran informasi covid-19.

## Twitter

Twitter adalah layanan pengiriman pesan yang memiliki banyak kesamaan dengan alat komunikasi yang sekarang ini digunakan. Twitter memiliki elemen yang mirip dengan *email*, IM, SMS, *blogging*, RSS, dan sebagainya (Undap dkk,. 2021:40). Komunikasi perpesanan pada media sosial Twitter disebut *tweet.* *Tweet* yang dapat dikirimkan di Twitter memiliki maksimal 280 karakter. Pesan-pesan di Twitter dapat bersifat publik, yang artinya sangatlah mungkin bagi banyak orang dapat membaca pesan yang dibuat oleh pengirim.

Twitter berulang kali mengubah kebijakan dalam persoalan API pada *platform*nya. Untuk saat ini per tanggal 9 Februari 2023 Twitter mengeluarkan kebijakan baru yakni menghapus akses gratis dalam pengambilan data *tweet* baik itu API v1.1 dan v2.

## *Text Mining*

*Text mining* merupakan suatu teknik dalam ilmu komputer yang digunakan untuk memecahkan permasalahan informasi yang sangat banyak dengan mengkombinasikan teknik dari *data* *mining* Seperti pada *data mining, text mining* berusaha mengekstrak informasi yang bermanfaat dari sumber data melalui identifikasi dan eksplorasi pola menarik. Pada *text mining* sumber data adalah kumpulan dokumen, ini berarti dapat berupa koran, majalah, artikel, surat, ataupun laporan-laporan penelitian seperti jurnal , tugas akhir, atau tesis (Feldman dan Sanger, 2007:42). Dengan *text mining* dapat diketahui pengelompokkan dokumen yang serupa, memperkirakan penulis dari sebuah dokumen, atau mengklasfikasikan dokumen ke dalam suatu kategori. Tetapi tidak seperti data yang pada *data mining* yang umumnya sudah terstruktur, dokumen merupakan data yang tidak terstruktur. Proses *text mining* sama dengan tahapan *Knowledge Discovery in Database* (KDD) pada *data mining.*

Berikut ini adalah proses *text mining* mengacu pada KDD antara lain:

a. *Text Data*

Dokumen adalah syarat utama dalam *text mining*. Data berupa teks merupakan suatu fragmen yang dianggap sebagai unit. Unit ini dapat berupa buku, paragraf, abstrak, atau pun judul. Untuk *web*, fragmen teks adalah halaman *web*.

b. *Text Preprocessing*

Pada tahap ini dilakukan pembersihan data dan tokenisasi. Pembersihan data dilakukan untuk menghilangkan informasi yang tidak perlu atau tidak diinginkan seperti tabel, gambar atau rumus-rumus. Pada tahap tokenisasi, dokumen diperlakukan sebagai bentuk *string* dan dipecah menjadi token. Pada token kalimat-kalimat yang ada pada dokumen dipisahkan setiap kata dan karakter yang membentuknya. Dengan membentuk token, maka dapat dilakukan analisis teks lebih lanjut (Bhumika, et al, 2013:91). Kata-kata yang bersifat *stopword* juga dihilangkan. Kata-kata *stopword* seperti “*a*”, “*the*”, dan ”*but*” adalah yang dibutuhkan pada struktur *grammer* dalam bahasa Inggris tetapi tidak memiliki arti dan tidak dibutuhkan dalam *text mining*. Pada tahap ini juga dilakukan proses *stemming* yang mengubah kata-kata dalam bentuk dasar, misal: *connection* menjadi *connect*.

c. *Feature Transformation (Attribute Generation)*

Sebuah dokumen teks diwakili oleh kata-kata (fitur) yang dikandungnya dan kejadiannya. Representasi dokumen merupakan salah satu teknik yang digunakan untuk mengurangi kompleksitas dokumen sehingga lebih mudah untuk ditangani. Dokumen harus ditransformasikan dari versi teks lengkap ke dalam bentuk *vector* dokumen. Representasi dokumen yang umum digunakan disebut *vector* *space* model dimana dokumen direpresentasikan menjadi vektor dari kata-kata. Beberapa keterbatasannya adalah: representasi dimensi yang tinggi, hilangnya korelasi dengan kata-kata yang berdekatan dan hilangnya hubungan semantik yang ada didalam dokumen. Untuk mengatasi masalah ini, metode pembobotan kata digunakan untuk menetapkan bobot yang sesuai dengan kata tersebut. Pada model ruang vektor, dokumen diwakili oleh vektor kata-kata karena kata-kata yang membentuk dokumen akan menentukan isi dari dokumen tersebut (Bhumika et al, 2013:93).

d. *Attribute Selection / Feature Selection*

Tahap ini adalah teknik pengurangan dimensi yang efektif untuk menghilangkan fitur *noise*. Fitur *noise* merupakan informasi-informasi yang tidak berguna yang dapat mengganggu hasil penelitian dalam *text mining*, seperti tulisan *copyright*, menu navigasi pada halaman *web*, dan lain-lain (Ting et al, 2011:39).

Pemilihan fitur juga dikenal sebagai pemilihan variabel, adalah proses pemilihan *subset* dari fitur yang penting untuk digunakan dalam pembuatan model. Hal ini dilakukan karena data mengandung banyak fitur yang berlebihan atau tidak relevan. Misalkan fitur yang berulang adalah salah satu yang tidak memberikan informasi tambahan. Fitur yang tidak relevan tidak memberikan informasi yang berguna atau relevan dalam konteks apapun. Pemilihan fitur dilakukan dengan menyimpan kata-kata dengan bobot tertinggi sesuai dengan ukuran yang telah ditentukan terhadap pentingnya kata tersebut.

e. *Mining Pattern*

Pada tahap ini teknik yang digunakan *text mining* digabungkan dengan teknik klasik pada *data mining*, seperti klasifikasi, pengelompokkan atau asosiasi. Teknik-teknik tersebut dapat mengolah data yang telah terstruktur sebagai hasil pengolahan dari tahapan *text mining* yang telah dilakukan.

f. *Evaluation*

Pada tahap ini dilakukan evaluasi dari hasil *mining pattern*. Hasil evaluasi umumnya menggunakan suatu nilai performansi sesuai dengan teknik data *mining* yang digunakan.

## Python

Bahasa Python dibuat pertama kali oleh Guido Van Rossum. Bahasa ini termasuk dalam bahasa pemrograman tingkat tinggi yang mudah dipelajari karena struktur bahasa yang digunakan rapi dan mudah untuk dimengerti. Tingkat fleksibilitas dan pengembangan Python sesuai untuk digunakan dalam memecahkan berbagai masalah kompleks dengan lebih efisien termasuk *artificial intelligence*, *machine learning*, dan berbagai aktivitas *data* *science* lainnya (Alamsyah dan Dian Puteri Ramadhani, 2020:37).

Beberapa pengguna menjalankan bahasa Python hanya dari teks *editor* biasa namun untuk meningkatkan kemudahan banyak pengguna bahasa Python menggunakan suatu *integrated development environment* (Alamsyah dan Dian Puteri Ramadhani, 2020:38).

Python diklaim sebagai bahasa yang menggabungkan kapabilitas, kemampuan, dengan sintaksis kode yang sangat jelas, dan dilengkapi dengan fungsionalitas pustaka standar yang besar serta komprehensif. Python dapat digunakan untuk berbagai keperluan pengembangan perangkat lunak dan dapat berjalan di berbagai *platform* sistem operasi.

Python didistribusikan dengan beberapa lisensi yang berbeda dari beberapa versi. Namun pada prinsipnya, Python dapat diperoleh dan dipergunakan secara bebas, bahkan untuk kepentingan komersial. Menurut Syahruddin dkk. (2018:2), beberapa fitur yang dimiliki oleh Python diantaranya sebagai berikut:

1. Memiliki kepustakaan yang luas. Dalam distribusi Python telah disediakan modul-modul;
2. Memiliki tata bahasa yang mudah dipelajari;
3. Memiliki aturan *layout* kode sumber yang memudahkan pengecekan, pembacaan kembali dan penulisan ulang kode sumber;
4. Berorientasi objek; dan
5. Dapat dibangun dengan bahasa Python maupun C/C++.

## NLTK

NLTK (*Natural Language Toolkit*) adalah salah satu *library* berbasis metode pembelajaran mesin dalam Python dan alat analisis sentimen. Yang menyediakan dasar untuk pemrosesan teks dan klasifikasi (Kameswari et al., 2019 : 4519). NLTK adalah *library* Python yang dapat memproses teks *natural language* dengan cepat dan mudah. *Toolkit* ini dikembangkan di *University of Pennsylvania* sebagai alat penelitian dan pengajaran untuk pemrosesan bahasa alami. NLTK memiliki sejumlah besar *corpora built-in*, termasuk berbagai jenis materi teks seperti novel, berita, teks obrolan jaringan, ulasan film, dll. termasuk korpus *Brown*, korpus *Gutenberg*, korpus alamat pelantikan, korpus *Reuters*, dll. (Kameswari et al., 2019 : 4520).

Selain itu, NLTK menyediakan antarmuka yang mudah digunakan ke lebih dari 50 kumpulan dan sumber daya leksikal seperti WordNet, bersama dengan rangkaian pustaka pemrosesan teks untuk klasifikasi, tokenisasi, *stemming*, *tagging*, *parsing*, dan penalaran semantik, pembungkus untuk perpustakaan NLP berkekuatan industri, dan forum diskusi aktif. Bekerja sama dengan Python yang kuat perpustakaan standar dan perpustakaan pihak ketiga lainnya, dapat melakukan pemrosesan sekunder dari hasil pemrosesan. Ini menyediakan dukungan yang kuat untuk memproses teks yang kompleks (Li, 2019:276).

## TwitterWebScraper

TwitterWebScraper adalah sebuah *tools scraping* yang di buat oleh WilfredNJH di *github*. *Tools* tersebut dibuat untuk mengumpulkan data Twitter tanpa memerlukan API dari Twitter. Namun *tools* tersebut memiliki kekurangan yang mana merujuk pada peraturan Twitter mengenai batasan dalam pengambilan data *tweet* per harinya. *Output* dari *tools* tersebut akan disimpan dengan format *file* *.csv*. *Tools* tersebut menggunakan *library* selenium dan beautifulsoup4 untuk fungsi *scraping* *tweet*nya, dan untuk GUI nya menggunakan *customtkinter*.

## NetworkX

Networkx adalah salah satu *package* pada bahasa pemrograman Python yang berfungsi untuk mengeksplorasi dan menganalisis jaringan dan algoritma jaringan. Networkx menyediakan berbagai jenis graf untuk menggambarkan jaringan, termasuk *simple graph* dan *directed graph*. Beberapa contoh algoritma graf yang dapat diimplementasikan dengan Networkx adalah *shortest paths, betweenness centrality, closeness centrality,* dan lain sebagainya (Mufidah dkk, 2020:38).

## Visual Studio Code

Visual Studio Code merupakan sebuah aplikasi *editor code open source* yang dirilis oleh Microsoft untuk sistem operasi Windows, Linux, dan MacOs. Visual Studio Code memudahkan dalam penulisan kode yang mendukung berbagai jenis bahasa pemrograman, seperti C++, C#, Java, Pyhton, PHP, GO. Visual Studio Code memiliki kemampuan untuk mengidentifikasi jenis bahasa pemrograman yang digunakan dan memberi variasi warna sesuai dengan fungsi dalam rangkain kode tersebut. Selain itu fitur lainnya adalam kemampuan untuk menambah ekstensi dimana para pengembang dapat menambah fitur yang tidak ada dalam Visual Studio Code (Ramdhan dan Devi Adi Nufriana, 2019:4).

## Flask

Flask merupakan *framework web* yang ditulis dengan menggunakan bahasa pemrograman Python. Flask tergolong ke dalam *micro*-*framework* karena tidak membutuhkan *tools* atau *library* tertentu serta memiliki *database* bawaan. Flask tidak memiliki lapisan abstraksi *database,* namun Flask dapat mendukung ekstensi yang dapat menambahkan fitur pada aplikasi (Singh et al,. 2019: 24894).

Flask menggunakan *Jinja* *template engine* dan *Werkzeug* WSGI *ToolKit*. Struktur Flask dibagi menjadi dua bagian, yaitu *template* *file* yang berisi *template* *jinja* termasuk halaman HTML dan *static* *file* yang berisikan *file* kode statis yang diperlukan pada pembuatan situs *web* seperti kode CSS, kode JavaScript, dan *file* gambar (Aslam *et al*., 2015:54).

## NodeJs

Node.js merupakan sebuah perangkat lunak yang di desain untuk pengembangan perangkat lunak. Node.js sendiri dieksekusi pada sisi *server*. *Platform* ini menggunakan bahasa pemrograman JavaScript. Selain itu, Node.js juga menggunakan teknik *nonblocking* untuk mempercepat prosesnya (Mubariz dkk., 2020:73).

## MySQL

MySQL adalah sebuah program *database server* yang mampu menerima dan mengirimkan datanya sangat cepat, *multi user*, serta menggunakan perintah dasar SQL (*Structured Query Language*) (Indrawan, 2021a: 3).

MySQL merupakan dua bentuk lisensi, yaitu FreeSoftware dan Shareware. MySQL yang biasa digunakan adalah MySQL FreeSoftware yang berada di bawah lisensi GNU/GPL. My SQL merupakan sebuah *database server* yang *free*, artinya bebas menggunakan *database* untuk keperluan pribadi atau usaha tanpa harus membeli atau membayar lisensinya. MySQL pertama kali dirintis oleh seorang *programmer database* bernama Michael Widenius. Selain *database server*, MySQL juga merupakan program yang dapat mengakses suatu *database* MySQL yang berposisi sebagai *server*, yang berarti program kita berposisi sebagai *client*. Jadi MySQL adalah sebuah *database* yang dapat digunakan baik sebagai *client* ataupun sebagai *server* (Indrawan, 2021b: 3).

## Metode Pengembangan *Waterfall*

*System* Development *Life Cycle* (SDLC) merupakan metodologi umum yang digunakan untuk mengembangkan sistem informasi. SDLC terdiri dari beberapa fase yang dimulai dari fase perencanaan, analisis, perancangan, implementasi hingga pemeliharaan sistem. Metode *waterfall* adalah salah satu model SDLC yang sering digunakan dalam pengembangan sistem informasi atau perangkat lunak. Model *waterfall* menggunakan pendekatan sistematis dan berurutan. Tahapan model *waterfall* antara lain *requirement, design, implementation, verification,dan maintenance*. Kelebihan menggunakan metode *waterfall* dalam pengembangan sistem informasi adalah kualitas dari sistem yang dihasilkan akan baik karena pelaksanaannya dilakukan secara bertahap, sementara untuk kekurangannya adalah proses pengembangan sistem membutuhkan waktu yang lama sehingga biaya yang diperlukan juga mahal. Metode *waterfall* cocok digunakan untuk proyek pembuatan sistem baru dan juga pengembangan sistem atau perangkat lunak yang berskala besar (Wahid, 2020:1).

## Pengujian *Black Box*

Metode *blackbox testing* adalah sebuah metode yang dipakai untuk menguji sebuah *software* tanpa harus memperhatikan detail *software*. Pengujian ini hanya memeriksa nilai keluaran berdasarkan nilai masukan masing-masing. Tidak ada upaya untuk mengetahui kode program apa yang *output* pakai (Ningrum dkk., 2019a:126). Proses *black box testing* dengan cara mencoba program yang telah dibuat dengan mencoba memasukkan data pada setiap *form*-nya. Pengujian ini diperlukan untuk mengetahui program tersebut berjalan sesuai dengan yang dibutuhkan (Ningrum dkk., 2019b:126).

*Black box testing* bertumpu pada pengkhususan fungsi dari perangkat. Penguji dapat mengartikan himpunan kondisi masukan dan menjalankan pengujian pada pengkhususan fungsi dari perangkat lunak (Anggana dkk., 2020a:96). Metode pengujian *black box testing* terdiri atas beberapa cara, antara lain adalah *Equivalence Partitioning, Boundary Value Analysis, Comparison Testing, Sample Testing, Robustness Testing,* dan lain-lain (Anggana dkk., 2020b:95).

# METODE PENELITIAN

## Tempat dan Waktu Penelitian

Penelitian ini telah dilaksanakan dengan bertempat di Politeknik Negeri Ujung Pandang dengan waktu pengerjaan selama 7 bulan mulai dari bulan Januari s/d Juli 2023.

## Alat dan Bahan

Adapun alat dan bahan yang digunakan dalam penelitian ini adalah sebagai berikut:

1. *Hardware*

Komputer dengan spesifikasi Intel *Core* i3 dengan RAM 8 GB.

1. *Software*
2. Sistem Operasi Windows 10
3. Visual Studio Code
4. NodeJS
5. *Library* Pandas, Scikitlearn, Tensorflow, NLTK, Twitter Web Scraper, NetworkX, Flask
6. MySQL

## Prosedur Penelitian

Dalam pembuatan penelitian ini, terdapat beberapa tahapan agar metodologi dan perancangan sistem tercapai. Tahap-tahap yang dilaksanakan adalah sebagai berikut:

### **Metode Penelitian dan Tahapan Pengembangan Aplikasi**

Perancangan sistem menggunakan bagan alur atau *flowchart*. *Flowchart* akan dibagi menjadi beberapa *flowchart*, yaitu *flowchart* yang menggambarkan cara kerja aplikasi dan *flowchart* yang akan menjelaskan masing-masing tahap sentimen *analysis* dan *network analysis*.

Chart, diagram

Description automatically generated

Gambar 3.1 Metode *waterfall*

### **Analisis Kebutuhan**

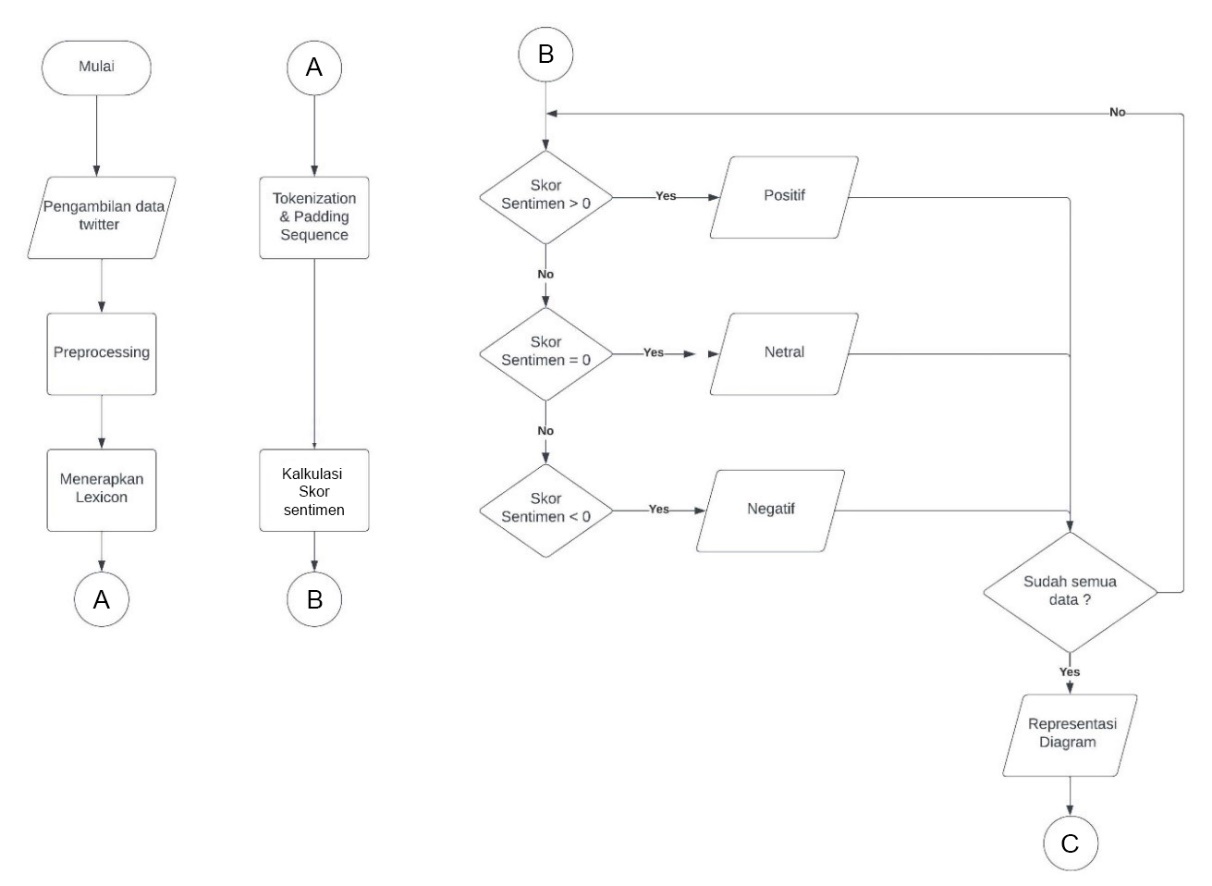
Tahap ini adalah tahap awal dimana dilakukan analisis terhadap kebutuhan apa saja yang diperlukan saat proses pengerjaan berlangsung. Adapun untuk kebutuhan pada proses pengambilan data melibatkan *tools* TwitterWebScraper yang dimana akan menyimpan *file* data ke dalam format .*csv* yang berisikan kolom (*username, handle, postDate, responding, retweet,* dan *likes*).

### **Desain Sistem**

Dalam tahap desain pada penelitian ini menggunakan beberapa pemodelan yaitu*: flowchart, use case diagram,* dan *activity diagram*. Untuk desain *user* *interface* pada penelitian ini penulis menggunakan pemodelan *wireframe*.

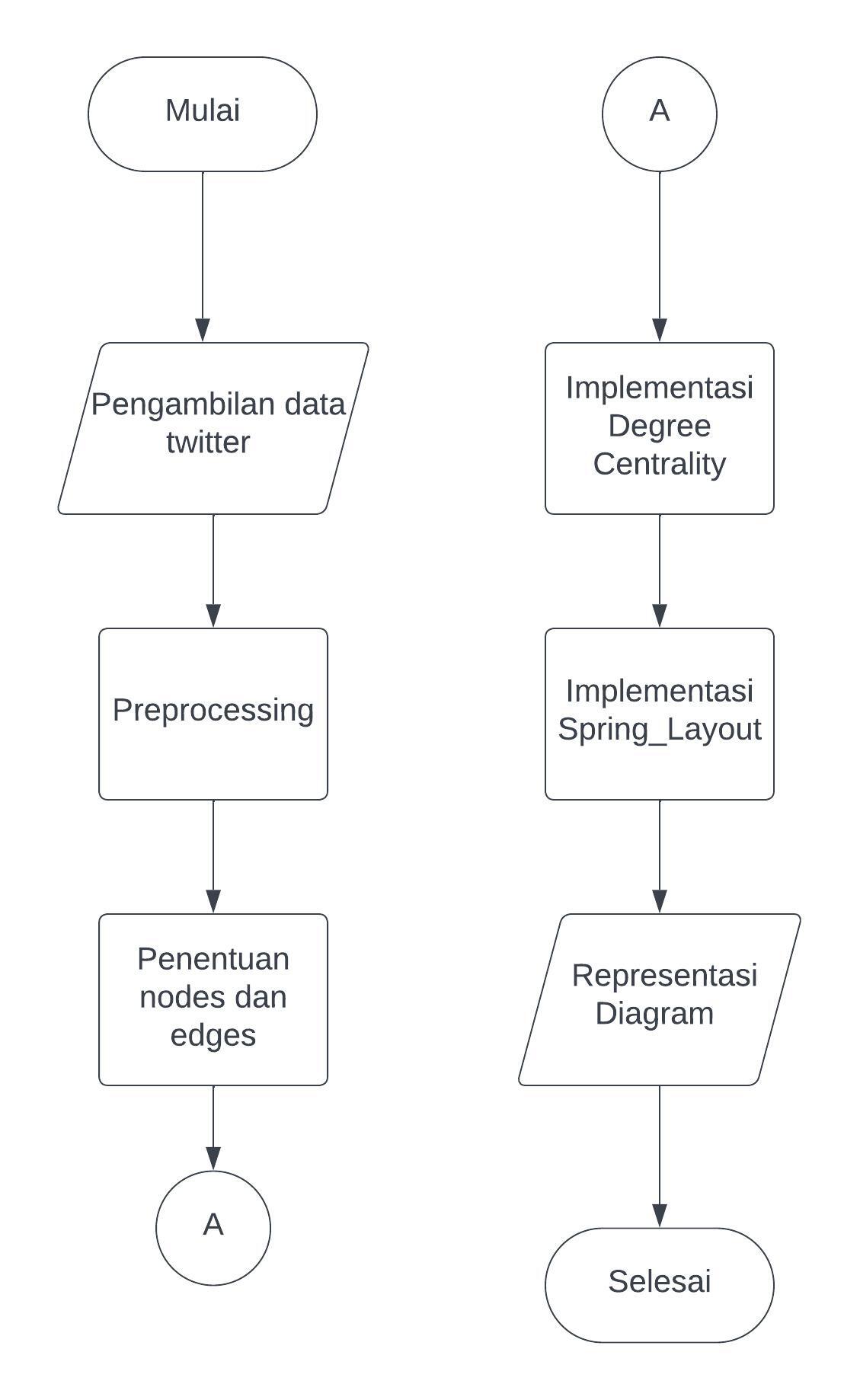
#### *Flowchart*

*Flowchart* berfungsi untuk mengetahui setiap proses yang terjadi. Dalam penelitian ini, penulis membuat 2 *flowchart* dikarenakan ada 2 proses yang berbeda yaitu *flowchart* untuk proses *sentimental analysis* dan *flowchart* untuk proses *network analysis*.



Gambar 3.2 Tampilan *flowchart sentiment analysis*

Berdasarkan gambar 3.2 diatas, langkah pertama diawali dengan pengambilan data Twitter, selanjutnya tahap *preprocessing* yang berisi tahap *cleaning* data, *stemming*, dan *tokenizing*. Setelah itu data hasil *preprocessing* akan diterapkan skor sentimen menggunakan *lexicon*. Selanjutnya hasil *tokenization* dan *padding* *sequence* sebelumnya yang dalam bentuk matriks akan disimpan dengan format .*pickle* yang nantinya akan dipakai untuk membantu model dalam melakukan prediksi sentimen. Langkah selanjutnya melakukan *labeling* dengan kalkulasi skor sentimen pada data yang telah diterapkan menggunakan *lexicon*, dengan *logic* sebagai berikut: jika nilai keseluruhan lebih dari 0 maka akan diberikan label positif, jika nilai keseluruhan sama dengan 0, maka akan diberikan label netral, dan jika nilai keseluruhan kurang dari 0, maka akan diberikan label negatif.

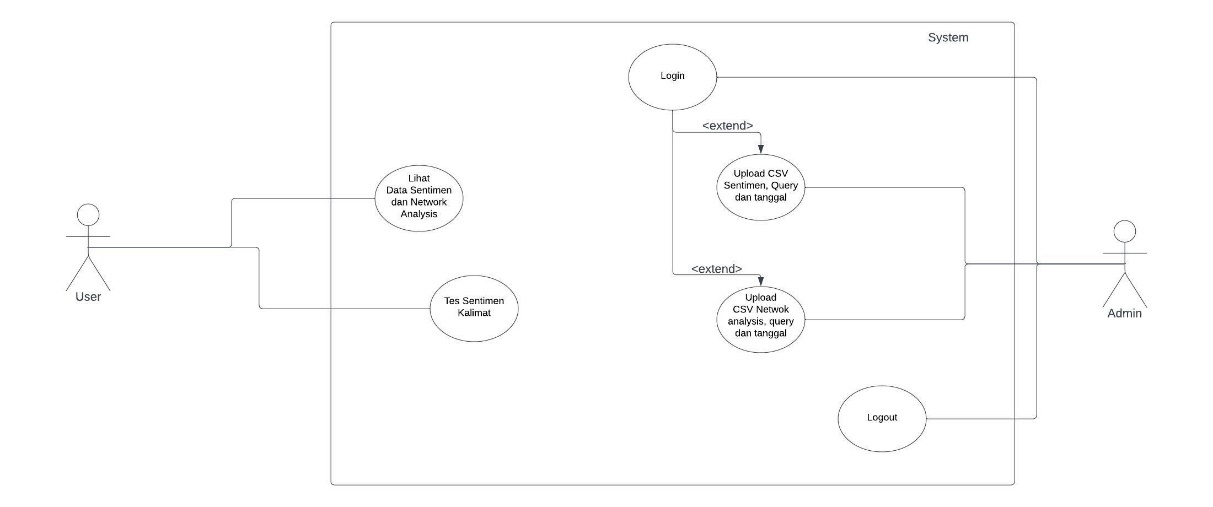


Gambar 3.3Tampilan *flowchart network analysis*

*Flowchart* *network* *analysis* seperti gambar 3.3 diatas memiliki tahap awal yang sama yaitu tahap pengambilan data, selanjutnya memasuki tahap *preprocessing* yang berisikan tahap *cleaning*, selanjutnya menentukan *nodes* dan *edges* pada data yang sudah dilakukan *cleaning.* Langkah selanjutnya akan dilakukan implementasi *degree centrality* dan *spring layout* dengan bantuan *library* Python. Langkah terakhir yaitu melakukan visualisasi.

#### *Use case diagram*

*Use case* diagramdisini berfungsi untuk memperlihatkan fase awal tiap proses dalam sistem yang diterapkan pada penelitian ini. Dimana hal tersebut dapat memudahkan dalam tahap pengembangan nantinya.

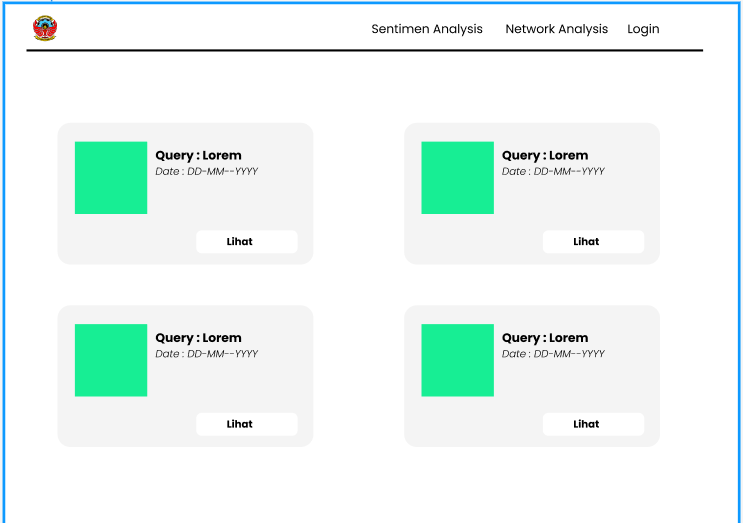


Gambar3.4 *Use case diagram*

Berdasarkan gambar 3.4 diatas *flowchart* memiliki 2 aktor yaitu *user* dan admin, *user* dapat melihat data hasil sentimen dan *network analysis* yang sudah di *upload* oleh admin. Selain itu, *user* dapat melakukan tes pada model LSTM dengan mengetikkan kalimat pada *form* yang disediakan. Admin sendiri dapat melakukan *login* dan memasuki halaman *dashboard*, yang dimana memiliki fitur tambah data pada 2 halaman yaitu halaman untuk *upload* data sentimen dan data *network* *analysis*.

#### *Wireframe*

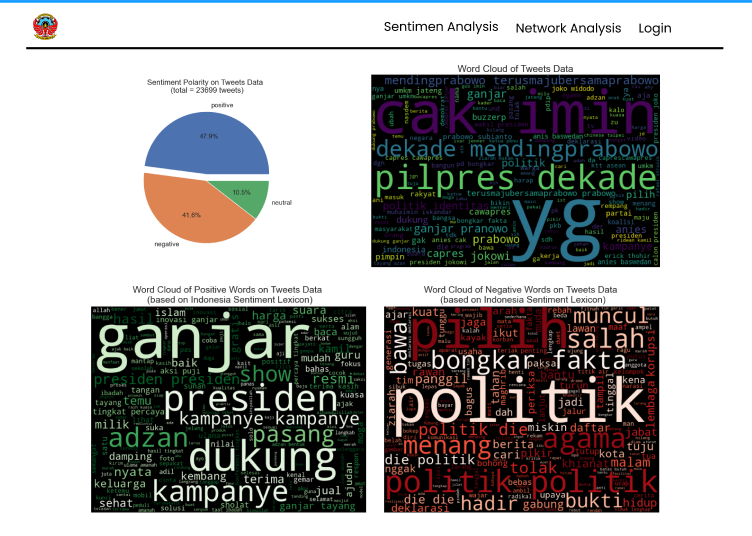
*Wireframe* adalah sebuah desain kerangka tahap awal yang menggambarkan struktur yang akan dibuat nantinya. Desain *wireframe* tersebut nantinya akan dilanjutkan ke tahap pembuatan desain *user* *interface* nantinya.

****

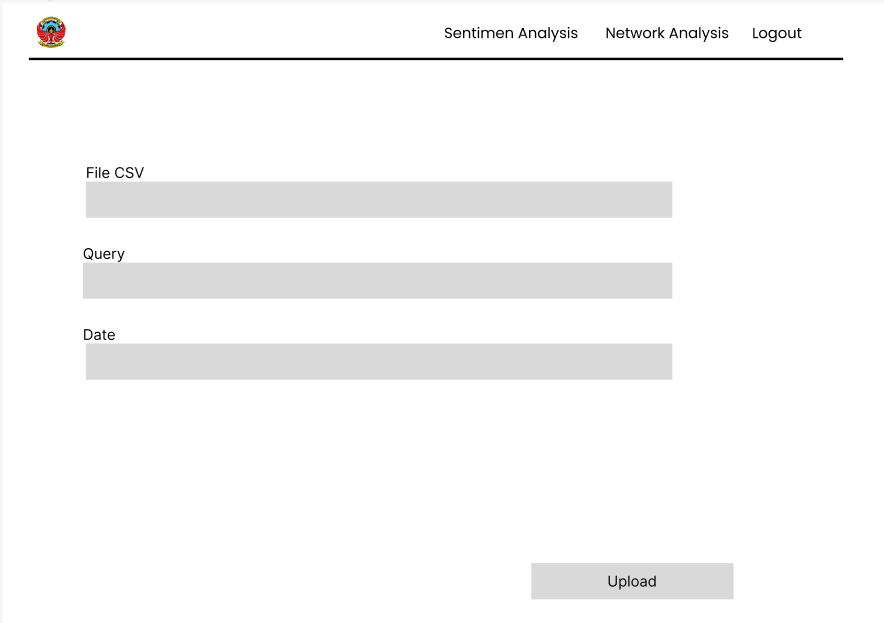
Gambar 3.5 Tampilan halaman *home*

****

Gambar 3.6 Tampilan halaman LSTM *predictor*



Gambar 3.7 Tampilan halaman hasil

****

Gambar 3.8 Tampilan halaman admin

### **Implementasi dan Pengkodean**

Dalam tahap ini penelitian mulai dibangun sesuai dengan analisis kebutuhan. Untuk pengkodeannya akan menggunakan bahasa Python dan NodeJs dengan IDE Google Colaboratory dan Visual Studio Code

### ***Testing***

Pada tahapan ini pengujian program dilakukan dengan menggunakan *blackbox* *testing* dengan harapan bahwa perancangan yang sudah dibuat dapat berjalan dengan sesuai kehendak.

### **Pemeliharaan**

Dalam proses pemeliharaan ini penulis mengupayakan pengembangan sistem yang telah di rancang terkait *software* untuk dapat dibuat maksimal agar aplikasi dapat berjalan dengan baik.

## Pengujian Sistem

Pengujian merupakan tahapan yang cukup berperan penting didalam sebuah perancangan dan pengembangan aplikasi. Adapun pengujian yang akan digunakan adalah menggunakan metode *black* *box* *testing*. *Black* *box* *testing* adalah pengujian yang dilakukan untuk melihat hasil keluaran atau *output* dari aplikasi yang telah dibangun. Pada metode *black box* ini akan dilakukan uji data dan pemeriksaan terhadap fungsional dari aplikasi (Adytio, 2017). Metode ini memfokuskan pada fungsional dari suatu sistem aplikasi.

Pengujian dalam *black box testing* ini lebih mengacu pada bagaimana *user* atau pengguna akhir dapat melakukan interaksi pada sistem aplikasi yang dibuat.

# BAB IV HASIL DAN PEMBAHASAN

**4.1. Hasil**

Penelitian ini menghasilkan sebuah *website* yang dapat membantu dalam prediksi *sentiment* dan *network analysis*, yang dikembangkankan dengan menggunakan Flask dan NodeJS. Pada website ini terdapat 2 Menu utama yaitu: *Home*, dan *Login*. Pada halaman Home terdapat tiga bagian yaitu sentimen *analysis*, *network analysis*, dan LSTM *predictor*. Untuk mengakses menu *login* diperlukan untuk mengetik *url* *localhost:3000/login*. Pada bagian *login* terdapat dua kolom yang dapat diisi oleh admin, yaitu kolom *username* dan kolom *password* untuk masuk ke dalam *dashboard* admin untuk melakukan *input* data. Dengan adanya struktur dan fitur-fitur ini, pengguna dapat mengakses dan memanfaatkan *sentiment* *analysis* dan *network analysis*.

**4.1.1 *Sentiment Analysis***

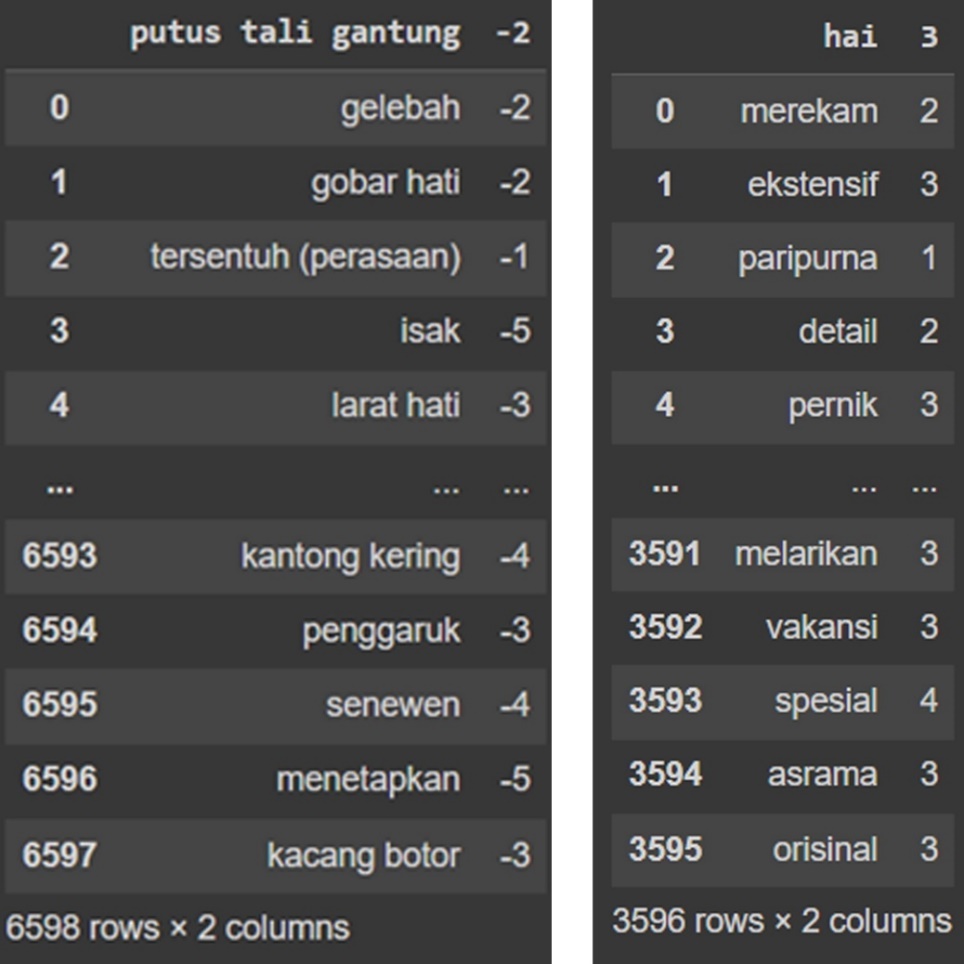
Penelitian ini menghasilkan model LSTM dengan tahap awal mengumpulkan data *tweet* yang di ambil menggunakan *tools* TwitterScraper lalu digabung dan memasuki tahap *preprocessing*. Tahap *preprocessing* tersebut memiliki beberapa *function* yaitu untuk menghilangkan unsur-unsur yang tidak diperlukan misalnya, simbol #, simbol @, angka, *link* *url*, kalimat *replying to*, mengubah *uppercase* menjadi *lowercase* dan selanjutnya akan memasuki tahap *tokenizing*.

Tabel 4.1 Tabel contoh hasil *preprocessing*

|  |  |  |
| --- | --- | --- |
| *Tweet* | Hasil *Cleaning* | Hasil *Tokenizing* |
| Jelang Pengumuman Pemilu, Massa Sempat Bakar Ban di Depan KPU https://t.co/zKNjR29AHB | jelang pengumuman pemilu massa sempat bakar ban di depan kpu | ['jelang', 'umum', 'milu', 'massa', 'bakar', 'ban', 'kpu'] |
| @prabowoisme Berikan suara saat pemilu adalah terserah kita, mayoritas dukung KitaAKU Kamu02JUARANYA untuk mewujudkan NKRI jadi sejahtera. Kemenangan Rakyat Maju | berikan suara saat pemilu adalah terserah kita mayoritas dukung kitaaku kamujuaranya untuk mewujudkan nkri jadi sejahtera kemenangan rakyat maju | ['suara', 'milu', 'serah', 'mayoritas', 'dukung', 'kitaaku', 'kamujuaranya', 'wujud', 'nkri', 'sejahtera', 'menang', 'rakyat', 'maju'] |

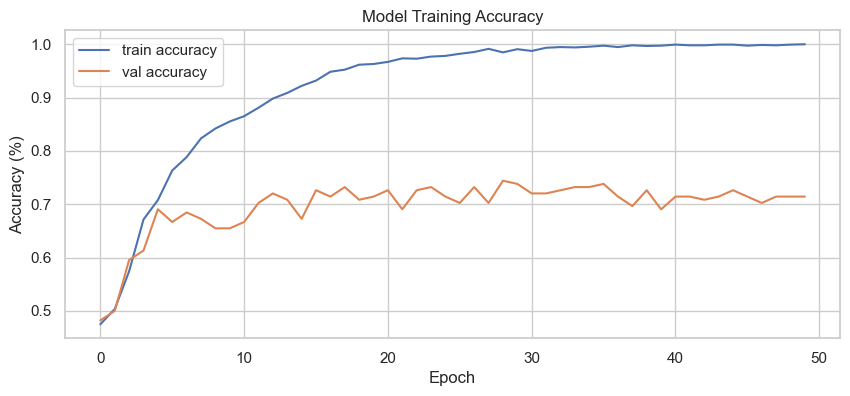
Pada tahap *tokenizing* kalimat akan dipecah menjadi bagian yang lebih kecil yamg dikenal dengan istilah ‘*token*’. Token ini dapat berupa kata atau karakter. Tujuan dari tahap ini yaitu untuk memudahkan proses analisis kalimat. Untuk selanjutnya menuju tahap *labeling*.

Tahap *labeling* pada penelitian ini menggunakan *lexicon*. *Lexicon* merupakan kumpulan kata yang memiliki bobot. *Lexicon* yang digunakan dalam penelitian ini merupakan *lexicon* *InSet*, yang merupakan hasil dari penelitian dari Fajri Koto, and Gemala Y. Rahmaningtyas pada tahun 2017. *Lexicon* pada penelitian tersebut memiliki jumlah 3.597 kata positif dan 6.599 kata negatif dan memiliki bobot dari -5 sampai +5. Merujuk pada jurnal dari penelitian tersebut, bobot yang dicantumkan dalam *lexicon* tersebut didapat dari hasil *labeling* manual dan hasil penerjemahan *lexicon* bahasa Ingggris. Hasil dari *lexicon* tersebut lalu di tingkatkan dengan menambahkan sejumlah *set* *stemming* kata dan sinonim. Seperti pada gambar 4.1 dibawah, gambar bagian kiri merupakan *lexicon* negatif dan gambar bagian kanan merupakan *lexicon* positif.



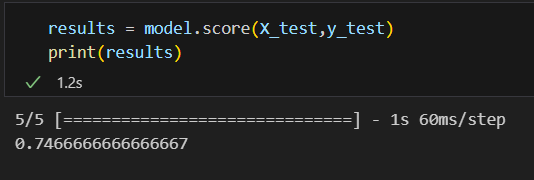
Gambar 4.1 *Lexicon* positif dan negatif dari *InSet*

Langkah selanjutnya yaitu membangun model LSTM dengan membagi *dataset* yang berjumlah 2625 menjadi *data* *training* dan *data* *test* dengan perbandingan 80:20 sedangkan untuk parameter modelnya sebagai berikut: *dropout* = 0.2, *embed\_dim* = 32, *hidden* *unit* = 16, *optimizer* = Adam, *learning* *rate* = 0.01, *epoch* = 50, *batch\_size* = 128, dan *validation*\_*split* = 0.1. Dengan parameter tersebut didapatkan akurasi *training* sebesar 99% dan *validation* 72%.



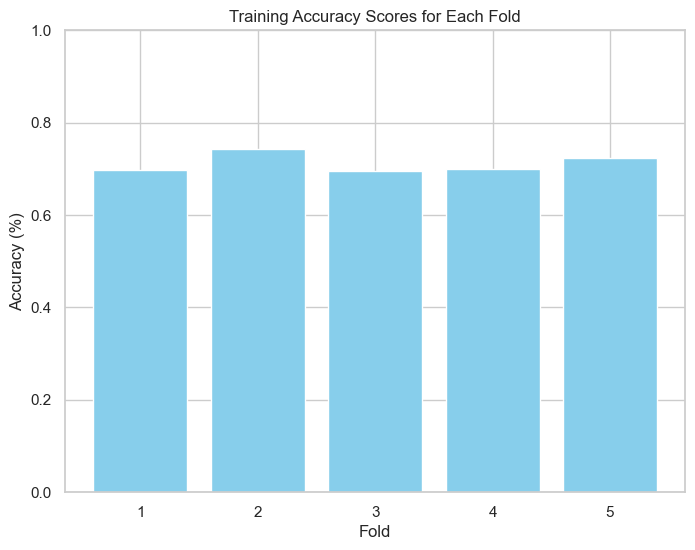
Gambar 4.2 Akurasi *training* model LSTM

Berdasarkan model yang telah di-*training* selanjutnya akan diterapkan pada data *test* yang berjumlah 525 data dengan *model.score(X\_test,y\_test)* didapatkan akurasi *testing* sebesar 74% sesuai gambar 4.3 dibawah.

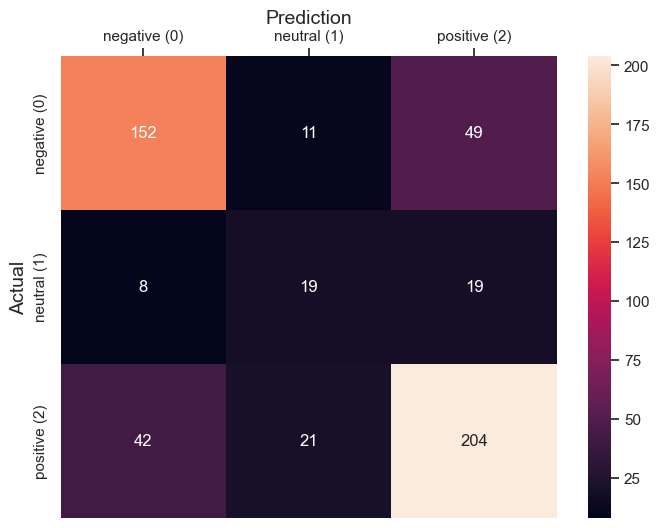


Gambar 4.3 Akurasi *testing* model LSTM

Dapat disimpulkan dari hasil *training* dan *testing* yang didapatkan bahwa model mengalami *overfitting*, maka dari itu langkah selanjutnya mengatasi *overfitting* tersebut dengan menggunakan metode K-Fold *cross validation* pada proses *training* model dengan parameter *n\_splits=5, shuffle=True,* dan *random\_state=42,* didapatkan hasil dari 5 *fold training* memiliki rata-rata akurasi sekitar 71% yang didapat dari hasil penjumlahan akurasi tiap *fold* dibagi denganjumlah *fold* seperti gambar 4.4 dibawah.



Gambar 4.4 Akurasi *training* setelah penerapan *K-Fold cross validation*



Gambar 4.5 *Confusion matrix* model LSTM

Gambar 4.5 diatas merupakan hasil *confusion matrix* dari model LSTM yang telah di-*training*, Dimana dalam perhitungan *confusion matrix* menggunakan data *test* sebanyak 525 dimana didapatkan dari hasil dari perbandingan 80:20 sebelumnya. Untuk rumus *accuracy* dari tiap kelas seperti berikut:

……………………………………………… (4.1)

Dimana :

TP adalah *True Positive.*

TN adalah *True Negative.*

FP adalah *False Positive.*

FN adalah *False Negative.*

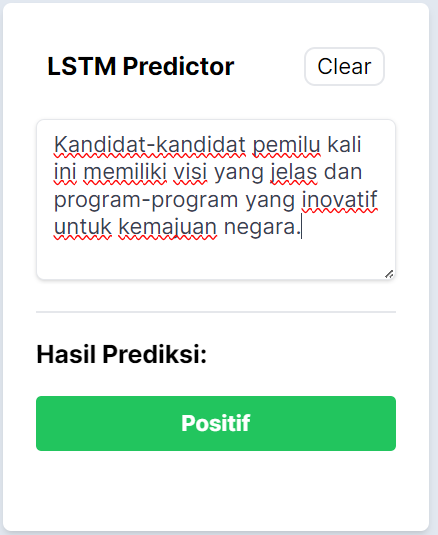
*Accuracy* adalah hasil dari perhitungan TP, TN, FP, dan FN.



Gambar 4.6 Hasil perhitungan *confusion matrix*

Dengan bantuan *library* Python didapatkan *accuracy* dengan menggunakan rumus diatas untuk masing-masing kelas ialah: kelas *negative* sebesar 79%, kelas *neutral* 89%, dan kelas *positive* 75% sesuai gambar 4.6 diatas.

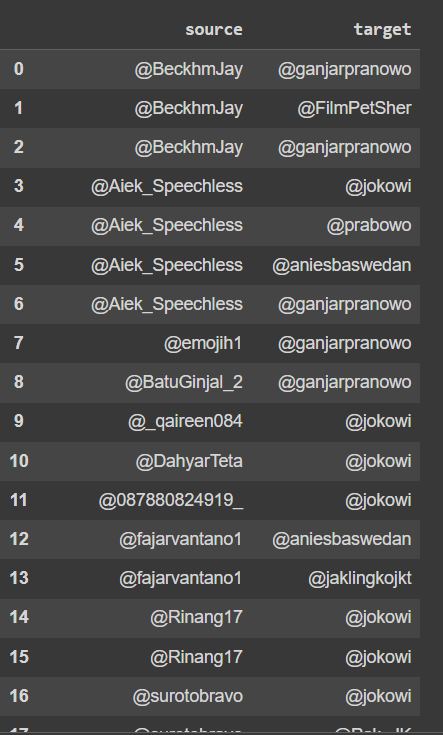
Model tersebut selanjutnya disimpan dalam format *.h5* untuk selanjutnya digunakan pada prediksi sentimen pada *form* LSTM *Predictor. User* dapat mengetikkan kalimat untuk mencoba melihat prediksi dari model LSTM yang telah di *training* sebelumnya seperti yang ditunjukkan pada gambar 4.7 dibawah.



Gambar 4.7 Prediksi sentimen dari model LSTM

**4.1.2 *Network Analysis***

Pada *network analysis* memiliki tahap awal yang sama dengan sentimen *analysis* yaitu pengambilan data *tweet* dengan *tools*. Data *tweet* yang digunakan memiliki kolom *handle* yang berisikan *username* akun dan kolom *responding* yang berisikan *tweet* dari akun yang ada pada kolom *handle*. Nama kolom *handle* akan diubah menjadi kolom *source* dan kolom *responding* akan diubah menjadi kolom *target* untuk memudahkan saat pemrograman. Kedua kolom tersebut yang akan digunakan sebagai *nodes* yang dibutuhkan untuk membuat *network analysis*. Kolom *responding* kemudian akan dibersihkan dengan menyisakan nama akun yang di *mention/*di *reply*. Hubungan antara *source* dan *target* akan dijadikan sebagai *edges*.

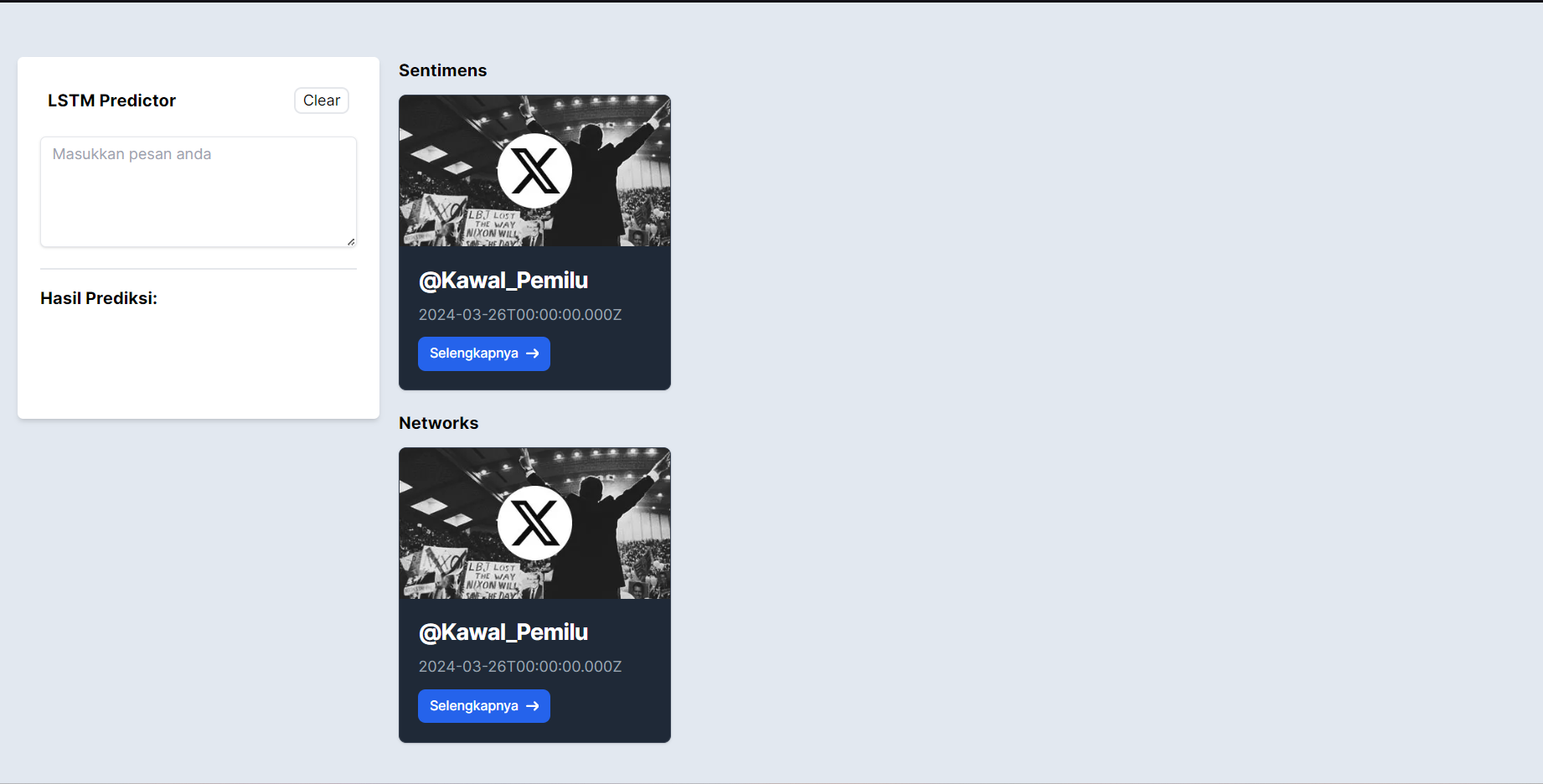


Gambar 4.8 Hasil *cleaning* pada tahap n*etwork analysis*

Pada tahap pembuatan visualisasi *network analysis* menggunakan bantuan *library* Python NetworkX. Adapun untuk visualisasi grafik *network analysis* menggunakan *degree centrality* dengan *spring layout*.

**4.1.3. Tampilan *Website***

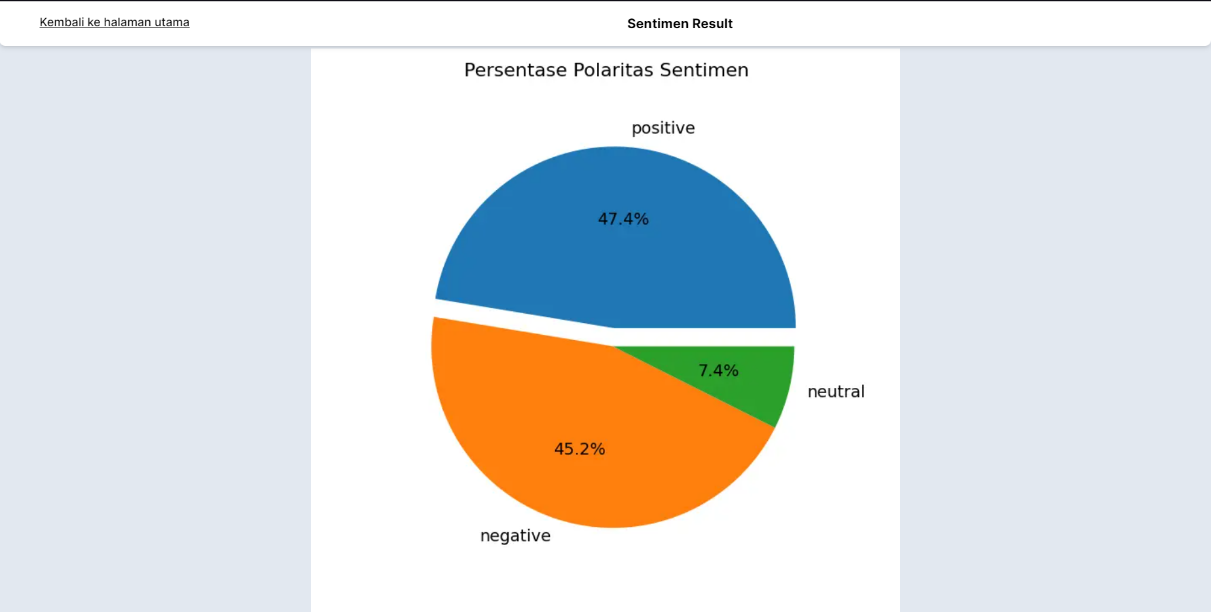
1). Tampilan *home*

****

Gambar 4.9 Tampilan halaman *home*

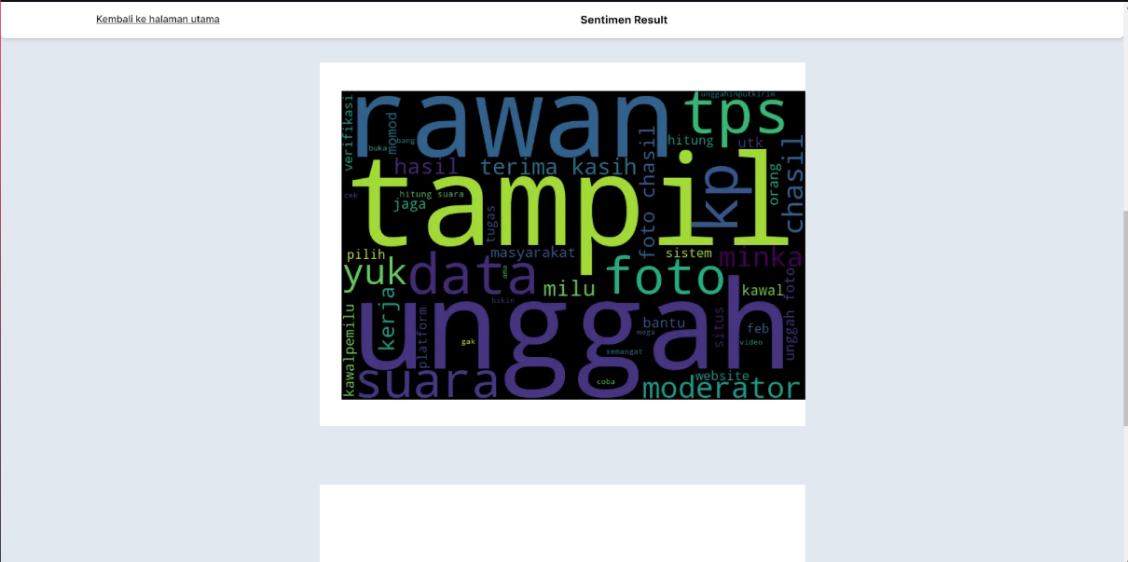
Gambar 4.9 merupakan tampilan halaman *home* yang memiliki 3 fitur yaitu LSTM *Predictor* yang berfungsi untuk mengetes model yang telah ditraining sebelumnya. *Cards* pada menu *Sentimens* akan muncul dengan menampilkan *thumbnail*, judul dan tanggal ketika telah diisi oleh admin pada halaman *dashboard*, begitu juga dengan menu *Networks*, masing-masing *cards* memiliki tombol ‘*selengkapnya’* yang dapat beralih ke halaman Sentimen *Result* maupun *Network Result*.

2). Tampilan halaman sentimen *result* dan *network result*



Gambar 4.10 Tampilan persentase polaritas sentimen

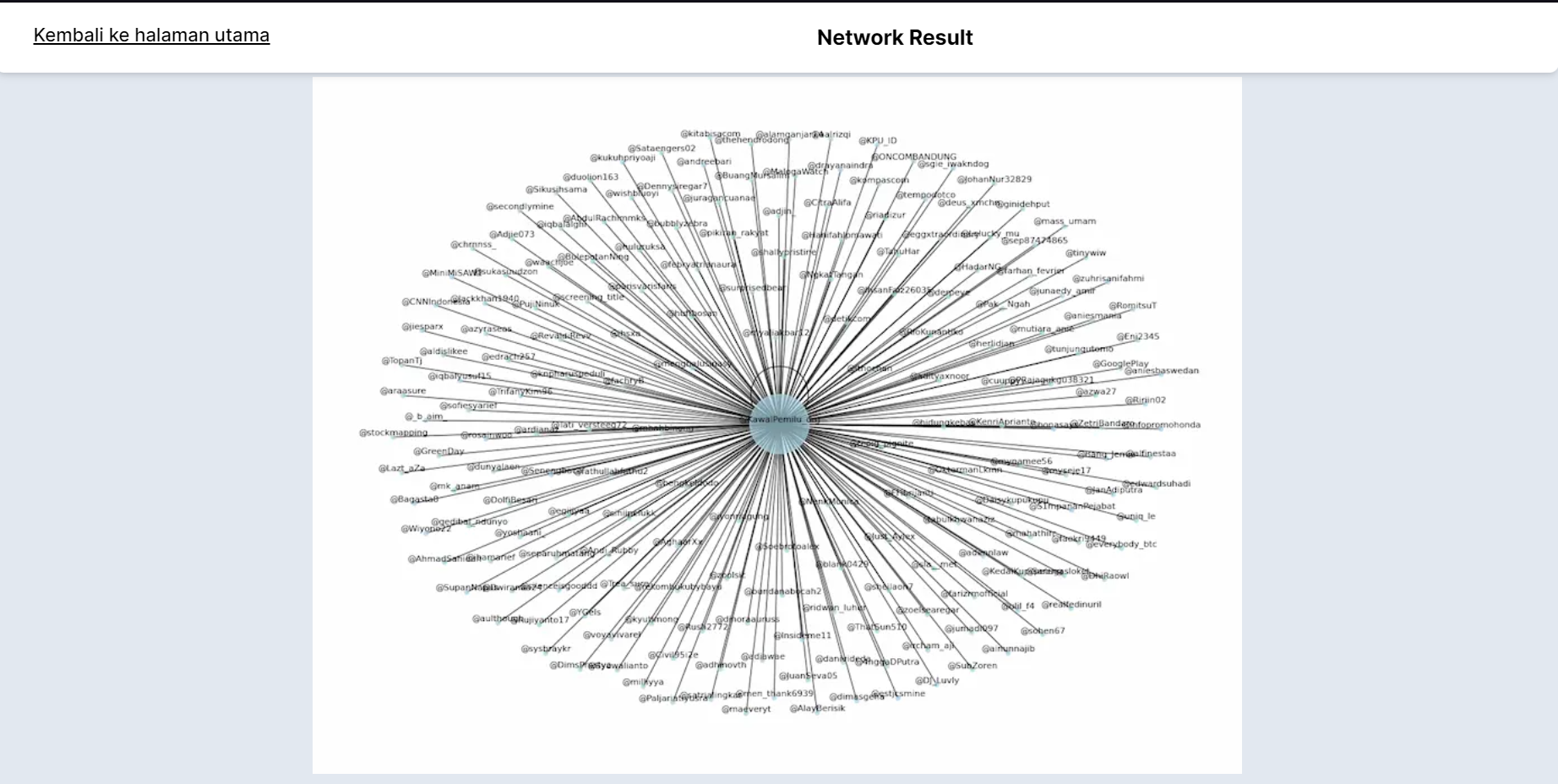
Jika tombol ‘*selengkapnya’* pada menu *Sentimens* ditekan maka *web* akan beralih ke halaman hasil pengolahan sentimen yang berisikan gambar 4.10 – gambar 4.12. Gambar 4.10 merupakan gambar hasil dari perhitungan persentase polaritas yang ditampilkan dalam bentuk diagram lingkaran. Gambar 4.11 merupakan hasil dari modus kata atau *keyword* dalam data *tweet* yang telah diproses sebelumnya. Gambar 4.12 merupakan hasil dari modus kata atau *keyword* dalam lingkup polaritas positif atau negatif dalam data *tweet* yang telah diproses sebelumnya. Selain itu pada halaman ini memiliki tombol ‘*kembali ke halaman utama*’ untuk kembali ke halaman *home*.



Gambar 4.11 Tampilan hasil *wordcloud* *keyword* teratas



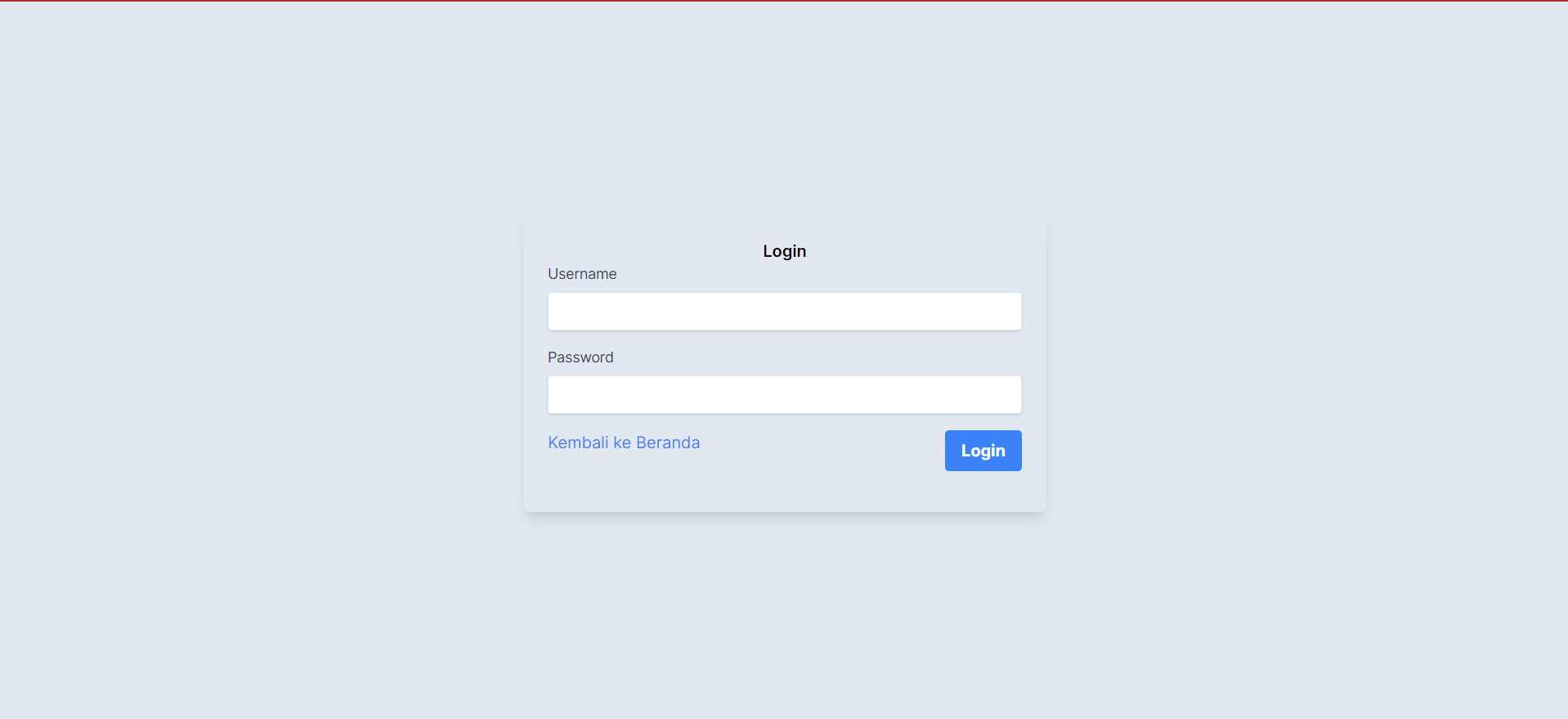
Gambar 4.12 Tampilan *wordcloud* positif dan negatif



Gambar 4.13 Tampilan hasil *network analysis*

Untuk *network result* sendiri akan tampil hasil seperti gambar 4.13 yang merupakan tampilan *network analysis* dengan *degree centrality* dan *layout spring* jika tombol ‘*selanjutnya’* pada *card* *networks* ditekan.

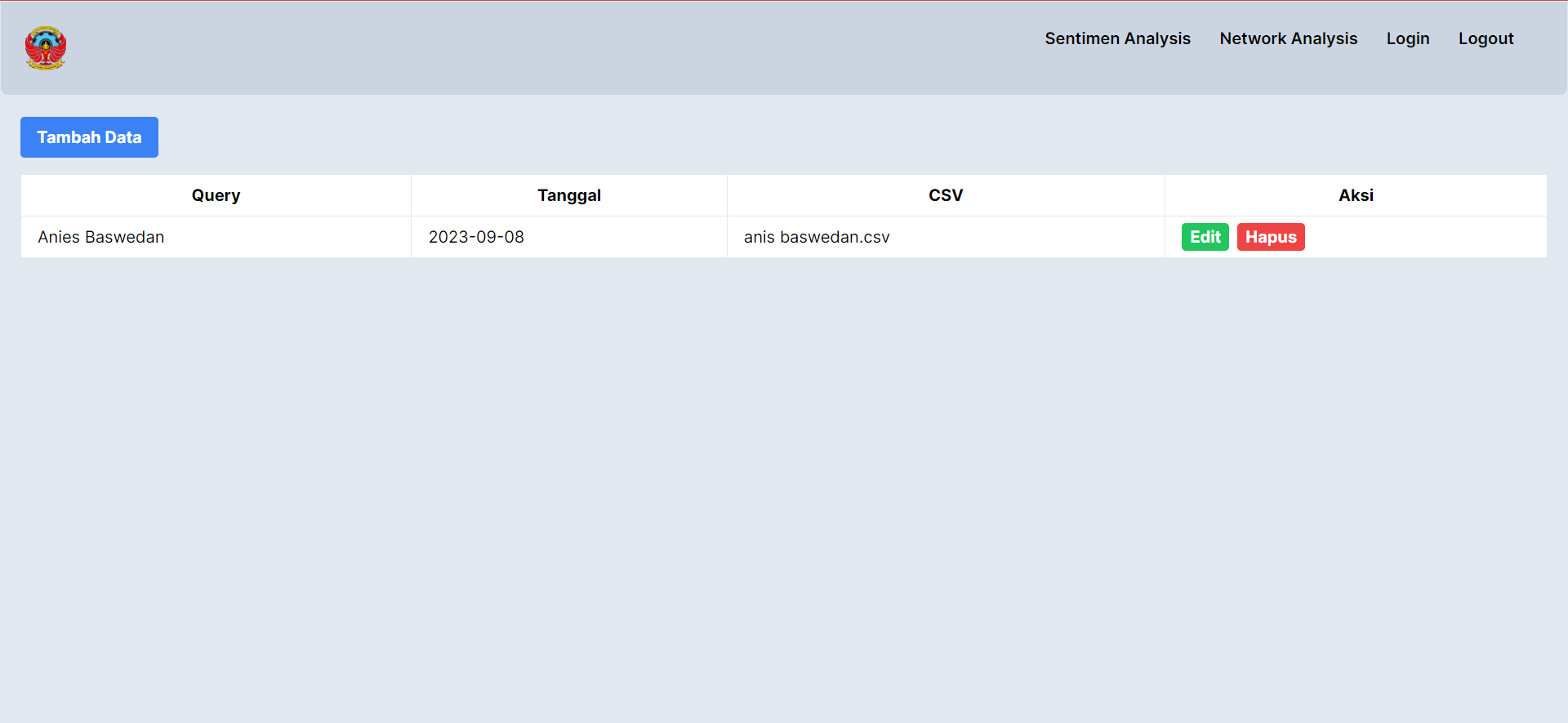
3). Tampilan halaman *login*



Gambar 4.14 Tampilan halaman *login*

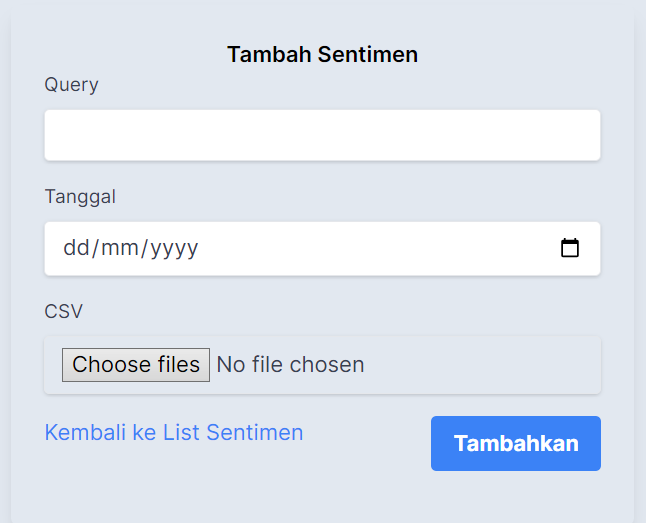
Gambar 4.14 merupakan tampilan halaman *login* yang hanya dapat diakses dengan *url* [*http://localhost:3000/login*](http://localhost:3000/login) dan ditujukan hanya untuk admin dikarenakan hanya admin yang dapat melakukan *input* data yang diperlukan dan nantinya dapat dilihat oleh *user*.

4). Tampilan *dashboard* admin



Gambar 4.15 Tampilan halaman admin

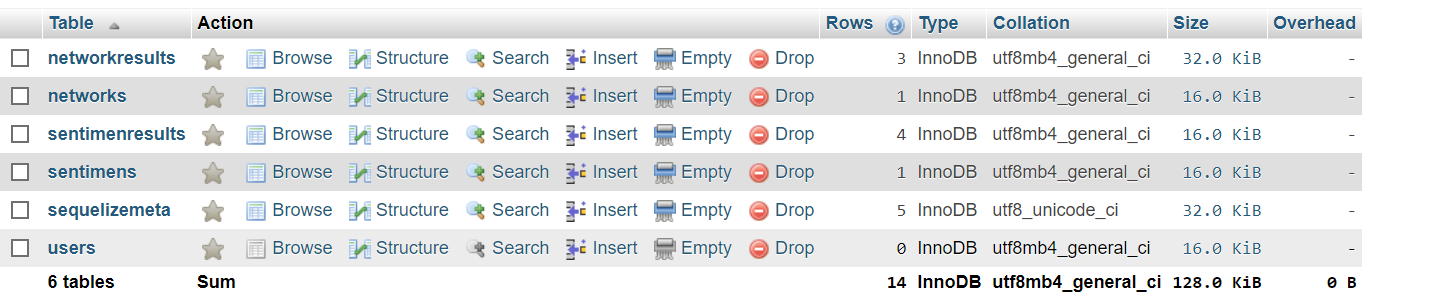
Pada gambar 4.15 merupakan tampilan setelah proses *login* selesai dan memasuki halaman *dashboard*. Pada halaman ini memiliki fitur tambah data, dan hapus data yang masing-masing dapat digunakan untuk mengatasi data baik *sentimental analysis* maupun *network analysis* pada masing masing halamannya.



Gambar 4.16 Tampilan tambah sentimen

Gambar 4.16 merupakan tampilan *form* data yang dapat diisi oleh admin. *Query* dapat diisi sesuai dengan konteks atau perihal *sentiment* yang akan diproses. Pada *form* ‘*Tanggal*’ berfungsi untuk menginformasikan tanggal saat data sentimen tersebut diambil dari Twitter. *Form* ‘*CSV*’ dapat diisikan *file* *csv* yang telah diambil dari Twitter. Tombol ‘*Tambahkan’* berfungsi untuk melakukan proses mulai dari *cleaning data* hingga menampilkan hasilnya pada halaman *sentiment result.* Pada bagian *form Network Analysis* juga sama seperti diatas

5). Tampilan *database*



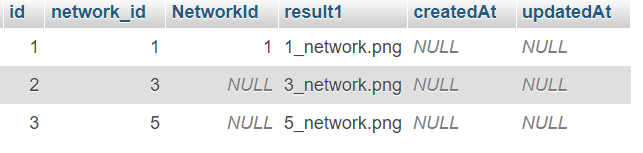
Gambar 4.17 Tampilan *database*

Pada gambar 4.17, hasil gambar akan masuk ke dalam masing-masing *sentimenresult* atau *networkresults*, *sequelizemeta* sendiri bertujuan untuk menghubungkan *database* pada NodeJs, untuk tabel *users* dipergunakan untuk menyimpan data admin seperti *username* dan *password*.



Gambar 4.18 *database* *sentimenresult*

Tabel *sentimenresult* akan menyimpan *id*, *file csv*, *file* gambar persentase, *wordcloud* keseluruhan, *wordcloud* negatif dan positif. Data tersebut yang nantinya akan dipakai untuk memanggil *file* untuk ditampilkan pada halaman sentimen *result*



Gambar 4.19 *database* *networkresuilt*

Tabel *network* sendiri akan menyimpan *id* dan hasil gambar *network* *analysis*. Setelah itu, data tersebut akan terpanggil jika *user* menekan tombol ‘*selengkapnya’* untuk melihat hasil masing-masing *sentimental* *analysis* atau *network* *analysis*

**4.2 Metode Validasi**

Pada tahap validasi, peneliti membandingkan hasil prediksi sentimen menggunakan model yang telah ditraining sebelumnya dengan hasil dari *lexicon*. Sedangkan untuk *network* *analysis* peneliti membandingan hasil visualisasi dari penelitian ini yang menggunakan *library* networkx dengan hasil visualisasi dengan bantuan *software* gephi.

**4.2.1 Metode Validasi *Sentimental* *Analysis***

Peneliti melakukan validasi data hasil *sentiment analysis* yang dilakukan dengan menggunakan model LSTM yang telah di-*training* dan membandingkan dengan hasil sentimen yang menggunakan *lexicon* InSet. Sampel yang diujikan diambil dari Twitter dengan jumlah 8 sampel. Untuk mendapatkan hasil sentimen model, peneliti menerapkan model tersebut pada *website* yang telah dibuat pada penelitian ini. Adapun tabel untuk validasi sampel yang akan dilakukan perbandingan sentimen ialah sebagai berikut:

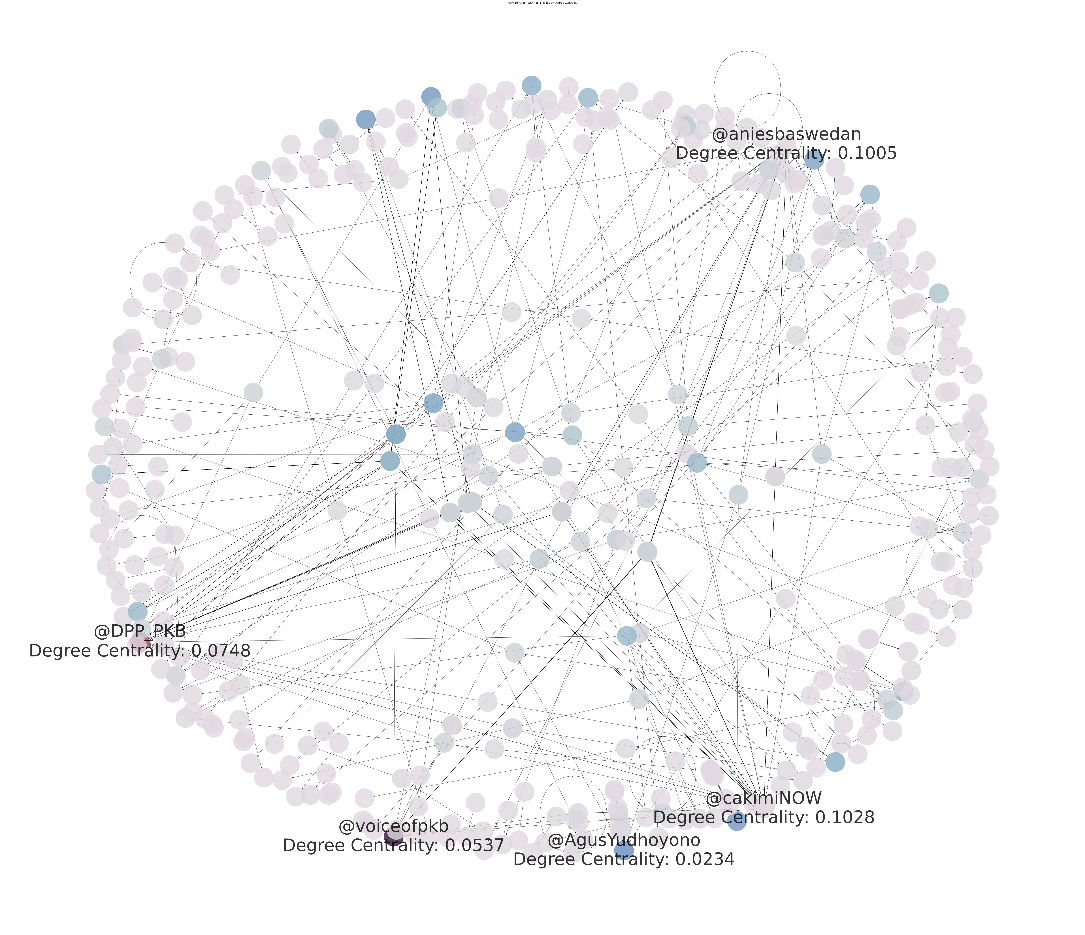
Tabel 4.2 Sampel validasi hasil sentimen

| No | *Tweet* | Sentimen dari l*exicon* InSet | SentimendariModelLSTM |
| --- | --- | --- | --- |
| 1 | Mahasiswa mengapresiasi kelancaran pelaksanaan pemilu 2024. | Positif | Positif |
| 2 | Kalau ada parpol yang pemilu janji "kami berada di pihak rakyat" tau tau ngomong gini, mending bubar aja. Munafik banget. Bangga banget hidup dari duit pajak. | Negatif | Positif |
| 3 | KITA BANTAI DI PEMILU 2029 tapi akankah pemilu tetap hadir di tahun 2029 (the concern is concerning) | Negatif | Netral |
| 4 | Justru kami merasa terhina dan dibodohi serta dicolok mata disiang hari saat ada orang yg mengatakan Pemilu berjalan wajar dan normal saat seorang Presiden terang2an mengaku cawe2 dan memang sangat terasa dan terlihat. Benar2 menghina akal sehat. semoga Allah menangkan kebenaran. | Negatif | Positif |
| 5 | AMIN (Anies- Muhaimin) Capres/Cawapres Alhamamdulillah..."AMIN"... kan untuk Indonesia yang lebih baik @aniesbaswedan  @cakiminNOW | Netral | Netral |
| 6 | Gamau tau pokonya yang pake politik identitas itu cuma Anies 😡😡😡 | Negatif | Negatif |
| 7 | Semoga menjadi partai yang selalu memiliki idealisme kerakyatan, dan Bisa membawa perubahan pada kesehatan iklim politik bangsa yg Hari ini byk masuk angin akut.... Dan Bisa Jadi partai pengusung Pak Anies di Pilkada DKI .... 🤔 Semangat Perubahan Gak pernah padam !! Tidur....... | Positif | Positif |
| 8 | Yg tidak setuju Makan Gratis, dianggap tidak waras. Guru Besar dan akademisi mengkritik, dituduh punya kepentingan politik. Himbauan politik lewat film "Dirty Vote", dibilang fitnah. Menanyakan penggunaan anggaran negara, diancam dilaporkan.  Kebayang gak jika mereka berkuasa? | Negatif | Negatif |

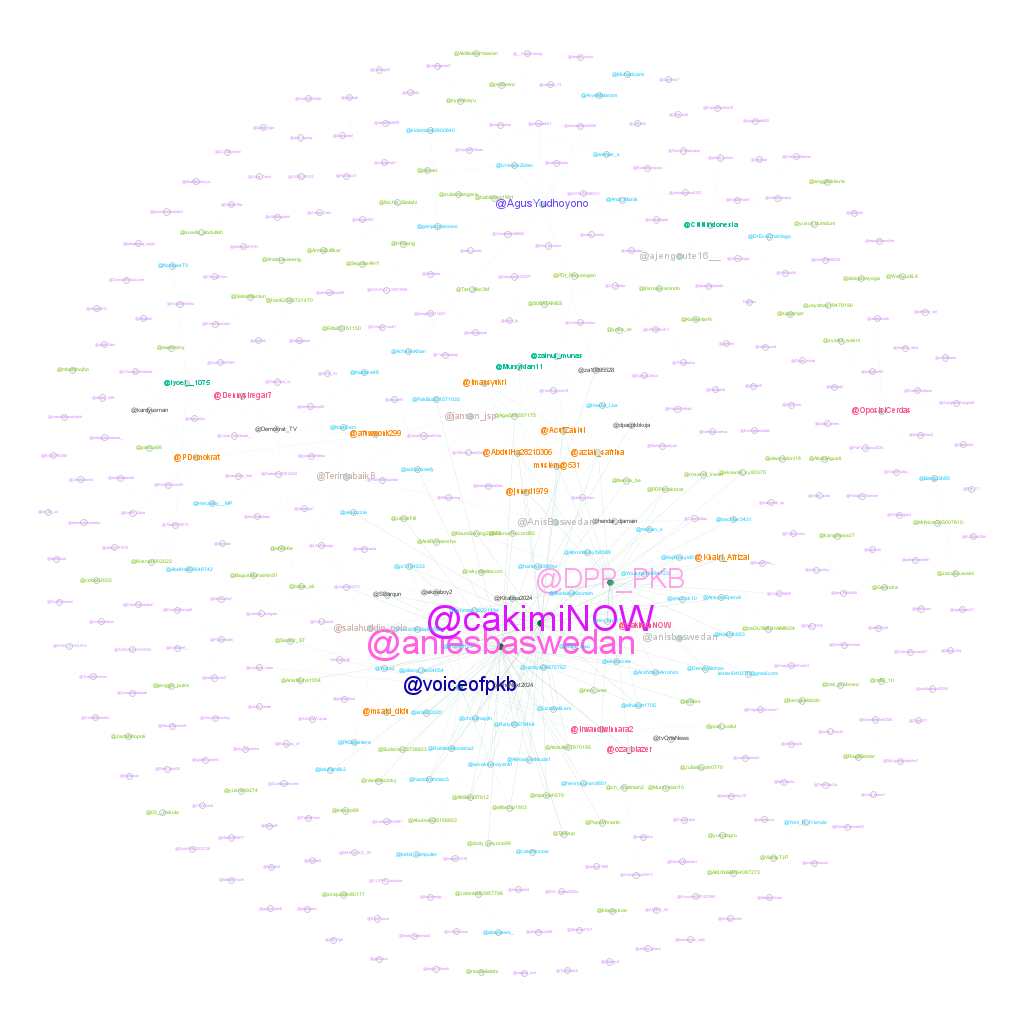
Tabel 4.2 merupakan tabel yang menampilkan hasil dari 2 metode *sentiment* *analysis*, yaitu metode dengan *lexicon* dan metode dengan model LSTM. Berdasarkan dari tabel diatas dapat disimpulkan bahwa dari 8 sampel hasil sentimen dari *lexicon* InSet dan sampel hasil sentimen dari model LSTM, terdapat 3 sampel yang memiliki perbedaan, yaitu pada teks nomor 2 dan 4 yang dimana hasil yang menggunakan *lexicon* menunjukkan sentimen negatif sedangkan model LSTM menunjukkan sentimen positif, dan teks nomor 3 dimana hasil yang menggunakan *lexicon* menunjukkan sentimen negatif sedangkan model LSTM menunjukkan hasil positif. Untuk sisanya 5 sampel memiliki hasil yang sama.

**4.2.2** **Metode Validasi *Network* *Analysis***

Peneliti melakukan metode validasi *network* *analysis* dengan membandingkan visualisasi antara visualisasi yang dihasilkan dari *library* Networkx dengan aplikasi gephi. Data yang diproses sebanyak 500 data *tweet*.



Gambar 4.20 Visualisasi dengan NetworkX



Gambar 4.21 Visualisasi dengan Gephi

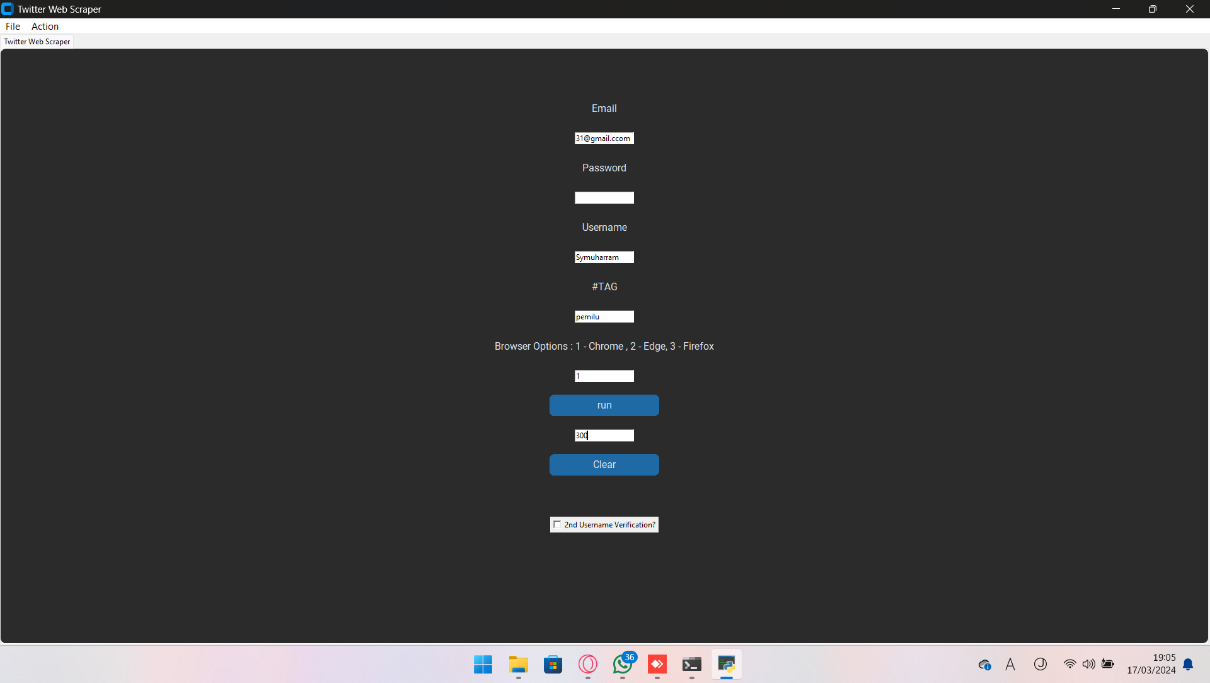
Terlihat persamaan antara gambar 4.20 dengan gambar 4.21 bahwa 5 akun teratas berdasarkan pendekatan *degree centrality*, yaitu @cakiminNow, @aniesbaswedan, @DPP\_PKB, @voiceofpkb, dan @AgusYudhoyono. Adapun perbedaan dalam pembuatan visualisasi *network* *analysis* tersebut terletak pada bagian *layout*-nya yang dimana pada *software* Gephi lebih menarik secara visual dikarenakan akses pengaturan terhadap pembuatan visualisasi lebih lengkap dibanding *library* NetworkX yang lebih sederhana dan kurang menarik.

**4.3 Metode Pengujian *Website***

Untuk pengujian *Website* terdiri atas 2 pengujian. Pengujian pertama dilakukan untuk menguji *website* terhadap data yang diambil dari Twitter dengan kondisi data diambil dari akun @KawalPemilu\_org dengan menggunakan *keyword*. Pengujian kedua dilakukan dengan metode *blackbox* untuk mencoba kinerja pada sistem *website* dengan hasil seperti pada tabel 4.3.

**4.3.1 Pengujian pada akun @KawalPemilu\_org**

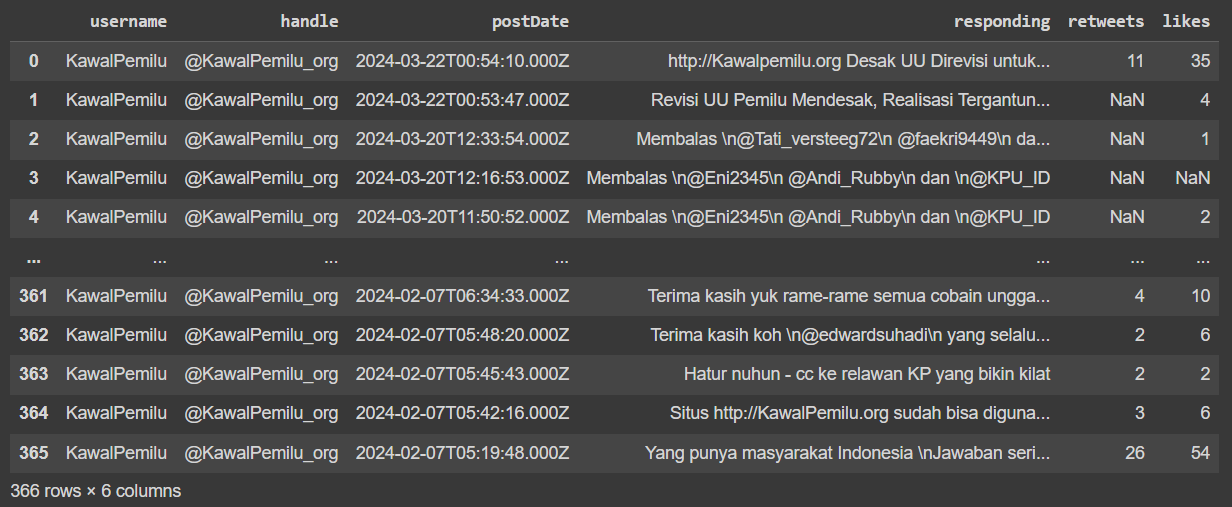
Untuk pengujian pada data akun tertentu yang relevan terhadap isu politik ini menggunakan *tools* Twitterscraper untuk mengambil data *tweet*.



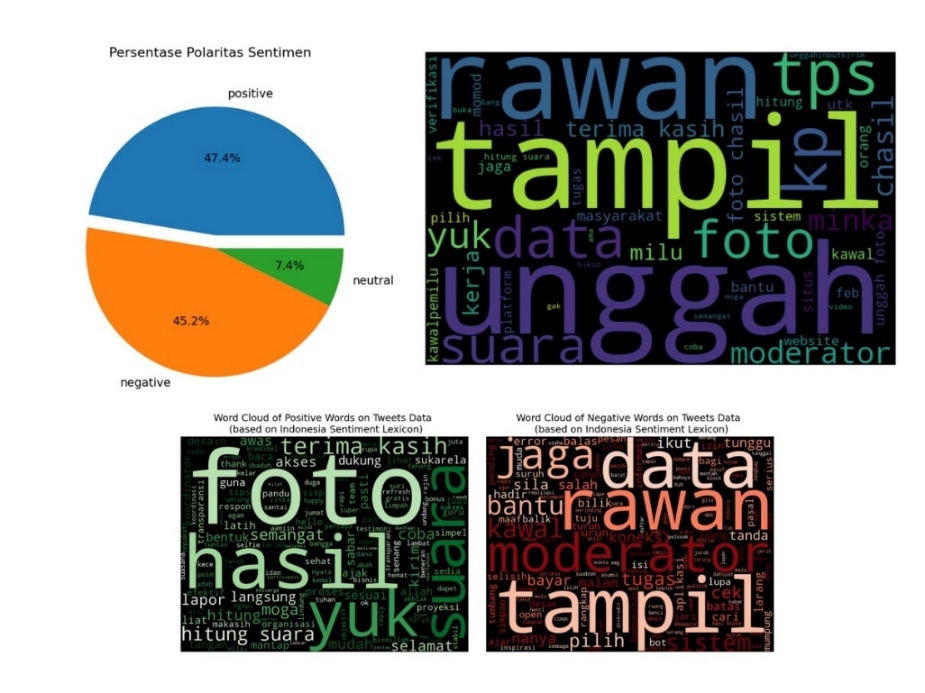
Gambar 4.22 Pengambilan data dengan *tool* TwitterScraper

*Tools* yang digunakan memiliki 6 *form* yaitu *form* *Email*, *Password*, dan *Username* berfungsi untuk melakukan otomatisasi dalam pengaksesan akun Twitter, *form* *#TAG* berfungsi untuk memasukkan *keyword* yang ingin di-*scraping*. Di bagian ini dapat memanfaatkan fitur *advanced search* yang disediakan oleh twitter untuk mencari *keyword* yang lebih spesifik, misalnya kata *lang:id* yang berguna untuk mencari *tweet* yang hanya berbahasa Indonesia, *form browser* *option* berfungsi untuk memilih *engine* *browser* yang akan digunakan dalam pencarian *tweet*, dan *form* *textbox* terakhir berfungsi untuk menentukan angka maksimal data yang di-*scraping*. Adapun untuk *checkbox* dibawah digunakan untuk melewati *second* *login* *authority* dari Twitter.

Pada pengambilan data *tweet*, *keyword* yang dimasukkan yaitu, “*pemilu (from:KawalPemilu\_org) lang:id”.* *Keyword* tersebut berfungsi untuk mencari *tweet* yang mengandung kata pemilu, mengeksklusifkan pencarian pada akun @KawalPemilu\_org, dan hanya mengambil *tweet* yang berbahasa Indonesia. *Output* dari *tools* tersebut memiliki *format* .csv yang dimana memiliki kolom *username* yang didapatkan dari *username* akun Twitter yang bersangkutan, *handle* yang didapatkan dari *Twitter handle*, *postdate* yang didapatkan dari tanggal *tweet* tersebut di-*post*, *responding* yang merupakan isi dari *tweet* tersebut, *retweet* dan *likes* yang berisikan jumlah akun yang melakukan *retweet* dan *likes*.

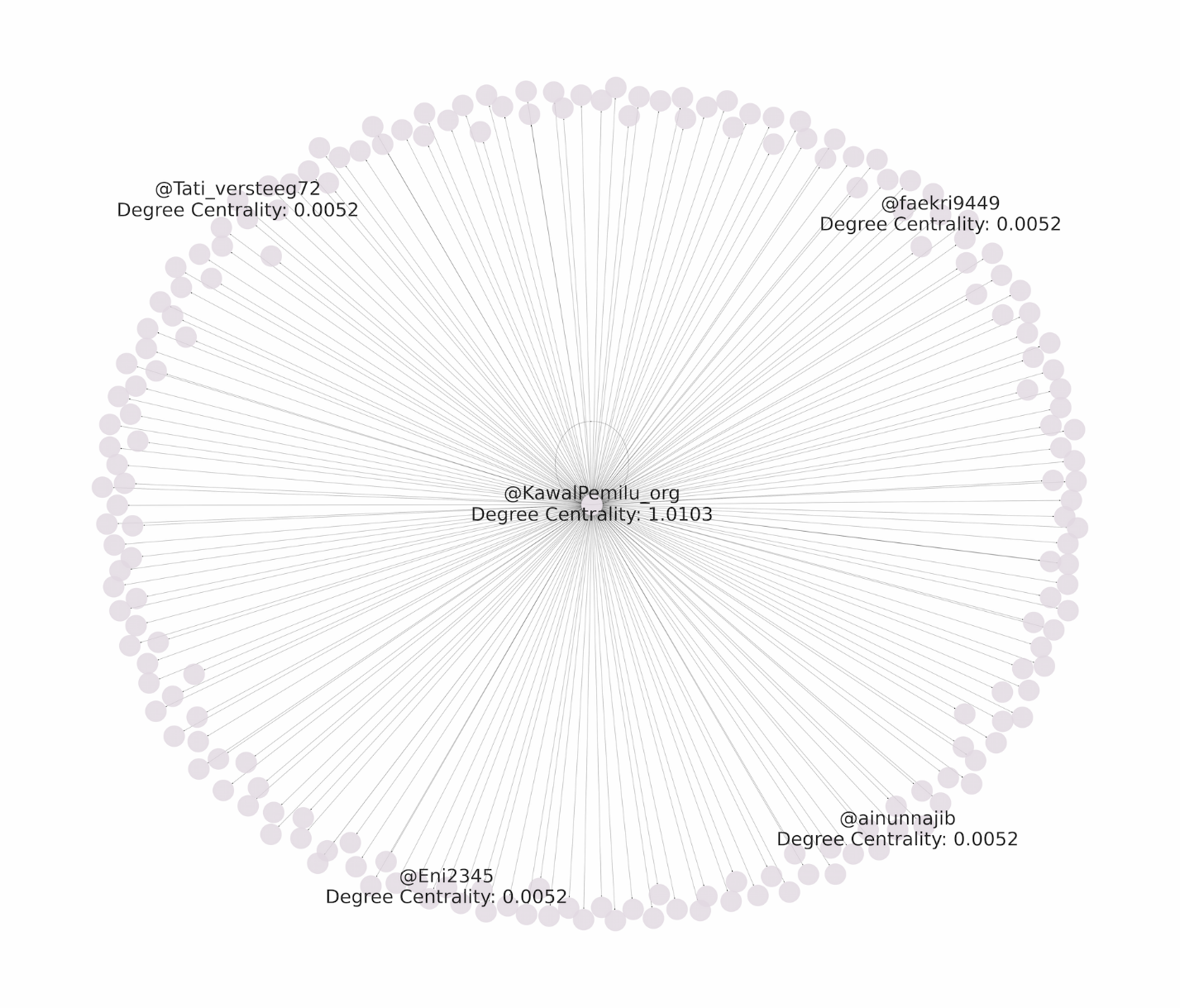


Gambar 4.23 Tampilan *dataframe*



Gambar 4.24 Hasil setelah *upload* data *file* *csv*

Data *tweet* tersebut nantinya akan di-*upload* melalui *website* dan setelah proses *sentiment* *analysis* selesai akan menghasilkan *output* seperti gambar 4.24 yang dimana pada gambar persentase terlihat bahwa pada data *tweet* yang digunakan memiliki persentase positif sebanyak 47,4%, persentase netral sebanyak 7,4%, dan persentase negatif sebanyak 45,2%. Gambar pada bagian kanan menampilkan kata yang paling banyak digunakan secara keseluruhan, pada data *tweet* yang digunakan terlihat kata tampil, unggah, dan rawan merupakan kata yang memiliki ukuran yang lebih besar daripada kata yang lainnya dan dapat disimpulkan bahwa ketiga kata tersebut yang paling banyak digunakan pada data *tweet* tersebut. Gambar paling bawah menampilkan kata yang paling banyak digunakan dalam lingkup polaritas positif dan negatif. Pada kelompok polaritas positif, kata foto, hasil, dan suara merupakan kata yang paling banyak digunakan. Sedangkan pada kelompok polaritas negatif, kata yang paling sering digunakan yaitu kata tampil, moderator, rawan, dan data.



Gambar 4.25 Hasil *network analysis* setelah *upload* data *file* csv

Gambar 4.25 merupakan hasil *network analysis* yang didapatkan dari data *tweets* dengan *keyword* *from*:@KawalPemilu\_org lang:id yang berjumlah 366 data *tweets*. Data *tweets* yang digunakan merupakan *tweet* yang diposting oleh akun @KawalPemilu\_org. Akun atau *node* pada gambar tersebut memiliki banyak hubungan interaksi terhadap akun-akun lainnya. Pada gambar tersebut ada 4 akun yang memiliki interaksi yang berpengaruh pada akun atau *node* @KawalPemilu\_org. Salah satu interaksi pada akun @KawalPemilu\_org, yaitu memposting *tweet* dengan melakukan *mention* terhadap akun populer seperti @ainunnadjib. Setelah ditelusuri, *tweet* tersebut cenderung berisikan komentar tentang ajakan memilih dalam pemilu 2024 seperti pada gambar 4.26 dibawah.



Gambar 4.26 *Tweet* dari akun @KawalPemilu\_org

**4.3.2 Pengujian *website***

Pengujian pada penelitian ini menggunakan pengujian *blackbox* yang merupakan metode pengujian untuk menguji sistem bekerja sesuai fungsionalitas yang diinginkan. Adapun hasil pengujian *blackbox* dapat dilihat pada tabel 4.3 sebagai berikut:

Tabel 4.3 Pengujian *Blackbox*

| No | Kelas uji | Skenario uji | Hasil yang diharapkan | Keterangan |
| --- | --- | --- | --- | --- |
| 1 | Tombol selengkapnya pada *card* s*entiments* | Menekan tombol selengkapnya | Menampilkan halaman hasil s*entiment analysis* | Berhasil |
| 2 | Tombol selengkapnya pada *card* *networks* | Menekan tombol selengkapnya | Menampilkan halaman hasil *network* *analysis* | Berhasil |
| 3 | Pengetikan kalimat pada form lstm *predictor* | Mengetik kalimat yang ingin diprediksi | Hasil dari prediksi sentiment tampil | Berhasil |
| 4 | Tombol *login* | Menekan tombol *login* | Menampilkan halaman *dashboard* admin | Berhasil |
| 5 | Tombol *logout* | Menekan tombol *logout* | Keluar dari halaman *dashboard* dan kembali ke halaman *home* | Berhasil |
| 6 | Tombol tambah data untuk *sentiment analysis* | Menekan tombol tambah data | Menampilkan tambahan data pada halaman *admin*, *home*, dan *database* mysql | Berhasil |
| 7 | Tombol tambah data untuk *network analysis* | Menekan tombol tambah data | Menampilkan tambahan data pada halaman *admin*, *home*, dan *database* mysql | Berhasil |
| 8 | Tombol hapus untuk *sentiment analysis* | Menekan tombol hapus | Menghapus data *sentiment analysis* yang telah di-*upload* | Berhasil |
| 9 | Tombol hapus untuk *network analysis* | Menekan tombol hapus | Menghapus data *network analysis* yang telah di-*upload* | Berhasil |

Berdasarkan pengujian *blackbox* yang dilakukan pada penelitian ini, dapat disimpulkan bahwa fungsionalitas dari setiap fitur yang ada bekerja sesuai dengan target yang diharapkan. Untuk penjelasan mendetail, fitur tombol ‘*selengkapnya’* pada kedua *card* pada halaman *home* dapat menampilkan hasil gambar sesuai proses masing-masing. Selanjutnya untuk fitur pengetikan prediksi sentimen juga berjalan baik sesuai dengan yang diharapkan walaupun *respond time* yang kian bertambah seiring pengetikan kalimat yang memiliki jumlah kata yang banyak. Pada tombol *login* dan *logout* berjalan dengan baik. Pada kedua tombol tambah data baik untuk *sentiment analysis* dan *network analysis* juga berjalan sesuai fungsionalitasnya, dengan catatan pada *sentiment analysis* memiliki *respond time* yang memakan waktu cukup lama sekitar 2-3 menit untuk data yang berjumlah kurang lebih 300-400 data *tweet* yang diakibatkan dari proses *cleaning data*.

# **BAB V KESIMPULAN DAN SARAN**

## Kesimpulan

Hasil pada penelitian ini adalah *website sentiment analysis* yang menggunakan arsitektur LSTM pada saat training mengalami kondisi *overfitting*, maka dari itu dilakukan K-Fold *cross validation* untuk mengurangi *overfitting*, sehingga didapatkan akurasi training dengan K-Fold *cross validation* dengan parameter *n\_splits=5, shuffle=True,* dan *random\_state=42* sebesar 71%. Untuk akurasi *testing* sebesar 74%, selanjutnya digunakan *confusion matrix* untuk mencari akurasi masing masing kelas dan didapatkan akurasi masing-masing yaitu: kelas *negative* sebesar 79%, kelas *neutral* 89%, dan kelas *positive* 75%.

Pada *network analysis* yang menggunakan *degree centrality* dengan *spring layout* dan diterapkan pada data Twitter yang berfungsi baik untuk mengetahui sentimen pada data *keyword* yang diambil dari Twitter maupun untuk mengetahui hubungan relasi antar pengguna Twitter yang saling *reply* dan/atau *mention*.

Berdasarkan pengujian *blackbox* yang diterapkan pada *website* didapatkan hasil berupa semua fungsionalitas fitur dan tombol yang berjumlah 7 pengujian pada *website* berjalan sesuai dengan fungsinya. Untuk penerapan *sentimental* *analysis* yang memiliki data *tweet* terhadap akun @kawal\_pemilu\_org dengan jumlah data sebanyak 366 data memiliki respond time yang cukup tinggi yaitu selama 2-3 menit yang diakibatkan dari proses cleaning data.

## Saran

Berdasarkan hasil penelitian yang dilakukan peneliti dapat memberikan saran yang dapat diterapkan di kemudian hari agar penelitian ini dapat dikembangkan lagi. Adapun saran yang peneliti sampaikan ialah sebagai berikut:

1. Menggunakan algoritma *machine learning* yang lain dan membandingkannya dengan hasil yang didapatkan.
2. Menambahkan fitur pada website yang telah ada.

# DAFTAR PUSTAKA

Alamsyah, Andri dan Dian Puteri Ramadhani. 2020 *Pengenalan Social Network Analysis : Konsep dan Praktis*. Cetakan ke-1 Bandung: CV.Sadari

Allen, James. 1995. *Natural Language Understanding*. ed. California: Benjamin-Cummings.

Anggana, Adi Surya dkk,. 2020. Pengujian *Black* *Box* pada Aplikasi Admin PT. World Trans Berbasis *Web* Menggunakan Metode *Equivalence* *Partitioning. Jurnal Informatika Universitas Pamulang, (Online),* 5(1), 95-99, (https://openjournal.unpam.ac.id/ ), diakses 15 Mei 2023.

Aslam, Fankar Armash, dkk. 2015. *Efficient Way Of Web Development Using Python And Flask.. International Journal of Advanced Research in Computer Science*. (*Online*), 6(2): 54–57, (<https://ijarcs.info/index.php/Ijarcs/article/view/2434>), diakses 4 April 2023.

Bhumika, S. Sehra, and A. Nayyar*.* 2013*. A Review Paper on Algorithms Used for Text Classification, International Journal of Application or Innovation in Engineering & Management (IJAIEM), (Online)*, 2(3):90–99*,* ([*www.ijaiem.org/Volume2Issue3/IJAIEM-2013-03-13-025.pdf*](http://www.ijaiem.org/Volume2Issue3/IJAIEM-2013-03-13-025.pdf)*),* diakses 4 April 2023.

Fauziah, Nisa Nur. 2023. *Perancangan Webcomic Tindakan Cyberbullying Sebagai Media Edukasi Untuk Remaja*. Skripsi. Bandung. Universitas Pendidikan Indonesia.

Feldman, Ronan and James Sanger. 2007*. The Text Mining Handbook,* ed.Cambridge: Cambridge University Press.

Handayanto, Rahmadya Trias dkk. 2021. *Analisis Sentimen Pada Situs Google Review dengan Naïve Bayes dan Support Vector Machine*. *Jurnal Komtika (Komputasi dan Informatika), (Online),* 5 (2):153-163, (<https://journal.unimma.ac.id/index.php/komtika/article/view/62 80/2874>), diakses 27 Maret 2023.

Hidayatullah, Priyanto., Jauhari Khairul Kawistara. 2014. *Pemrograman Web.* Cetakan ke-2 Bandung : Informatika Bandung.

Inayah, Dwi dan Fredy Law Purba. 2020 *Implementasi Social Network Analysis Dalam Penyebaran Informasi Virus Corona (Covid-19) Di Twitter*. Prosiding *Seminar Nasional Official Statistics.* 292-299. Politeknik Statistika STIS.

Indrawan, Gede. 2021. *Database MySQL dengan Pemograman PHP.* Cetakan Ke-1. Depok:Rajawali Pers.

Izzulsyah, Imam dkk. 2022. *Analisis Penggunaan Media Sosial Di Masa Pandemi (Analysis Of Social Media Use During Pandemic)*. *Jurnal Fraction, (Online),* 2 (1):21-31, (<https://fraction.ubb.ac.id/index.php/fraction/article/view/28/11>). Diakses 27 Maret 2023.

Kameswari, Kambhampati Kalyana et al. 2019. *Predicting Election Results using NLTK. International Journal of Innovative Technology and Exploring Engineering. 9(1)*, 4519-4529, (<https://www.ijitee.org>), diakses 5 April 2023.

Koto, Fajri dan Gemala Y. Rahmaningtyas. 2017. *InSet Lexicon: Evaluation of a Word List for Indonesian Sentiment Analysis in Microblogs. 2017 International Conference on Asian Language Processing (IALP)*, Singapore

Li Chen dan Liu Weiguo. 2019. *Chinese Text Information Extraction Based on NLTK. Computer Systems & Applications (28)1.* 275−278

Mubariz, Ahsan dkk. 2020. Perancangan Back-End Server Menggunakan Arsitektur Rest dan Platform Node.JS (Studi Kasus: Sistem Pendaftaran Ujian Masuk Politeknik Negeri Ujung Pandang). Prosiding *Seminar Nasional Teknik Elektro dan Informatika (SNTEI).* 72-77. Makassar:Politeknik Negeri Ujung Pandang.

Mufidah, Karima dkk. 2020. *Analisis Aktor Popular Dan Sutradara Berpengaruh Berdasarkan Data Dbpedia Menggunakan Algoritma Closeness Centrality Dan Node2vec*.Majalah *Ilmiah UNIKOM* XVIII(1). 37-48

Nasir, Muhamad dan Novia Natasya. 2020*. Sistem Monitoring Akuarium Berbasis Mikrokontroler Dan Django Web Framework*. *Jurnal: Elektrika Borneo (JEB*, (*Online*). *6*(1), 25–28, (http://jurnal.borneo.ac.id), diakses 3 April 2023.

Ningrum, Ayu Ahadi, dkk. 2021. *Algoritma Deep Learning-Lstm Untuk Memprediksi Umur Transformator. (Online),* *Jurnal Teknologi Informasi dan Ilmu Komputer (JTIIK)*, 8(3), 539–548, (<https://jtiik.ub.ac.id/index.php/jtiik>), diakses 2 April 2023 .

Ningrum, Fadhila Cahya, dkk. 2019. Pengujian *Black Box* pada Aplikasi Sistem Seleksi *Sales* Terbaik Menggunakan Teknik *Equivalence* *Partitions.* *Jurnal Informasi Univiversitas Pamulang*, (*Online*), 4(4). 125–130, (<https://openjournal.unpam.ac.id>), diakses 3 Mei 2023.

Nugraha, Ivan Dwi dan Azhar Y. 2022. *Deteksi Depresi Pengguna Twitter Indonesia Menggunakan LSTM-RNN*. *Jurnal Nasional Pendidikan Teknik Informatika : JANAPATI*, 11(3), 320–329.

Putra, Jasen Aprian dan Akhmad Budi. 2020. *Penerapan Natural Language Processing Dalam Aplikasi Chatbot Sebagai Media Pencarian Informasi Dengan Menggunakan React (Studi Kasus: Institut Bisnis Dan Informatika Kwik Kian Gie)*. *Jurnal Informatika dan Bisnis*, (*Online*), 9(2).1-12, (<https://jurnal.kwikkiangie.ac.id>), diakses 1 April 2023.

Ramdhan, Nur Ariesanto dan Devi Adi Nufriana. 2019. Rancang Bangun Dan Implementasi Sistem Informasi Skripsi Online Berbasis Web. *Jurnal Ilmiah Intech : Information Technology Journal of UMUS, (Online),* 1(2), 1-12, (https://jurnal.umus.ac.id), diakses 20 April 2023.

Selisker, Scott. 2017*. Social networks. American Literature in Transition*. Cambridge: Cambridge University Press.

Singh, M. et al,. 2019. *Implementation of Database Using Python Flask Framework. International Journal Of Engineering And Computer Science, (Online),* 8(12)*,* 24894-24899, (<https://www.ijecs.in/index.php/ijecs>), diakses 28 April 2023.

Soemedhy, Chandra Ayunda Apta dkk,. 2022. *Analisis Komparasi Algoritma Machine Learning untuk Sentiment Analysis*. *Jurnal Informatika: Jurnal pengembangan IT (JPIT)*, (*Online*) 7(2).80-84, (https://ejournal.poltekharber.ac.id), diakses 2 April 2023 .

Soen, Surya Soenaryo dkk. 2017. *Pembuatan Aplikasi Salesforce Automation System Pada Toko IV Berbasis Android*. Dalam Infra, V(1).192-198.

Subagia, Anton. 2018*. Membangun Aplikasi Web dengan Metode OOP.* Cetakan ke-1.Jakarta: PT*.* Elex Media Komputindo.

Sudiartha, I Ketut Gede dkk. 2018. *Membangun Struktur Realtime Database Firebase Untuk Aplikasi Monitoring Pergerakan Group Wisatawan*. Dalam *Jurnal Ilmu Komputer*. XI(2). 96-102.

Syahrudin, Akbar Nur, dan Tedi Kurniawan. 2018. *Input* dan *Output* pada Bahasa Pemrograman Python. *Jurnal Dasar Pemrograman Python* STMIK, (Online), 1–7, ([www.academia.edu](http://www.academia.edu)), diakses 5 April 2023.

Tabassum, Shazia et al. 2018. *Social network analysis: An overview, (Online), 8(5),* (<https://wires.onlinelibrary.wiley.com>), diakses 3 April 2023.

Ting, S.L. et al.2011*, Is Naïve Bayes a Good Classifier for Document Classification?.* *International Journal of Software Engineering and Its Applications, (Online),* 5(3)*,* 37-46*,* (<http://sersc.org/journal/index.php/ijseia>), diakses 4 April 2023.

Undap, Michella G. dkk. Analisis Sentimen Situs Pembajak Artikel Penelitian Menggunakan Metode *Lexicon-Based*. *JOINTER:Journal Of Informatics Engineering, (Online),* 2(2), 39-46, diakses 4 April 2023.

Wahid, Aceng Abdul. 2020. Analisis Metode Waterfall Untuk Pengembangan Sistem Informasi*. Jurnal Ilmu-Ilmu Informatika Dan Manajemen STMIK*, (*Online)*, 1-5, (<https://ejournal.stmik-sumedang.ac.id>), diakses 3 Mei 2023.

WilfredNJH. 2022. *TwitterWebScraper*[kode sumber]*.* [*https://github.com/wilfredNJH/Twitter-WebScraper*](https://github.com/wilfredNJH/Twitter-WebScraper)*.* diakses 23 Maret 2023.

Winarno, Edi dkk*.* 2014*. 3 in 1: Javascript, jQuery dan jQuery Mobile.* Cetakan ke-1Jakarta: PT. Elex Media Komputindo.

**LAMPIRAN**

Lampiran 1. *Script* Program Flask

from flask import Flask, request, jsonify,request,redirect

from flask\_mysqldb import MySQL

import pandas as pd

import json

from sentimen import \*

from cleaning\_data import \*

from werkzeug.utils import secure\_filename

import os

from netwok\_als import \*

from lstm\_predictor import \*

from flask\_cors import CORS,cross\_origin

from io import StringIO

app = Flask(\_\_name\_\_)

app.config['MYSQL\_HOST'] = 'localhost'

app.config['MYSQL\_USER'] = 'root'

app.config['MYSQL\_PASSWORD'] = ''

app.config['MYSQL\_DB'] = 'sentimen-analysis'

mysql = MySQL(app)

@app.route('/submit', methods=['POST'])

@cross\_origin()

def index():

    hasil\_sentimen = proses\_data\_kalimat()

    return jsonify({

    "data" : {

        "sentimen":hasil\_sentimen

        }

    })

@app.route('/upload\_senti\_graph', methods=['POST'])

def upload\_sentimen():

    id = request.form.get('id')

    nama\_file = request.form.get('nama\_file')

    print(nama\_file)

    df = pd.read\_csv(StringIO(request.form.get('csv')))

    csv\_file\_path = r'D:\Skripsi\archive\Skripsi\Web\flask\static\upload\\'+id+'\_'+nama\_file # Specify the path where you want to save the CSV file

    print(csv\_file\_path)

    if not os.path.exists(csv\_file\_path):

        df.to\_csv(csv\_file\_path, index=False)

        proses\_sentimen(nama\_file,id)

        result1 = str(id)+'\_'+'Sentimen\_Analysis.csv'

        result2 = str(id)+'\_'+'persentase.png'

        result3 = str(id)+'\_'+'wc\_all.png'

        result4 = str(id)+'\_'+'wc\_pos\_neg.png'

        errors = {}

        success = False

    success = True

    cur = mysql.connection.cursor()

    cur.execute("INSERT INTO sentimenresults (sentimen\_id, result1, result2, result3, result4) VALUES (%s, %s, %s, %s, %s)", (int(id),result1,result2,result3,result4))

    mysql.connection.commit()

    if success and errors:

        errors['message'] = 'File(s) successfully uploaded'

        resp = jsonify(errors)

        resp.status\_code = 500

        return resp

    if success:

        resp = jsonify({

            'message' : 'Files successfully uploaded',

            'data' : {

                'id': id,

                'success': success

            }

        })

        resp.status\_code = 201

        #query database untuk menyimpan data nama file gambar dan id sentimen

        return resp

    else:

        resp = jsonify(errors)

        resp.status\_code = 500

        return resp

@app.route('/upload\_network', methods=['POST'])

def upload\_network():

    id = request.form.get('id')

    nama\_file = request.form.get('nama\_file')

    #print(nama\_file)

    df = pd.read\_csv(StringIO(request.form.get('csv')))

    csv\_file\_path = r'D:\Skripsi\archive\Skripsi\Web\flask\static\upload\\'+id+'\_'+nama\_file # Specify the path where you want to save the CSV file

    #print(csv\_file\_path)

    if not os.path.exists(csv\_file\_path):

        df.to\_csv(csv\_file\_path, index=False)

        buat\_network(nama\_file,id)

        result = str(id)+'\_'+'network.png'

        errors = {}

        success = False

    success = True

    cur = mysql.connection.cursor()

    cur.execute("INSERT INTO networkresults (result1, network\_id) VALUES (%s, %s)", (result,id))

    mysql.connection.commit()

    if success and errors:

        errors['message'] = 'File(s) successfully uploaded'

        resp = jsonify(errors)

        resp.status\_code = 210

        return resp

    if success:

        resp = jsonify({

            'message' : 'Files successfully uploaded',

            'data' : {

                'id': id,

                'success': success

            }

        })

        resp.status\_code = 201

        #query database untuk menyimpan data nama file gambar dan id sentimen

        return resp

    else:

        resp = jsonify(errors)

        resp.status\_code = 500

        return resp

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

Lampiran 2. *Script* program *sentimental analysis*

import re

import string

from nltk.tokenize import word\_tokenize

from nltk.corpus import stopwords

from Sastrawi.Stemmer.StemmerFactory import StemmerFactory

from Sastrawi.StopWordRemover.StopWordRemoverFactory import StopWordRemoverFactory

from wordcloud import WordCloud

import os

import pandas as pd

from cleaning\_data import \*

from lexicon import \*

import matplotlib.pyplot as plt

from io import StringIO

def proses\_sentimen(filename, id):

    a = id

    file\_path = 'static/upload/' + str(a) + '\_' + filename

    tweets\_data = pd.read\_csv(file\_path)

    columns\_to\_convert = ['username', 'handle','responding']

# Convert selected columns to string

    tweets\_data[columns\_to\_convert] = tweets\_data[columns\_to\_convert].astype(str)

    tweets\_data = tweets\_data.rename(columns={'username': 'account\_name', 'handle': 'username', 'postDate': 'created\_at', 'responding': 'tweet'})

    tweets\_sentimen = tweets\_data[['account\_name', 'username', 'created\_at', 'tweet', 'retweets', 'likes']]

    #bersih-bersih karena kebersihan sebagian dari iman

    tweets\_sentimen['text\_clean'] = tweets\_sentimen['tweet'].apply(cleaningText)

    tweets\_sentimen['text\_clean'] = tweets\_sentimen['text\_clean'].apply(casefoldingText)

    tweets\_sentimen.drop(['tweet'], axis = 1, inplace = True)

    tweets\_sentimen['text\_preprocessed'] = tweets\_sentimen['text\_clean'].apply(tokenizingText)

    tweets\_sentimen['text\_preprocessed'] = tweets\_sentimen['text\_preprocessed'].apply(filteringText)

    tweets\_sentimen['text\_preprocessed'] = tweets\_sentimen['text\_preprocessed'].apply(stemmingText)

    tweets\_sentimen.drop\_duplicates(subset = 'text\_clean', inplace = True)

    tweets\_sentimen = tweets\_sentimen[tweets\_sentimen['text\_clean'].str.split().apply(len) > 1]

    results = tweets\_sentimen['text\_preprocessed'].apply(sentiment\_analysis\_lexicon\_indonesia)

    results = list(zip(\*results))

    tweets\_sentimen['polarity\_score'] = results[0]

    tweets\_sentimen['polarity'] = results[1]

    tweets\_sentimen.to\_csv(r'D:\Skripsi\archive\Skripsi\Web\flask\static\CSV\\'+str(a)+'\_'+'Sentimen\_Analysis.csv', index=False)

    fig, ax = plt.subplots(figsize = (7, 7))

    sizes = [count for count in tweets\_sentimen['polarity'].value\_counts()]

    labels = list(tweets\_sentimen['polarity'].value\_counts().index)

    explode = (0.1, 0, 0)

    ax.pie(x = sizes, labels = labels, autopct = '%1.1f%%', explode = explode, textprops={'fontsize': 14})

    ax.set\_title('Persentase Polaritas Sentimen  \n', fontsize = 16, pad = 20)

    plt.savefig(r'D:\Skripsi\archive\Skripsi\Web\flask\static\plot\\'+str(a)+'\_'+'persentase.png')

    plt.close()

    partikel = ('yg','pk', 'ma','jd','ga','ya','aja','tdk','dgn','und','klo','sih','nya','si')

    list\_words=''

    for tweet in tweets\_sentimen['text\_preprocessed']:

        for word in tweet:

            if word not in partikel:

                list\_words += ' '+(word)

    wordcloud = WordCloud(width = 600, height = 400, background\_color = 'black', min\_font\_size = 10).generate(list\_words)

    fig, ax = plt.subplots(figsize = (8, 6))

    ax.set\_title('Word Cloud of Tweets Data', fontsize = 18)

    ax.grid(False)

    ax.imshow((wordcloud))

    fig.tight\_layout(pad=0)

    ax.axis('off')

    ax.set\_title('Wordcloud Keseleruhan data \n', fontsize = 16, pad = 20)

    plt.savefig(r'D:\Skripsi\archive\Skripsi\Web\flask\static\plot\\'+str(a)+'\_'+'wc\_all.png')

    plt.close()

    sentiment\_words = tweets\_sentimen['text\_preprocessed'].apply(words\_with\_sentiment)

    sentiment\_words = list(zip(\*sentiment\_words))

    positive\_words = sentiment\_words[0]

    negative\_words = sentiment\_words[1]

    fig, ax = plt.subplots(1, 2,figsize = (12, 10))

    list\_words\_postive=''

    for row\_word in positive\_words:

        for word in row\_word:

            list\_words\_postive += ' '+(word)

    wordcloud\_positive = WordCloud(width = 800, height = 600, background\_color = 'black', colormap = 'Greens'

                                , min\_font\_size = 10).generate(list\_words\_postive)

    ax[0].set\_title('Word Cloud of Positive Words on Tweets Data \n (based on Indonesia Sentiment Lexicon)', fontsize = 14)

    ax[0].grid(False)

    ax[0].imshow((wordcloud\_positive))

    fig.tight\_layout(pad=0)

    ax[0].axis('off')

    list\_words\_negative=''

    for row\_word in negative\_words:

        for word in row\_word:

            list\_words\_negative += ' '+(word)

    wordcloud\_negative = WordCloud(width = 800, height = 600, background\_color = 'black', colormap = 'Reds'

                                , min\_font\_size = 10).generate(list\_words\_negative)

    ax[1].set\_title('Word Cloud of Negative Words on Tweets Data \n (based on Indonesia Sentiment Lexicon)', fontsize = 14)

    ax[1].grid(False)

    ax[1].imshow((wordcloud\_negative))

    fig.tight\_layout(pad=0)

    ax[1].axis('off')

    plt.savefig(r'D:\Skripsi\archive\Skripsi\Web\flask\static\plot\\'+str(a)+'\_'+'wc\_pos\_neg.png')

    plt.close()

    return tweets\_sentimen

Lampiran 3. *Script p*rogram *network analysis*

import pandas as pd

import re

import string

import networkx as nx

import matplotlib

matplotlib.use('agg')

import matplotlib.pyplot as plt

from cleaning\_data import clean\_data\_net

def buat\_network(filename,id):

    a = id

    file\_path = 'static/upload/' + str(a) + '\_' + filename

    network\_analysis = pd.read\_csv(file\_path)

    network\_analysis['responding'] = network\_analysis['responding'].apply(str)

    network\_analysis['target'] = network\_analysis['responding'].apply(clean\_data\_net)

    network\_analysis.drop(['username', 'postDate','responding','retweets','likes'], axis=1, inplace=True)

    new\_data = []

    for index, row in network\_analysis.iterrows():

        sources = row["handle"].split(',')

        targets = row["target"].split()

        for source in sources:

            for target in targets:

                new\_data.append({"source": source.strip(), "target": target.strip()})

    new\_df = pd.DataFrame(new\_data)

    new\_df["target"] = "@" + new\_df["target"]

    G = nx.Graph()

    nodes = set(new\_df["source"]).union(new\_df["target"])

    G.add\_nodes\_from(nodes)

    edges = list(zip(new\_df["source"], new\_df["target"]))

    G.add\_edges\_from(edges)

    degree\_centrality = nx.degree\_centrality(G)

    # Visualisasi grafik dengan Spring layout

    pos = nx.spring\_layout(G, k=0.4)

    node\_size = [3000 \* degree\_centrality[node] for node in G.nodes()]

    plt.figure(figsize=(12, 9))

    nx.draw(G, pos, node\_color='lightblue', node\_size=node\_size, font\_size=12, alpha=0.7)

    nx.draw\_networkx\_labels(G, pos, font\_size=8, verticalalignment="bottom")

    # Menampilkan gambar

    plt.savefig(r'D:\Skripsi\archive\Skripsi\Web\flask\static\plot\\'+a+'\_'+'network.png')

    plt.close()

    return network\_analysis